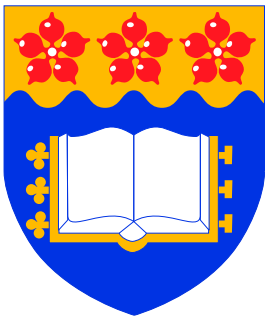
2010

Scalable watermarking for images

Angela Piper



Scalable Watermarking for Images

A thesis submitted in fulfilment of the
requirements for the award of the degree

Doctor of Philosophy

from

University of Wollongong

by

Angela Piper

School of Computer Science and Software Engineering
Faculty of Informatics
March 2010

For my grandparents,
who encouraged my interest in all things.

Certification

I, Angela Piper, declare that this thesis, submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Computer Science and Software Engineering, Faculty of Informatics, University of Wollongong, is wholly my own work unless otherwise referenced or acknowledged below. The document has not been submitted for qualifications at any other academic institution.

Angela Piper
September 2, 2010

Abstract

Digital watermarking allows the embedding of a signal in multimedia content without affecting its quality or usability. This signal can then be used to identify or confirm the creator or owner, track or prevent unauthorised distribution or verify the integrity of the content, and for a variety of other purposes; making digital watermarking a useful component in the protection of digital multimedia content.

With growing demand for both high quality and mobile content, scalable compression has become an increasingly important tool in the distribution of digital content online. It allows content to be scaled to suit a wide range of users and device capabilities in changing network conditions. In particular, resolution scalable compression allows adaptation to different display resolutions, and quality scalable compression caters for to different bandwidths.

However, for low end devices and low bandwidth connections, the scaling process typically alters the content beyond what a traditional digital watermark is designed to handle. Thus the field of scalable watermarking has emerged to provide digital watermarking algorithms that are suitable for scalably compressed content.

In this thesis, resolution and quality scalable watermarking is examined in the context of images, with the aim of developing a watermarking algorithm that is both resolution and quality scalable.

Precisely what it meant for a watermarking algorithm to be *scalable* had not previously been formally defined, and informal descriptions would often focus on a certain desirable property for a scalable watermarking algorithm to the exclusion of some other important property. A definition of a scalable watermarking algorithm is proposed, which considers watermark scalability in terms of two properties that describe the ability of the watermark to both survive high levels of scalable compression and yet still adequately protect all layers of the image. Quantitative measures of these properties are also constructed, to allow the scalability of a watermarking algorithm to be evaluated according to the proposed definition.

Although scalable image compression allowed two types of scalability – resolution and quality – prior to the work in this thesis, scalable watermarking algorithms typically

provided only one type of scalability or the other (or provided both with the provision that one of the two types be selected at the time of embedding). The problem of creating a single watermarking algorithm that provides both resolution and quality scalability, simultaneously, is considered, using JPEG2000 as a representative scalable compression algorithm.

A non-blind spread spectrum based watermarking algorithm is developed, by combining resolution scalable coefficient selection with a human visual system based embedding strength. The resulting watermark provides both detectability and graceful improvement, allowing resolution scaling to $\frac{1}{256}$th the original image area, and quality scaling to $\frac{1}{100}$th the original file size, exceeding the resolution and quality scalability reported for other scalable image watermarking algorithms.

A blind, quantization based algorithm is also developed, that provides detectability and graceful improvement for resolution scaling to $\frac{1}{1024}$th the area and quality scaling to $\frac{1}{100}$th the size, and additionally maintains an exact match between the candidate and extracted watermarks under scaling. This algorithm is adapted for image authentication, and is sensitive to small changes, including Holliman-Memon and collage attacks, but remains undamaged by JPEG2000 scaling. Previous image authentication watermarks that both tolerate scaling and are secure against Holliman-Memon and collage attacks typically offer only one type of scalability and do not provide graceful improvement.

Acknowledgements

Despite the nature of PhD candidature as a largely solitary process, I find myself with many people to thank[1].

Firstly I would like to thank my supervisor, Reihaneh Safavi-Naini, for all her time and effort, and for teaching me the importance of hard work and determination, in research and in life generally. I would also like to thank both Alfred Mertins, my co-supervisor during the early stages of candidature, and Philip Ogunbona, my co-supervisor during the later stages, for their technical advice and for helping me to improve my research.

I have benefited greatly from the advice of Pam Davy and David Griffiths, who generously provided their time and expertise to further my understanding of statistical analysis. This thesis would be poorer were it not for their assistance.

This thesis has also been enhanced by the efforts of Ian Piper and Gary Watson. I am grateful to them both for proofreading my thesis, and for acting as though that task were not the tedious and unpleasant chore that I am sure it must have been.

I very much appreciate the assistance of the Smart Internet Technology CRC, not only for the partial funding of my research, but for providing opportunities to converse with a variety of PhD students and other researchers from many disciplines, whom I would not otherwise have encountered.

So many staff in the Faculty of Informatics offered their encouragement, support or assistance. I wish there were room to acknowledge more than Gene Awyzio, Bob Brown, Peter Castle, Wanqing Li, Luke McAven, Yi Mu, Jennifer Seberry and Minjie Zhang from the academic staff; Michael Milway and David Wilson from the IT support staff; and Donna Edwards, Sonia Jennings, Anne Nealon, and Karen Williams from the administrative staff.

I am also grateful to Wily Susilo, in his role as Co-Director of the Center for Computer and Information Security Research, for providing part of the funding that allowed me to visit my supervisor in Calgary. This visit was enhanced immeasurably by the kind hospitality of Sirpa Torrence and her family, who provided me with a second home for the

[1] This provides me with a sudden sympathy with anyone who has ever felt compelled to write an overlong acceptance speech. I comfort myself with the thought that should my own audience become weary, they need only turn the page

duration of my stay; and by the students and other researchers at iCORE, who welcomed me into their lab.

To my many labmates in Wollongong, especially Gareth Brisbane, Jeffrey Horton, Wenming Lu, Mohammad Reyhanitabar, Martin Surminen, Siamak Shahandashti, Pairat Thorncharoensri, Dongvu Tonien, Takeyuki Uehara, and Rungrat Wiangsripanawan, thank you. I have enjoyed our discussions, and my time as a PhD student has been greatly enriched by sharing a lab with you. (In addition, my heartfelt gratitude to the unknown labmate whose abandoned heater I inherited.)

Thank you to my friends, for your patience at my neglect, your understanding when I couldn't stop talking in maths or computerese, and for your support, your encouragement and your excitement on my behalf. I am lucky to have you. Particular thanks to Mathew Baddams, Jennifer Cowland and Suzanne Thompson, who have been there for me to rely on since I began this.

Thank you to my cousin Emily Milburn and her husband Nathan, for convincing me to take a break and visit you in Melbourne right when I needed it, and for looking after me so well. Also to my friends Kieron Briggs, Steven Kreusser and Emma Lord for making that break even better.

Indeed, thank you to all of my family. Especially to my aunt and uncle, Julie and Clark Hodgson, for their love and support always but also for doing all the things that I couldn't do to help my grandparents during these past years.

Finally, thank you to my parents, Ian and Sandra Piper, for everything and everything.

Publications

The following is a list of the jounral articles and peer reviwed conference papers that have been presented during the course of the work conducted as part of this thesis.

A. Piper and R. Safavi-Naini. "How to compare image watermarking algorithms," Y. Q. Shi, Ed., *Trans. Data Hiding and Multimedia Security IV*, pp. 1–28, ser. Lecture Notes in Computer Science, vol. 5510, Springer Berlin/Heidelburg, 2009.

A. Piper, R. Safavi-Naini, and A. Mertins. "Resolution and quality scalable spread spectrum image watermarking," in A. M. Eskicioglu, J. J. Fridrich and J. Dittmann, Eds., *Proc. 7th ACM Workshop on Multimedia & Security (MM&Sec'05), New York, NY, USA, Aug. 1-2*, ACM Press, 2005. pp. 79–90.

A. Piper. "Refined threshold adaptation for scalable watermarking," (best student paper award) [Abstract], in J. Fulcher and K. Ward, Eds., *SITACS Research Student Conference Abstracts.* School of Information Technology and Computer Science, University of Wollongong, Oct. 27, 2004.

A. Piper, R. Safavi-Naini, and A. Mertins. "Coefficient selection methods for scalable spread spectrum watermarking," in T. Kalker, I. J. Cox, Y. M. Ro, Eds., *Proc. Digital Watermarking: Second Intl. Workshop (IWDW), Seoul, Korea, Oct. 20–22*, ser. Lecture Notes in Computer Science, vol. 2939. Springer Berlin/Heidelburg, 2003, pp. 235–246.

Contents

1 Introduction **1**

1.1 Historical Background . . . 1

1.1.1 Marking Before Watermarks . . . 1

1.1.2 Paper Watermarking . . . 2

1.1.3 Digital Watermarking . . . 4

1.1.4 Scalable Compression . . . 6

1.2 The Motivation for Scalable Watermarking . . . 8

1.3 Objectives, Structure and Scope . . . 10

2 Background **13**

2.1 Watermarking . . . 13

2.1.1 Definition . . . 13

2.1.2 Properties . . . 15

2.1.3 Applications of Watermarking Algorithms . . . 19

2.1.4 Attacks on Watermarking . . . 23

2.1.5 Watermarking Techniques . . . 26

2.2 Scalable Compression . . . 29

2.2.1 Compression . . . 29

2.2.2 Scalable Compression . . . 35

2.2.3 JPEG2000 . . . 40

3 Definition and Evaluation of Scalable Watermarking **51**

3.1 Definition of Scalable Watermarking . . . 51

3.1.1 Proposed Definition . . . 52

3.1.2 Measuring Scalability Properties . . . 54

3.1.3 Scalable Watermarking Literature . . . 60

3.2 Evaluation of Scalable Watermarking . . . 65

3.2.1 Experimental Method . . . 66

3.2.2 Benchmarking Literature . . . 66

3.2.3 Fair Evaluation . . . 68
3.2.4 Comparing Algorithms . . . 74

4 Scalable Spread Spectrum Watermarking 87
4.1 FalsePositive Modelling for Spread Spectrum Watermarks . . . 89
4.2 The Effect of Coefficient Selection on Watermark Scalability . . . 100
4.2.1 Coefficient Selection Schemes . . . 102
4.2.2 Experimental Evaluation of Coefficient Selection Schemes . . . 103
4.2.2.1 Experimental Framework . . . 103
4.2.2.2 Results . . . 105
4.2.3 The Resolution/Quality Tradeoff . . . 111
4.3 HVS Adaptation to Alleviate the Resolution-Quality Tradeoff . . . 113
4.3.1 HVS Adaptation . . . 114
4.3.2 Comparative Evaluation of Scalability with HVS Adaptation . . . 122
4.3.2.2 Experimental Framework . . . 124
4.3.2.3 Results . . . 133
4.4 Conclusion . . . 137

5 A Blind Scalable Watermark for JPEG2000: Basic Algorithm 139
5.1 Design . . . 140
5.1.1 Coefficient Selection and Embedding . . . 140
5.1.2 Effects of scaling on a quantized coefficient . . . 143
5.1.3 Watermark Extraction . . . 144
5.1.4 Watermark Generation . . . 146
5.1.5 Candidate Truncation . . . 152
5.1.6 Calculating the Number of Missing Bits . . . 153
5.1.7 Calculating the Number of Watermark Bits . . . 155
5.1.8 The Blind Scalable Watermarking Algorithm . . . 156
5.2 Evaluation of the Basic Algorithm . . . 158
5.2.1 Correctness and Fragility . . . 159
5.2.2 Scalability . . . 186
5.2.3 Tamper Detection . . . 192
5.3 Conclusion . . . 210

6 A Blind Scalable Watermark for JPEG2000: with Improved Security 213
6.1 Design of a Secured Algorithm . . . 214
6.1.1 Coefficient Selection and Embedding . . . 215
6.1.2 Indexing . . . 216

6.1.3 Watermark Element Generation . . . 217
6.1.4 Calculating the Number of Missing Bits . . . 225
6.1.5 Candidate Generation . . . 226
6.1.6 Watermark Extraction . . . 227
6.1.7 Detection Output . . . 228
6.1.8 Blind Scalable Watermarking Algorithm with Improved Security . . 229
6.1.9 Design Outcomes . . . 231
6.2 Related Work . . . 232
6.2.1 Fragile Watermarking . . . 233
6.2.2 Semi-Fragile Watermarking . . . 235
6.3 Scalability in Image Authentication . . . 239
6.3.1 A Note on Fragility to Modifications Other than Scaling . . . 242
6.4 Evaluation of the Secured Watermarking Algorithm . . . 243
6.4.1 Experimental Framework . . . 243
6.4.2 Correctness and Fragility . . . 245
6.4.3 Scalability . . . 250
6.4.4 Tamper Detection . . . 255
6.5 Conclusion . . . 263

7 Concluding Remarks **267**
7.1 Main Contributions . . . 268
7.1.1 Definition of Scalable Watermarking . . . 268
7.1.2 Non-Blind Resolution and Quality Scalable Watermarking . . . 269
7.1.3 Blind Resolution and Quality Scalable Watermarking . . . 269
7.1.4 Resolution and Quality Scalable Authentication . . . 270
7.2 Limitations and Future Work . . . 271
7.2.1 A Standard, Comprehensive Database of Test Images . . . 271
7.2.2 Statistical Methods for Algorithm Evaluation . . . 272
7.2.3 Watermarking-Specific Human Visual System Models . . . 273
7.2.4 Extension to More Advanced Watermarking Tecniques . . . 274
7.2.5 Near Perfect Graceful Improvement . . . 274
7.2.6 Improved Resolution and Quality Detectability . . . 274
7.2.7 Extension to Other Media . . . 275

Glossary **277**

Acronyms **285**

Bibliography **287**

Index **310**

Appendix A Images used in this Thesis **317**

Appendix B Definition and Evaluation of Scalable Watermarking **347**

B.1 Normalization of Graceful Improvement 347

B.2 Two Sample t-test 349

Appendix C Development of the Texture Scoring Algorithm **351**

C.1 Characterizing Texture in the Wavelet Domain 351

C.2 Improving the Separation of Textured Regions 356

C.3 Experiments 360

C.4 Evaluation 367

Appendix D A Blind Scalable Watermark for JPEG2000: Basic Algorithm **377**

D.1 Algorithm Examples 377

D.2 Lemmas 381

D.3 Proofs of Basic Design Features 383

D.4 Additional Details on the Evaluation of the Basic Algorithm 408

D.5 Identifying Coefficients Corresponding to a Rectangular Region 478

Appendix E A Blind Scalable Watermark for JPEG2000: with Improved Security **487**

E.1 Proofs of Design Features for the Secured Algorithm 487

E.2 Additional Details on the Evaluation of the Improved Algorithm 526

List of Tables

2.1 Classification of watermarking outcomes 14

4.1 Standard deviations of paired differences between `hvs` and other algorithms 127
4.2 Standard deviation of detectability values for each algorithm 128
4.3 Standard deviation of graceful improvement values for each algorithm . . 129
4.4 Minimum 'substantial' performance difference for each measure 129
4.5 Required number of images for each algorithm and measure 130
4.6 Paired t-test – detectability comparison between `hvs` and other algorithms 133
4.7 Sign test – detectability comparison between `hvs` and `nohvs` 133
4.8 False negative rate calculated from 65 original images watermarked with 10 keys 134
4.9 Estimated rate of false negative errors for each algorithm 135
4.10 Sign test – graceful improvement comparison between `hvs` and other algorithms 136
4.11 Paired t-test – graceful improvement comparison between `hvs` and other algorithms 136

5.1 Total BER and extracted bits under resolution scaling 163
5.2 Total BER and extracted bits under quality scaling 163
5.3 BER after recompression under resolution scaling 166
5.4 BER after recompression under quality scaling 166
5.5 BER after Gaussian filtering under resolution scaling 168
5.6 BER after Gaussian filtering under quality scaling 168
5.7 BER after hardthresh filtering under resolution scaling 169
5.8 BER after hardthresh filtering under quality scaling 169
5.9 BER after JPEG100 under resolution scaling 170
5.10 BER after JPEG100 under quality scaling 170
5.11 BER after JPEG40 under resolution scaling 171
5.12 BER after JPEG40 under quality scaling 171
5.13 BER after median2 $\times$ 2 filtering under resolution scaling 172

5.14 BER after median2 × 2 filtering under quality scaling 172
5.15 BER after midpoint3 × 3 filtering under resolution scaling 173
5.16 BER after midpoint3 × 3 filtering under quality scaling 173
5.17 BER after trimmed mean under resolution scaling 174
5.18 BER after trimmed mean under quality scaling 174
5.19 BER after sampledownup75 under resolution scaling 175
5.20 BER after sampledownup75 under quality scaling 175
5.21 BER after Laplacian3 × 3 filtering under resolution scaling 176
5.22 BER after Laplacian3 × 3 filtering under quality scaling 176
5.23 BER after JPEG2000 at rate 0.0125 under resolution scaling 177
5.24 BER after JPEG2000 at rate 0.0125 under quality scaling 177
5.25 BER after 10% cropping under resolution scaling 179
5.26 BER after 10% cropping under quality scaling 179
5.27 BER after linear transformation under resolution scaling 180
5.28 BER after linear transformation under quality scaling 180
5.29 BER after projective transformation under resolution scaling 181
5.30 BER after projective transformation under quality scaling 181
5.31 BER after 1° rotation under resolution scaling 182
5.32 BER after 1° rotation under quality scaling 182
5.33 BER after 45° rotation under resolution scaling 183
5.34 BER after 45° rotation under quality scaling 183
5.35 BER after 50% rescaling under resolution scaling 184
5.36 BER after 50% rescaling under quality scaling 184
5.37 BER after a copy attack under resolution scaling 185
5.38 BER after a copy attack under quality scaling 186

6.1 Total bit errors and extracted bits for resolution scaled subimages 246
6.2 Total bit errors and extracted bits for quality scaled subimages 246
6.3 BER given an incorrect detection key under resolution scaling 248
6.4 BER given an incorrect detection key under quality scaling 248
6.5 BER after recompression under resolution scaling 250
6.6 BER after recompression under quality scaling 251
6.7 BER after mark transfer attack under resolution scaling 256
6.8 BER after mark transfer attack under quality scaling 256
6.9 BER after collage attack under resolution scaling 260
6.10 BER after collage attack under quality scaling 260

List of Figures

1.1 Depiction of the earliest known watermark 3
1.2 The earliest audio watermarking method 5
1.3 Progressive transmission and non-progressive transmission 7

2.1 JPEG2000 encoder structure 41
2.2 JPEG2000 decoder structure 41
2.3 Subbands in an R resolution layer wavelet decomposition 43
2.4 The Mandrill image and its wavelet transform 44
2.5 Embedded quantizer interval boundaries and reconstruction points 46

4.1 Spread spectrum embedder within JPEG2000 encoder structure 88
4.2 Spread spectrum detector within JPEG2000 decoder structure 88
4.3 Average embedding strengths for the coefficient selection schemes 106
4.4 Resolution detectability results for the coefficient selection schemes 108
4.5 Quality detectability results for the coefficient selection schemes 109
4.6 Resolution graceful improvement results for the coefficient selection schemes 111
4.7 Quality graceful improvement results for the coefficient selection schemes 112
4.8 The tradeoff between resolution and quality detectability 113
4.9 A Campbell-Robson contrast sensitivity chart 116
4.10 High resolution texture masks a high resolution modification 120
4.11 Low resolution texture masks a low resolution modification 120
4.12 Normal quantile plots of paired differences 132
4.13 Average detectability values for each algorithm over 65 images 134
4.14 Average graceful improvement values for each algorithm over 65 images . 136

5.1 Blind scalable embedder within JPEG2000 encoder structure 141
5.2 Blind scalable detector within JPEG2000 decoder structure 143
5.3 BER given an incorrect detection key under resolution scaling 165
5.4 BER given an incorrect detection key under quality scaling 165
5.5 Resolution detectability 189

5.6 Quality detectability . . . 189
5.7 Quality detectability at compression rate 0.01 . . . 190
5.8 Resolution graceful improvement . . . 191
5.9 Quality graceful improvement . . . 191
5.10 The original Greek isles image . . . 194
5.11 The watermarked Greek isles image . . . 194
5.12 Tamper map for the watermarked Greek isles image . . . 196
5.13 Spatially tampered image . . . 197
5.14 Tamper map for the spatially tampered image . . . 197
5.15 Wavelet tampered image . . . 199
5.16 Tamper map for the wavelet tampered image . . . 199
5.17 Tampered image with transferred watermark . . . 202
5.18 Tamper map with transferred watermark . . . 202
5.19 The original Lena image . . . 204
5.20 Tampered image with transferred watermark, compression rate 0.02 . . . 204
5.21 Zoomed view of tampered image, compression rate 0.02 . . . 205
5.22 Tamper map with transferred watermark, compression rate 0.02 . . . 205
5.23 Tampered image with 6-image collage . . . 207
5.24 Tamper map for 6-image collage . . . 207
5.25 Tampered image with 21-image collage . . . 208
5.26 Tamper map with 21-image collage . . . 208
5.27 Tampered image with 81-image collage . . . 209
5.28 Tamper map with 81-image collage . . . 209

6.1 Tampering without changing the lowest resolution or quality layer . . . 240
6.2 BER given an incorrect detection key under resolution scaling . . . 249
6.3 BER given an incorrect detection key under quality scaling . . . 249
6.4 Resolution detectability . . . 252
6.5 Quality detectability . . . 252
6.6 Resolution graceful improvement . . . 253
6.7 Quality graceful improvement . . . 254
6.8 Per-layer proportion by which extracted bits exceed the ideal . . . 254
6.9 Tampered image with transferred watermark . . . 257
6.10 Tampered image with transferred watermark compression rate 0.02 . . . 258
6.11 Tamper map with transferred watermark compression rate 0.02 . . . 258
6.12 Watermarked Greek isles image . . . 260
6.13 Wavelet tampered image . . . 261

6.14 Tampered image with 21-image collage 261
6.15 Tamper map with 21-image collage 262
6.16 Tamper map with 21-image collage, without full saturation 262

Chapter 1

Introduction

1.1 Historical Background

1.1.1 Marking Before Watermarks

Scalable watermarking is a recent chapter in the long story of marking in general. Throughout history, people have unobtrusively marked objects for purposes similar to those for which scalable watermarking is intended: to identify the owner or creator, track theft, establish authenticity or simply to provide information about the marked object.

The earliest known examples of this are potters' marks and livestock branding – both of which still occur today. Markings which appear to denote ownership[1] have been found on pottery in Iran dated as early as 3000-2800 BC [147], and in Transylvania perhaps as early as 5000 BC [153]. Unlike decorations, these markings were placed so as to be unobtrusive, near the base of the vessel or beneath the stopper. More certain evidence that potter's marks were used to identify the owner or creator can be found on Chinese pots c. 2700 BC [135], which were marked with the name of the Emperor, maker or place of origin.

Cave paintings in southeastern Europe, of the late stone or early bronze age [153], and in Saharan Africa, c. 3000 BC [162], depict branded livestock. Branded marks [98] enabled assertion of legitimate ownership, were used to track animals and deter theft, and could also carry information about the lineage or social status of the owner; furthermore, they were often carefully positioned so as not to reduce the usefulness of the animal by damaging its hide.

[1] Potts [147] suggests that (with the exception of a group of marks that appear to be numerical in nature) the marks are most likely to be owners' or makers' marks, but notes the use of the marks as content descriptors as a possible alternative. He speculates, based on the number of different symbols used, that they may identify family ownership for object tracking in a communal kiln.

Earlier examples of marked objects found in China are reported in [104], most notably tortoise shell and bone at the Jiahu site, dated 6600–6200 BC and pottery at Yangshao sites, dated 5000–4500 BC; however, the purpose of the markings is not stated and Li et al. suggest that the markings on the Jiahu objects may be ritual in nature.

By Roman times [152], similar marks could be found on a wide variety of goods, including bronze, gold, silver and iron articles, lead pipe, marble, gems and bread. Dry cakes of Roman eye salve have been found stamped with the name of the physician, the formula and directions for use, as have earthen vessels containing liquid medicines. A potential example of counterfeit marks, imprinted for the purpose of falsely claiming authenticity, occurs early in this period: it has been suggested that some Etruscan vases c. 800-400 BC, bearing Greek inscriptions, are Roman counterfeits of Greek products.

In the middle ages [60], marks were not merely widespread but often compulsory. Guild regulations required that all members mark their products with both the unique mark of the guild and the registered mark of the individual artisan. This helped maintain the guilds' monopoly on production and trade by providing a recognisable assurance of quality to the customer and allowing counterfeiters, and guild members producing poor quality goods, to be identified. In 1266, the marking of bread became required by English law, to better track the compliance of bakers with regulations regarding the weights of different types of loaves.

The marking of objects, and the provision of laws surrounding such marking, continues to the present day, in forms such as company trademarks, artists' signatures and serial numbers. However, we leave this history here, as the middle ages saw the development of a specific kind of mark: the watermark.

1.1.2 Paper Watermarking

The earliest known example of a watermark, a simple Greek cross pommée (figure 1.1), was produced in Bologna, Italy and is believed to date from 1282 [17]. The practice spread to other papermakers and there followed a profusion of other heraldic designs (and, later, non-heraldic designs such as initials, dates and, eventually, portraits). While some watermarks are believed to be purely artistic in purpose [75], other uses of early paper watermarks were as creator's marks for individual papermakers [17], as trademarks for paper-mills [85] and for the identification of paper size[2].

Watermarks were produced not by water but by wire designs, sewn onto the upper surfaces of the wire moulds used in the papermaking process. The design would press into the paper pulp, reducing the fibre density relative to the surrounding pulp, so that the imprint of the design could be seen in the finished product. The French term 'filigrane' is more indicative of this process; the English term 'watermark', and the German equivalent 'Wasserzeichen', occur perhaps because the mark is made while the paper is wet or perhaps because the appearance of a watermark is similar to that of wet paper held up to a light.

[2]For example, foolscap paper, traditionally $8\frac{1}{2}$ by $13\frac{1}{2}$ inches, derives its name from the watermark, of a jester's head (made recognisable by his fools' cap), that was used to identify it.

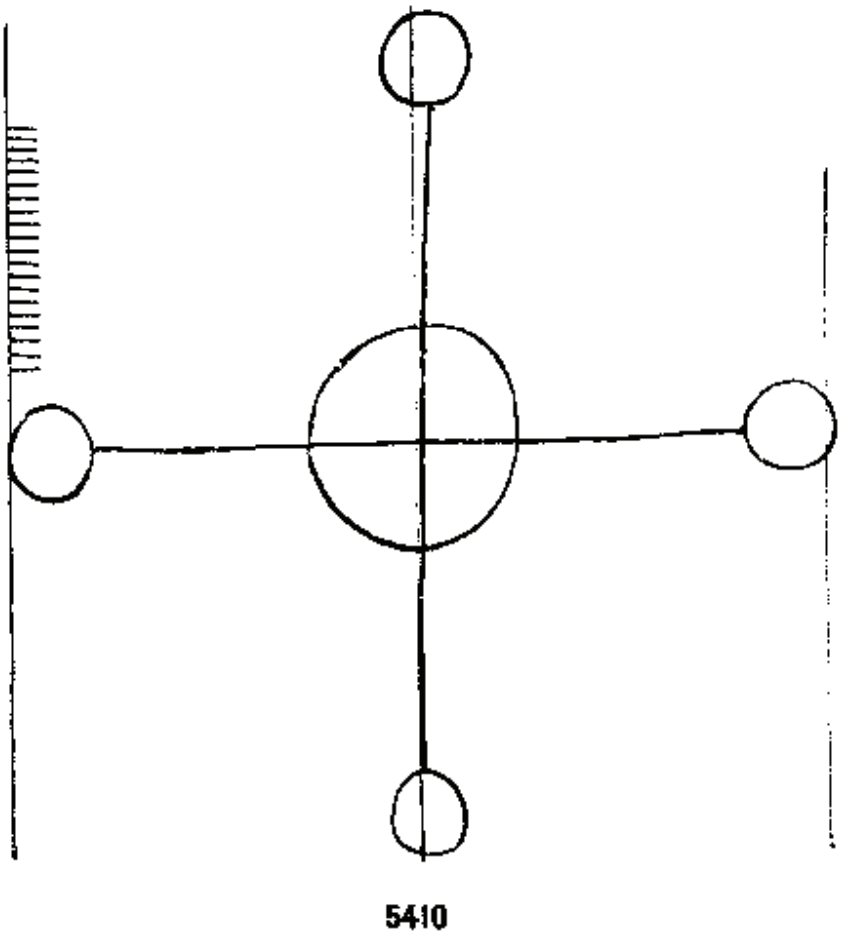


Figure 1.1: A depiction of the earliest known watermark, a Greek cross pommée design. Briquet No. 5410 [17], Bologna Italy, 1282? and also 1287-88. obtained from [134].

The method of watermarking, compressing the wet pulp using a wire design, remained essentially unchanged through a number of improvements in both papermaking and watermarking. Most notably, the invention of the dandy roll watermark in 1826, following the introduction of machine-made paper, used a wire design on the surface of a cylindrical roller that was pressed into the paper while still damp, and the dramatic move to full greyscale 'light and shade' watermarks in 1848, from the essentially binary watermarks that preceded it, used a wire mesh formed from a sculptured template[3]. While similar methods are still in use today, recent history has also seen the patenting of a number of pseudo-watermarking methods that do not alter fibre density, but instead manipulate opacity through the addition of chemicals or layered opaquing agents (see e.g. [62, 195]).

One important feature of paper watermarks is that they are an integral part of the

[3]A more complete examination of paper watermarking, and papermaking, can be found in Hunter's "Papermaking: The History and Technique of an Ancient Craft" [75].

paper, formed by altering the paper itself (typically by changing the density of the paper during the manufacturing process) rather than by marking a design on the paper's surface. This makes them difficult to remove without destroying the paper and also difficult to fake on already existing paper.[4] The latter property in particular led to their playing an important role as an authentication measure on items such as stamps and bank-notes. In 1773, the death penalty was extended to include imitation of the watermark present in English bank-note paper.[5]

Another important feature of paper watermarks is that they are unobtrusive. The presence of the watermark, although clearly visible when the paper is held up to a light, causes only a subtle change in the shade of the paper, that does not distract from whatever is printed upon it. This property allows the identification and security functionality of the watermark to be provided without noticeably reducing the utility of the document.

1.1.3 Digital Watermarking

The unobtrusiveness and integral nature of paper watermarks undoubtedly inspired the first patent of an audio watermarking algorithm. The widespread availability of audio recording technology, specifically the adoption of the home tape recorder in the United States of America shortly after the end of World War II, allowed musical performances to be copied with unprecedented ease. In response, Hembrooke filed a patent [68] for the Muzak corporation in 1954. It described a method that would allow

> "the identification of recorded music or other audio signals by coded signals which are not evident to a listener but which can nonetheless be easily detected and which are such an integral part of the audio signals that they are difficult but impossible to obliterate"

in a manner that

> "may be likened to a watermark in paper".

The method worked by timing the repeated suppression of a very narrow band around a selected frequency in the audio range, to embed a coded message (figure 1.2) that could be decoded and used to identify the origin of the recorded performance.

Although the watermark signal was itself embedded digitally (using the presence and absence of the selected frequency), the Muzak watermark was not a 'digital watermark'

[4] The forgery of paper watermarks during the manufacturing process appeared to have been relatively straightforward, however; at least for simple watermarks. Indeed the use of other printers' marks appears to have been relatively common, limited only by the skill of the forger in reproducing the wire design [152].

[5] The difficulty of faking a watermark on existing paper is underlined by the belief, albeit mistaken, of the counterfeiter John Matheison that he might escape this same penalty in 1778, by offering the secret of his post-manufacture counterfeiting method in exchange for his life [75]

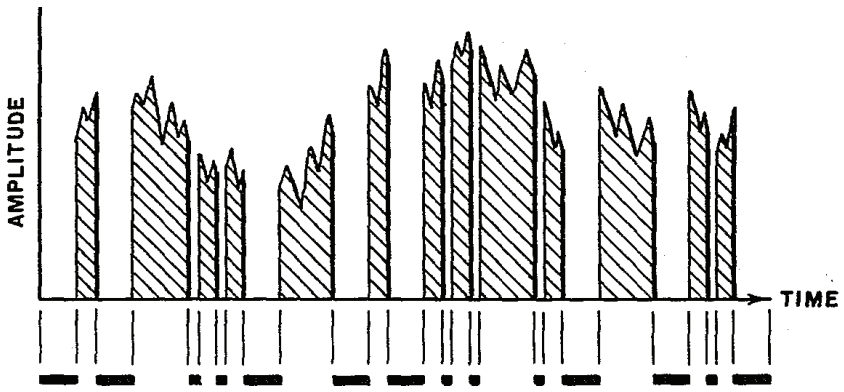


Figure 1.2: A graph depicting the earliest audio watermarking method, taken from the Hembrooke patent [68]. The graph shows the amplitude of the selected audio frequency over time. The selected frequency is alternately suppressed and passed to embed the message 'MUZAK' using Morse code.

in the conventional sense, because the (audio) signal that it protected was analogue.[6] The same may be said for many early watermarks used in the control, enrichment or identification of radio and television services. Interestingly, the method described in the Szepanski paper [183], which is commonly held to be the earliest digital watermarking algorithm, involves the watermarking of paper documents (which are again analogue) and thus also fails to satisfy the conventional definition, for the same reasons. Thus the first digital watermark, in the conventional sense, may well coincide to what is believed [32] to be the first use of the term 'digital watermark', by Komatsu and Tominaga [91] in 1988. Around the same time, CDs began to rival LPs and magnetic tape as the dominant audio technology and in 1989 Turner [191] patented a method for digitally watermarking CDs.

It was in the 1990s, with increasing digitization and the rapidly expanding popularity of the Internet, that watermarking research really took off. The Internet ushered in a new age for copyright violation. Not only was it possible to copy multimedia content simply and cheaply, using a home computer, but also, where previously one could expect the circulation of home-produced copies to extend no further than the local area, the pirated content could be distributed across the globe. Furthermore, digital technology had removed another barrier to piracy, the degradation of content quality inherent to analogue

[6] Although the definition of digital watermarking (section 2.1.1, page 13) used in this thesis follows the conventional one, the wisdom of requiring a 'digital watermark' to be embedded in digital content is somewhat questionable. The Muzak watermarking method, despite being proposed before the invention of digital recording, would have been equally applicable to digital audio. Indeed it carries other hallmarks of a digital watermark, being embedded by altering the signal itself (rather than the storage medium) and detected via machine (i.e. not directly human readable). Thus, but for the unavailability of digital audio in 1954, the Muzak watermarking method would have been the earliest known digital watermarking algorithm.

media. Unlike copies in analogue media, such as videos, music cassettes or photocopied articles, the quality of digital copies remained identical to that of the original, so second- (and third-, and fourth-) generation copies could be produced with no loss of quality.

Digital watermarking was hailed as a solution to all of these piracy problems, and, although it was not quite the silver bullet everyone had hoped, significant advances were made. Like the continuum of marking technologies that preceded it, digital watermarking provided an unobtrusive label, allowing tracking and identification. This was a natural fit to applications such as fingerprinting [14, 16, 184], which enabled the source of a pirated copy to be tracked, and copyright labelling (and, in particular, proof of ownership) [3, 30, 34, 90], which allowed the legitimate owner to be identified (ideally with enough certainty to convince a court). A less traditional use for watermarks in the fight against piracy was copy control [31], which restricted the copying or playback of content, by compliant hardware, according to the contents of the embedded watermark signal.

Not all uses envisaged for digital watermarking were piracy related. While paper watermarks were merely unobtrusive, the move from visual to computerized detection meant that digital watermarks could be invisible to humans yet remain effective. This made digital watermarking ideal for data enrichment [140, 154, 173], allowing information such as subtitles, annotations or web addresses to be imperceptibly embedded in the content itself, and made digital watermarking techniques useful for steganography [51, 94, 176, 209].

Another traditional use of paper watermarking gathered significant interest among digital watermarkers. Authentication watermarking was extended, beyond bank-notes and stamps, to authentic digital media distribution and secure cameras [28, 91, 52, 198, 207, 212]. Cryptographic and content-adaptive techniques were applied to develop watermarks that were increasingly difficult to fake and sensitive to changes in content. These fragile watermarks allowed not only the detection of counterfeit content, but also the location of tampered regions in modified content.

1.1.4 Scalable Compression

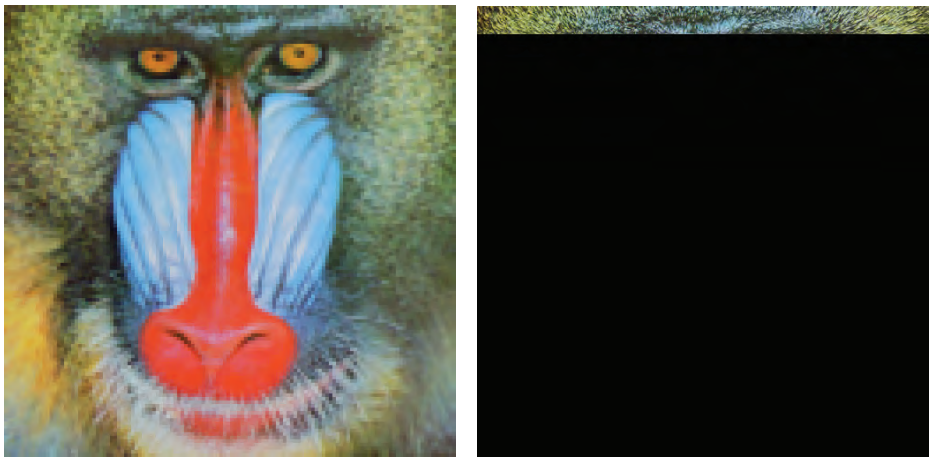
Of course, the Internet was not exclusively the domain of pirates, thieves and counterfeiters; it also provided the creators and distributors of multimedia content with a direct and cost-effective means of reaching a global audience. However, a significant problem with online access to content was the time required to download it, particularly if the content contained many images.

In the late 1970s and early 1980s, most modems were capable of transmission speeds of 1200 bits per second. At this speed, a single uncompressed 512×512 greyscale image at 8 bits per pixel could take half an hour to download. Although each line of the image

was displayed as soon as it was received, it could still be 15 minutes before the user had a reasonable idea what the final picture would be like.

In 1979, Sloan and Tanimoto [171] published their solution to this problem, which may be considered[7] the first resolution scalable compression algorithm. They suggested that that, rather than using line-by-line image transmission, a low-resolution approximation of the image could be sent first, and then successively refined until the complete image had been received.

The effect was dramatic (figure 1.3). Using this approach, the main image features became visible very early in the transmission, so very little waiting was required to obtain a reasonable picture. Furthermore, if the quality was already acceptable, or the image was not what the user had hoped for, transmission could be stopped early, freeing up the bandwidth for other downloads. It was even possible to target a specific area of interest and only download the refinement data for that area.



(a) (b)

Figure 1.3: (a) A progressively transmitted image after $\frac{1}{16}$th of the data has been received. (b) A line-by-line transmitted image after $\frac{1}{16}$th of the data has been received.

The first quality scalable compression algorithm was published five years later, in 1984, by Sanz, Muñoz and García [156]. Before this point, each refinement layer defined by a scalable compression algorithm increased the spatial resolution of the image, but pixel quantization was the same at all refinement layers. Sanz et al. recognised that not

[7]Whether Sloan and Tanimoto's scalable transmission method constitutes 'compression' is debatable. To transmit a full image using their example methods required, at best, the same number of bits as line-by-line methods. However, as observed by Knowlton, who in 1978 filed his patent [89] of a similar method (based in part on earlier work by Tanimoto and Pavlidis' [185]), the ability to cease downloading as soon as each image has achieved its purpose may be considered equivalent to lossy compression. Regardless of whether or not one accepts them as compression techniques, the methods of Sloan and Tanimoto [171] and Knowlton [89] clearly served as the foundation for the many scalable compression algorithms that have been developed since.

all users would require an unquantized image and that a similar, multi-level transmission method could be defined that used images quantized to different numbers of bits per pixel, rather than different spatial resolutions. They further suggested that these techniques could be combined, using either a fixed or an interactively defined sequence of resolution and quality improvements, forshadowing compression algorithms with both resolution and quality scalability such as HS-SPIHT [38], JPEG2000 [81] and EBCOT [187].

Despite a dramatic increase in achievable transmission speeds since the 1980s, the need for scalable compression has not diminished. With non-text data now commonplace, users have significantly higher expectations in terms of both content quality and delivery times. Not only this, but the range of devices being used to access the Internet has expanded, so while a typical mobile phone user may desire content suited to a display with 320×240 resolution, a high-end desktop user would prefer content tailored to a display resolution of 1900×1200. Equally important are the widening differences in access speeds available; some service providers currently offer 200Mbps residential broadband services (to say nothing of the speeds achievable on local networks), while poor location or economic status can result in speeds of at most 0.056Mbps.

Although it is possible to satisfy all device and connection constraints by offering only highly compressed, low resolution content, this would completely fail to meet the quality expectations of users with fast connections and/or high resolution displays. To cater for the widest possible range of users, it is necessary to provide access to content at different resolutions and levels of compression. This could, of course, be achieved using multiple separate copies of the content, at a range of resolutions and compression ratios. However, scalable compression offers a superior solution, in that it only requires the creation and storage of one copy of the content and can adapt dynamically to changing network conditions or user demands.

1.2 The Motivation for Scalable Watermarking

Creators and distributors require the same protections, against illegal copying and distribution, for scalably compressed content as for ordinary content. Consumers of scaled content wish to verify that they have received an authentic copy of the original material, just as for ordinary content. Enrichment services, such as subtitles or links to other works by the same artist, should not disappear simply because the content has been scaled to meet bandwidth or resolution requirements. Indeed, the entire variety of digital watermarking applications remain as desirable in scalably compressed content as they are in any other form of content.

However, it is rarely effective to simply apply a traditional watermarking algorithm to scalably compressed content. This because the scaling process, which tailors the content to

suit the user's requirements, involves discarding layers of content. Thus, in the majority of cases, when the scalably compressed content is received, some layers will have been discarded. Any watermark data contained in those layers will have been lost. Depending on the user's requirements, the discarded layers may comprise a substantial majority of the content. As a result, a substantial majority of the watermark data may not be received, causing the watermark detection to fail.

One approach to the watermarking of scalable content has been presented by a number of authors in the emerging field of scalable watermarking [37, 105, 174, 116, 181]. Firstly, the lowest acceptable resolution or quality is determined. Secondly, the maximum level of scaling associated with the lowest acceptable resolution or quality is determined (content scaled beyond this level is considered no longer valuable and thus no longer in need of a watermark). Finally, the watermark is embedded in the portions of the content that are largely unaffected by that level of scaling. This approach is undoubtedly successful in the case of ordinary (non-scalable) compression, where the requirements of all users are assumed to be fairly similar, and for data enrichment watermarks, in which there are no security requirements. However, scalable compression caters for users with widely differing requirements; so what is acceptable to a user with a small screen and poor connection may be completely unacceptable to a user with a large screen and a fast connection. As will be discussed in sections 3.1.1 and 6.3 (pages 52 and 239), restricting embedding to the portions of the content deemed acceptable by low-end users can result in a lack of watermark protection in the content received by high-end users.

An alternative approach focuses on the relative value of the content at different levels of scaling [25, 110, 177, 178, 201]. The full content, containing all layers, is undoubtedly of higher value than a highly scaled version of the content, containing only a few layers, whose resolution or quality has been dramatically reduced to suit a small screen size or a low-bandwidth connection. This suggests that the level of watermark protection required by the full content is higher than that required by highly scaled content. More generally, as the number of layers of received content increases, so should the protection provided by the watermark. To achieve this, the watermark is distributed across all layers of the content. This approach ensures that the content received by high-end users is protected but ignores the problem of watermark loss in the highly scaled content received by low-end users[8].

If the benefits of watermarking are to be successfully brought to scalably compressed content, it will be necessary to use scalable watermarking algorithms specifically developed

[8]Indeed, some proponents of this approach have suggested [178] that, although detection in highly-scaled content is desirable, if the watermark detection result does not exceed the desired threshold then acceptance of the content should be delayed until more data has been received. However, requiring the user to download otherwise unnecessary content simply for the purposes of detecting a watermark seems rather incompatible with the ideals of scalable compression

for the task. This thesis proposes that a scalable watermarking algorithm should fulfil the aims of both the aforementioned approaches. That is, the watermark should be detectable in the most highly scaled content but should also provide improved protection as more of the content is received. Furthermore, it contends that, at least in the case of images, it is possible to produce a scalable watermarking algorithm, achieving the aims of both approaches, that supports both resolution and quality scalable compression in a single watermark.

1.3 Objectives, Structure and Scope

The aim of this thesis is to develop a scalable watermarking algorithm that supports both resolution and quality scalable compression. Although many of the concepts considered will apply to scalable watermarking in general, the focus of this thesis will be almost exclusively on scalable watermarking for images, and all algorithms designed and tested in this thesis will be image watermarking algorithms.

It is clearly desirable to have watermarking algorithms that support both resolution and quality scalability, as this will allow watermarked images to be successfully scaled to meet both display and bandwidth constraints. However, previous scalable image watermarking algorithms have concentrated on providing only one type of scalability. Wang and Kuo [201] and, later, Su, Wang and Kuo [177, 178] considered only quality scalability, as did Lin and Chang [107], Ho and Li [70] and Maeno et al. [116]. On the other hand, Steinder, Iren and Amer [174], Chen and Chen [25] and Guo and Georganas [61] considered resolution scalability but did not consider quality scalability. This is presumably because providing both resolution and quality scalability is substantially more difficult than providing only a single type (for reasons that are examined in detail in chapter 4). Indeed, prior to the work in this thesis[9], the only attempt to provide both types was by Sun, Chang et al. [181]. This attempt met with limited success. Although their method supported either resolution or quality scalability, such support was not simultaneous. They suggested that good performance could be achieved provided the type of scalability was fixed, to either resolution or quality, at the time of embedding.

The most effective approach to constructing a resolution and quality scalable watermarking algorithm is to design the algorithm expressly for integration with a particular compression system. This is because the scalability of a watermark in any given application depends on the compatibility between the watermark design and the particular compression system being used in that application. That is, a scalable watermark will perform

[9]The algorithms of Lu et al. [114], Danyali and Amiri [37] and Meerwald and Uhl [123] certainly consider both resolution and quality scalability. However, these publications are all subsequent to the associated work in chapters 3 and 4 of this thesis.

best if the compression system used is the one for which it was designed, and will decrease as the compression system diverges from the one for which it was designed. Of course, for such an approach to be useful in this thesis, the particular compression system must support both resolution and quality scalability. The obvious choice for such a compression system is JPEG2000, which is a relatively recent, internationally recognized, standard for image compression that was developed specifically to provide both resolution and quality scalability. JPEG2000 will be used throughout this thesis in the design, and also the testing, of scalable watermarking algorithms. Chapter 2 provides some background concepts and terminology for the two major aspects of scalable watermarking for images, watermarking and scalable compression. It also provides specific details on JPEG2000 that are relevant to later chapters.

In order to demonstrate that a scalable watermarking algorithm has been successfully developed, it must be possible to evaluate the scalability of that algorithm. This requires both a solid definition of what scalable watermarking is and a mechanism for evaluating watermark scalability. While there have been many papers proposing new scalable watermarking algorithms, there has been relatively little discussion on what constitutes a scalable watermarking algorithm, and still less on how to evaluate one. Chapter 3 provides a formal definition of a scalable watermarking algorithm, discussing the properties it should possess, how these properties might be measured and how they have been viewed in the scalable watermarking literature. The remainder of the chapter focuses on the evaluation of watermarking algorithms, describing the experimental and analytical framework used to evaluate watermark scalability in the chapters that follow.

Chapters 4, 5 and 6 focus on providing concrete designs for resolution and quality scalable watermarking algorithms, and evaluating their performance. In particular, two different types of watermarking algorithm are developed, non-blind and blind. A non-blind algorithm is one in which the original image is required as part of the watermark detection process; this causes the watermark extraction to be relatively straightforward, so that the bulk of the design work can be focused on achieving resolution and quality scalability simultaneously. A blind algorithm does not require the original image as part of the watermark detection process; this makes it suitable for a wider range of applications, but presents the additional challenge of maintaining watermark synchronization when only a highly scaled image is available at the detector.

A scalable watermark must be able to survive the loss of a substantial amount of image content. This suggests a highly robust watermarking technique may be the most suitable choice for integration with scalable compression. Chapter 4 examines the effects of scalable compression on one of the most popular robust watermarking techniques, spread spectrum watermarking. It considers how a spread spectrum watermarking algorithm

might be altered to enhance its scalability. In particular, three options are considered: adjusting the watermark detection threshold, selecting a different set of coefficients to be watermarked and adapting the watermark according to the human visual system. A non-blind watermarking algorithm is designed using these alterations with the aim of achieving both resolution and quality scalability.

Non-blind watermarks are not suitable for all applications; so, to fully demonstrate that both resolution and quality scalability can be achieved, it is also necessary to consider blind watermarking. Chapter 5 presents a blind watermarking algorithm that is designed to be both resolution and quality scalable. In addition to resolution and quality scalability, an important goal in the design is to maintain an exact match between candidate and extracted watermarks at all levels of scaling. With this exact match property, the algorithm becomes a good candidate for use in scalable image authentication, so a number of experiments are performed to assess the weaknesses of the basic algorithm in an authentication scenario.

The majority of papers on scalable watermarking have presented robust watermarking algorithms, for copyright protection, so it is worth examining the possibility of a scalable, semi-fragile watermark fully. Chapter 6 presents a semi-fragile scalable watermarking algorithm, designed specifically to perform in an authentication scenario. The aim of the design is to improve on that of the previous chapter by enhancing security without compromising scalability. In light of the weaknesses identified in chapter 5, alternative approaches to providing security against counterfeiting attacks, that have appeared in the fragile and semi-fragile watermarking literature, are examined. Previous semi-fragile watermarks designed to tolerate JPEG and JPEG2000 compression only satisfy half of the definition of watermark scalability proposed in chapter 3. This definition is revisited in terms of its relevance to semi-fragile watermarking. Finally, the proposed algorithm is experimentally evaluated to determine whether the algorithm is both secure and scalable.

Chapter 7 provides some concluding remarks. It briefly presents the main contributions of this thesis and the importance of each contribution in the context of previous work. The limitations of the thesis are also presented, along with a number of open problems in the field of scalable watermarking and in watermarking more generally.

For ease of reference, an equation defined in one chapter may be reprinted in a later one. For any such equations, the original equation number is retained and is displayed, in blue, beside the reprinted equation. At many points throughout this thesis, a comprehensive explanation of a procedure, full presentation of experimental results or complete proof of a claimed property is desirable, but would distract the reader from the main concepts being explored. Rather than omit these details, they are provided as a set of appendices (A through E).

Chapter 2

Background

This chapter contains background information and terminology on watermarking, scalable compression and JPEG2000. It is intended to provide an introduction to the general concepts that influence scalable watermarking, and to establish some of the terms and notation used in subsequent chapters. A reader who is familiar with one or all of these topics may wish to skip the associated background sections.

2.1 Watermarking

2.1.1 Definition

Digital watermarking refers to the embedding of a content-related signal, known as a watermark, into digital content by making slight alterations to the content itself. A digital watermarking algorithm X consists of an embedding algorithm $\mathsf{Embed}_{\mathrm{X}}$ and a detection algorithm $\mathsf{Detect}_{\mathrm{X}}$.

The *embedding algorithm* takes an image (or other content) I, embedding key sk_e and, optionally, a message M and some additional parameters Λ, and produces a watermarked image I' by imperceptibly altering the content of I.

$$I'_{\mathrm{X},M,sk_e} = \mathsf{Embed}_{\mathrm{X}}(I, M, sk_e, \Lambda). \tag{2.1}$$

The additional parameters Λ typically at least include an *embedding strength* parameter, which allows the robustness of the watermark to be increased at the cost of increased perceptibility. Parameters controlling the size of the embedded watermark, the locations at which embedding occurs and many other aspects of algorithm behaviour may also be included in Λ.

The *detection algorithm* takes a candidate image I^*, which may or may not have been derived from I'_{X,M,sk_e}, the detection key sk_d and, optionally, the original image I

and/or parameters Λ and outputs either *True* and any accompanying message M if the watermark is detected or *False* if no watermark has been detected.

$$\mathsf{Detect}_{\mathrm{X}}(I^*, I, sk_d, \Lambda) = \begin{cases} \{True, M\} & \text{if the watermark corresponding to key} \\ & sk_d \text{ is detected in } I^* \\ \{False\} & \text{otherwise.} \end{cases} \tag{2.2}$$

Often, particularly for zero-bit watermarking algorithms[1], the detection algorithm decides between the two possible outputs by calculating a *detection statistic* $\gamma = \gamma(I^*, I, sk_d)$. This detection statistic represents the degree of correspondence between the *candidate watermark*, i.e. the watermark corresponding to sk_d , and the *extracted watermark*, i.e. the watermark obtained directly from the image I^*. The detection statistic is then compared to some pre-defined threshold T . The particular threshold used will be application dependent, and is often chosen according to a false positive model (section 4.1, page 89) to fit a given, application specific, allowable rate of error. If the value of the watermark detection statistic exceeds the threshold, $\gamma > T$, the watermark is detected.

The detection algorithm $\mathsf{Detect}_{\mathrm{X}}(I^*, I, sk_d, \Lambda)$ results in one of four outcomes, depending on whether or not the watermark was embedded and detected in the candidate image I^* (table 2.1). A correct result is an output $\{True, M\}$ if I^* is derived from I'_{X,M,sk_e}, or an output $\{False\}$ if I^* is *not* derived from I'_{X,M,sk_e}. These are known as *true positive* and *true negative* results respectively.

Table 2.1: Classification of watermarking outcomes. False positive and false negative outcomes represent detection errors, true positive and true negative outcomes represent correct results.

		Watermark Embedded	
		Yes	**No**
Watermark	**Yes**	True Positive	False Positive
Detected	**No**	False Negative	True Negative

It is important to note that the detection algorithm may not always produce correct results. The candidate image I^* could be any of a vast number of possible images, and may appear to contain the watermark signal purely by chance, resulting in an output of $\{True, M\}$ even though I^* is not derived from I'_{X,M,sk_e}. This is known as a *false positive*

[1] In a zero-bit watermarking algorithm no message M is embedded (section 2.1.2.5, page 19), so the output of detection is simply either $\{True\}$ or $\{False\}$.

result or *false alarm*. The same result might also occur due to deliberate attacks involving unauthorized embedding or counterfeiting of the watermark.

Alternatively, the watermark signal may be too faint to be correctly detected, resulting in an output $\{False\}$ (or $\{True, M'\}$ where $M' \neq M$) even though I^* is derived from I'_{X,M,sk_e}. This is known as a *false negative* result or *missed detection*. There are two main causes of a faint watermark signal. Firstly, having an imperceptible watermark, one which does not noticeably affect the quality of the image, may be more important than ensuring that all watermarks can be detected. In this case, for certain images [2] the watermark will be embedded too weakly in I'_{X,M,sk_e} to allow detection. Secondly, the processing applied to I'_{X,M,sk_e} to derive I^*, may have reduced the strength (or masked the presence) of the watermark signal to such an extent that detection is no longer possible. This may be the result of transmission errors or incidental processing or of deliberate attacks on the image. The expected proportions of false positive and false negative detection errors are important factors in deciding whether a certain algorithm is suitable for a given application.

In most watermarking algorithms, key generation is not explicitly specified and sk_e and sk_d are assumed to be identical seeds for some pseudorandom number generator, which is contained in both the embedding and detection algorithms. In more general terms, there exists a *key generation* algorithm that takes some security parameters ζ and outputs matching embedding and detection keys sk_e and sk_d:

$$\{sk_e, sk_d\} = \mathsf{Generate}_{\mathrm{X}}(\zeta). \tag{2.3}$$

These keys are used to provide security against detection, modification or removal of the watermark by unauthorized parties, who do not possess the appropriate key. As a result, the associated *keyspace* $\mathcal{K}$, which is the set of all embedding and detection key pairs $\{sk_e, sk_d\}$ that may be generated by the algorithm $\mathsf{Generate}_{\mathrm{X}}(\zeta)$, must be sufficiently large that identifying the correct key by exhaustively searching $\mathcal{K}$ is impractical. The secure distribution and management of keys, while critical to the success of many watermarking applications, can generally be considered a separate problem from that of watermarking algorithm design.

2.1.2 Properties

2.1.2.1 Perceptible/Imperceptible

A perceptible watermark is intended to be discernible to any person accessing the content in which it is present, without the aid of a watermark detector. The primary application

[2] The images for which watermark embedding at sufficient strength to be correctly detected causes an unacceptable level of distortion will depend on the watermarking algorithm, message and embedding key, as well as the perceptual distortion constraint for watermark embedding.

for this type of watermark is to include a 'fairly unobtrusive' representation of a company, such as a logo or announcement, within the content. Because the watermark is designed to be perceptible, all content users are made aware that the content originates from, or is owned by, that company without the need for a detector. How unobtrusive a perceptible watermark should be will depend on the application.

By far the majority of watermarking research, including this thesis, deals with imperceptible watermarks. This type of watermark is designed so that it is not discernible simply from viewing the content, but only through the use of a detection algorithm. An imperceptible watermark is less damaging to the content, from a user's perspective, than the perceptible watermark and also allows for a greater variety of applications.

A watermark is *imperceptible* if the distortion caused by the embedding algorithm is not noticeable to a human observer.

Imperceptibility is achieved through the use of a *perceptual model*, which is an abstraction of a human perceptual system[3] and is used to describe the perceptibility of the changes to the original content due to watermark embedding.

A *distortion measure* [4] D' quantifies the amount of distortion caused by watermark embedding in accordance with some perceptual model. Let T' denote a threshold indicating an acceptable level of watermarking induced distortion. A watermark is imperceptible in an image $I' = \mathsf{Embed}(I, M, sk_e, \Lambda)$ if

$$D'(I, I') < T' \tag{2.4}$$

The threshold T' should be close to zero, but its precise value depends not only on the chosen distortion measure but also on the application. Medical or archival applications may have strict distortion limits, while applications such as surveillance or movie dailies may allow increased perceptible distortion in order to obtain better robustness and security.

Imperceptibility is also known as *transparency*, and other domain-specific terms such as *visible* and *invisible* or *audible* and *inaudible* are common, for perceptible and imperceptible respectively.

2.1.2.2 Blind/Non-Blind

A *blind* watermarking algorithm is one in which the original image I is not required as input to the detection algorithm. This property is important in situations where the watermark is intended to be detectable by parties who will not have access to the original image. This includes applications such as authentication and reconstruction, in which

[3] Human visual system models are the relevant perceptual models for images. Which perceptual system is abstracted depends on how the watermarked signal is consumed. Watermarking music or speech will require auditory system models, while watermarked video may use both.

[4] See section 3.2.3.1 (page 69) for a discussion of perceptual distortion measurement.

access to an authentic original image obviates the need for a watermark altogether, as well as applications in device control or labelling, where providing a multitude of detectors with access to the original image is impractical for reasons of security or bandwidth.

A *non-blind* watermarking algorithm is one in which the original image I is required as input to the detection algorithm. Non-blind algorithms lack the flexibility of blind algorithms, in that they are rarely useful in applications where the watermark should be detectable by many parties other than the copyright owner. However, in situations where the detection is performed by the copyright owner or where obtaining the original image is practical, non-blind algorithms may offer better robustness or faster embedding times and the original image may act as an additional secret key, thereby improving security.

Blind watermarking has also been called *oblivious* watermarking, and even *public* watermarking due to its use in applications where the detector is public. Similarly, non-blind watermarking has been called *non-oblivious*, and *private* watermarking. The terms public and private watermarking are best avoided, however, as they can easily be confused with asymmetric and symmetric watermarking (public-key and private-key watermarking). Non-blind watermark detection is also known as *informed* detection.

2.1.2.3 Robust/Fragile/Semi-Fragile

A *robust* watermark is one which is designed to be resistant to manipulations of the content. That is, a robust watermark can still be detected after the content has undergone *processing*, such as lossy compression, resampling, cropping etc.

The *robustness* of a watermarking algorithm X with respect to processing F can be defined as the probability, taken over all original images I, messages M and key pairs $\{sk_e, sk_d\}$, that applying processing F to the watermarked image, (producing a processed watermarked image $I^F_{\mathrm{X},M,sk_e} = F(I'_{\mathrm{X},M,sk_e})$) does not affect the watermark detection result

$$P(\mathsf{Detect}_{\mathrm{X}}(I^F_{\mathrm{X},M,sk_e}, I, sk_d) = \mathsf{Detect}_{\mathrm{X}}(I'_{\mathrm{X},M,sk_e}, I, sk_d)). \tag{2.5}$$

Although it can be applied to any form of processing, the term robustness is most frequently used when referring to processing that occurs in the usual course of the content life-cycle, between the time of embedding and subsequent detection. Robustness to processing that occurs as part of a deliberate attempt to remove the watermark may also be referred to as *security*, specifically security against unauthorized removal.

Robust watermarking algorithms are designed specifically for robustness, and are preferred in the majority of watermarking applications. Although, ideally, robust watermarking algorithms would be robust to any processing, in practice a watermark is not expected to be robust to all possible choices of F. Even where robustness to many different types of processing is required, the level of processing is generally limited so as not to exceed a

threshold T^F that denotes an *acceptable level of processing induced distortion*

$$D^F(I'_{\mathrm{X},M,sk_e}, I^F_{\mathrm{X},M,sk_e}) < T^F, \tag{2.6}$$

as measured by some appropriate (section 3.2.3.2, page 72) distortion measure D^F.

The value of T^F for a given type of processing depends on the application, and typically represents either the expected processing during general use or a point beyond which the image is no longer of sufficient value that detection is deemed important. Note that this distortion threshold may well be greater than the distortion threshold T' allowed for watermark embedding.

A *fragile* watermark is one in which any manipulations to the content will damage the watermark. This does not necessarily make the fragile watermark inferior to the robust watermark, as different applications will demand different amounts of robustness or fragility.

The *fragility* of a watermarking algorithm X with respect to processing F can be defined as the probability, taken over all original images I, messages M and key pairs $\{sk_e, sk_d\}$, that applying processing F to the watermarked image will remove or alter the watermark

$$P\left(\mathsf{Detect}_{\mathrm{X}}(I^F_{\mathrm{X},M,sk_e}, I, sk_d) = \begin{cases} \{False\} \\ \{True, M'\} \quad \text{where } M' \neq M \end{cases}\right). \tag{2.7}$$

Fragile watermarking algorithms are designed specifically for fragility, are generally used for content authentication purposes, so that the damage to the watermark allows the changes in content to be detected. Note that a fragile watermarking algorithm is sensitive to *any* change in the content, including both malicious and non-malicious changes.

Semi-fragile watermarks are designed to be fragile with respect to some changes but to tolerate other changes; for example they may be robust to compression but will detect malicious tampering. This can be achieved by carefully designing the watermark to be robust to specific allowed manipulations. Alternatively, it can be achieved by using only the most visually significant image information to construct the watermark, as most allowed processing will preserve the most visually significant image information.

2.1.2.4 Symmetric/Asymmetric

A *symmetric* (or private-key) watermarking algorithm is one in which the embedding and detection keys are identical $sk_e = sk_d$. The vast majority of watermarking algorithms are of this type. Because knowledge of the detection key implies knowledge of the embedding key, and hence the ability to embed the watermark, access to the detection algorithm is often restricted to trusted parties. However, the detector may be distributed more widely

than this in applications, such as labelling, where the watermark is primarily designed to add value for the consumer, who therefore has little to gain by attacking the watermark.

An *asymmetric* (or public-key) watermarking algorithm is one in which the embedding and detection keys are different $sk_e \neq sk_d$, and access to the other parameters Λ (and, if the algorithm is non-blind, the original image I) that are required for detection is provided for any detector. Such algorithms are designed to allow detection by many but to prevent embedding by anyone who does not have access to embedding key. Asymmetric algorithms are typically more complex and/or time consuming than their symmetric counterparts, and the widespread availability of the detector may encourage sensitivity or gradient descent attacks.

2.1.2.5 Payload and Capacity

The *payload* of a watermarking algorithm is the size of the message M, measured in bits, embedded using the watermark; it may also be expressed relative to the size of the content or as a rate, rather than in absolute terms.

Many watermarking algorithms are developed to embed zero-bit watermarks. In *zero-bit* watermarking systems no message is embedded and the detection algorithm only detects the existence of the watermark.

The payload may be increased at the expense of reduced robustness or increased perceptibility. The *capacity* of a watermarking algorithm is the maximum possible payload that can be reliably embedded and detected; it may also be expressed relative to the size of the content or as a rate, rather than in absolute terms. Note that the capacity depends not only on the algorithm's structure but also on application based constraints such as imperceptibility and robustness.

2.1.3 Applications of Watermarking Algorithms

2.1.3.1 Proof of Ownership

Proof of ownership is perhaps the most well known of all watermarking applications and is also termed *copyright protection*. In proof of ownership, the watermark contains information that can be used to establish the originator or copyright holder of the content. The presence of such a watermark in an unauthorized copy or derivative of the content may be used as evidence of copyright infringement in court.

Proof of ownership watermarking is not designed to directly prevent the violation of copyright, but may allow the copyright owner to gain compensation after violations have occurred, and may also act as a deterrent to unauthorized copying. Proof of ownership watermarks are not necessarily required to directly identify the correct copyright owner

from amongst all possible individuals (although this may be desirable), but instead allow the copyright holder to prove ownership of the content when involved in a dispute.

Proof of ownership is a robust watermarking application, in which security against unauthorized embedding, modification and removal is critical. Furthermore, they must be secure against the invertability attack, as described by Craver et al. [34]; that is, it should be infeasible for an unauthorized user to generate a fake original image such that watermark detection, using the unauthorized content and fake original, indicates the unauthorized user to be the copyright holder.

2.1.3.2 Fingerprinting

In fingerprinting, the watermark contains information which can be used to identify the customer or legitimate user of the content. The presence of such a watermark in an unauthorized copy of the content may be used to determine the legitimate user from whom the copy was ultimately obtained. That is, it allows the identification of the point in the distribution system at which the content 'leaks' from an authorized user to an unauthorized user. This identification process is known as *traitor tracing*, and the legitimate user from whom the content was obtained is known as a *traitor*. Note that the traitor may be actively responsible for the leak or may merely be responsible for inadequately protecting the content, for example a cinema failing to notice and prevent illegal filming during a movie presentation.

Fingerprinting may be useful in any situation in which confidentiality or unauthorized distribution is of concern. It is most effective when there is an ongoing relationship between the content provider and user, which the provider may sever if the user is identified as a traitor.

Fingerprinting is a robust watermarking application, and requires security against unauthorized embedding, modification and removal. Of particular concern is security against collusion attacks, in which multiple legitimate copies with different watermarks are combined to produce an unauthorized copy. In the case of collusion, it should be possible to identify at least one (although preferably all) of the traitors, provided less than some pre-defined number of legitimate copies was used. Similarly, content may be watermarked with different fingerprints at multiple points along a distribution chain, and these watermarks should not seriously interfere with each other.

2.1.3.3 Broadcast Monitoring

In broadcast monitoring, the watermark contains information relating to the identity of the content itself, generally in the form of a unique identification code that represents the

individual piece of content (or some predefined class of content to which it belongs). This allows a watermark detector to easily identify that content from amongst other broadcast content and automatically generate usage data. This may be used to (independently) verify that broadcasters are fulfilling their obligations, both to advertisers and content creators, or for timely market research.

Broadcast monitoring is a robust watermark application. Specifically, it requires robustness to any processing expected during the normal distribution process, such as scaling, change of format or transmission errors. Depending on the application, security against unauthorized embedding, modification and removal may also be important.

2.1.3.4 Copy Control

Unlike proof of ownership or fingerprinting applications, in which the watermark becomes useful only *after* a copyright violation has occurred, copy control or *copy protection* watermarks are designed to *prevent* unauthorized copying or use of content. The watermark contains information indicating how the content is allowed to be used, for example "copy freely", "copy once" or "copy never"; the device copying the content first detects the watermark, then performs the requested operations only if they are allowed by the watermark, updating the watermark if necessary.

Note that this application requires the widespread use of compliant devices, which will both detect the watermark and behave honestly, for the watermark to be effective. Thus there must be sufficient incentives for users and device manufacturers to opt for compliant devices over non-compliant devices (which play and copy content regardless of the watermark).

Copy control is primarily a robust watermarking application, although fragile watermarks may be used in some circumstances. It requires security against modification, but does not necessarily require security against both embedding and removal. For example, if the most restrictive behaviour, "copy never", is associated with unwatermarked data, then security against unauthorized removal is not required. Widespread access to watermark detectors opens the possibility of sensitivity attacks, in which the attacker uses the detector output to guide a sequence of modifications.

Copy control is the best known application in the broader category of applications known as *device control*, in which the watermark information is used to alter the behaviour of the receiving device. Other applications have included the provision of extra functionality for new devices, while maintaining the content format for backward compatibility with old devices, differentiation of services for different customers using the same device, or even interactive toys that respond to watermarks in television signals.

2.1.3.5 Authentication

In authentication applications, the watermark typically consists of information derived directly from the content, such as a set of significant edges, local averages or a hash of the content. The watermark is used to determine whether or not the content has been modified and, ideally, to indicate which parts of the content have been altered. Other information, such as a timestamp and the identity and location of the recording device, may also be included in an authentication watermark.

Authentication is primarily a fragile (or semi-fragile) watermarking application, as the watermark need only remain intact when the content has not been modified, and the damage to parts of the watermark may aid in identifying the tampered regions. If imperceptibility requirements allow, the presence of a robust watermark containing the most important content data may enable partial reconstruction of the tampered regions. Also, the detector must be able to function without the original content (as authentication is only necessary when the original is unknown or unverified).

Security against unauthorized removal is not important for authentication watermarks, as the *presence* of the watermark is required for the content to be authenticated. Security against unauthorized embedding is critical, however, to prevent the authentication of arbitrary unauthorized content. Other attacks that apply to authentication watermarks are discussed in sections 5.2.3 and 6.3 (pages 192 and 239).

2.1.3.6 Data Enrichment

In data enrichment applications, the watermark contains content-related information that may be considered valuable to the user, such as subtitles, scene descriptions, labels, or links to websites offering additional content. By using a watermark to store this information, instead of a header or separate file, the information cannot be accidentally separated from the image, although a detector will generally be required to access it.

Most watermarking applications are designed to protect against the unauthorized distribution or modification of content, so the user may want to circumvent them. Data enrichment watermarks, on the other hand, are beneficial to the user and thus require little to no security. Data enrichment is typically a robust watermarking application, as resistance to incidental processing is desirable. In circumstances where no processing is expected, however, a higher capacity fragile watermark may be preferable.

2.1.3.7 Steganography

Steganography is the embedding of a signal in such a way that its existence can not be detected other than by the intended receiver. Both watermarking and steganography

involve the embedding of an imperceptible signal for subsequent detection and thus certain watermarking algorithms may also be used for steganography. However, the disciplines of steganography and watermarking differ in two main ways.

Firstly, steganography demands that the existence of the signal (and not merely the signal itself) be undetectable by unauthorized parties. This requirement is a fundamental aspect of steganography, in which the secrecy of the communication is paramount, but is not necessary in most watermarking applications[5].

Secondly, in steganography the embedded signal need not be in any way related to the content in which it is embedded; it is the signal that is most important and the content is merely a suitable carrier. In watermarking, however, the embedded signal is related to the content, representing the owner, authorized consumer, allowed use or other content-related information; it is the content that is important and the embedded signal is designed to protect or augment that content in some way. Thus, while a steganographic algorithm is potentially free to select some content in which to embed the message, in watermarking the choice of content is already fixed before embedding occurs.

2.1.4 Attacks on Watermarking

Depending upon the intended application, a watermarking algorithm may need to provide security against a variety of attacks. Attacks on watermarking schemes can be divided into three main categories, according to how the watermark detection process is affected. These categories are removal, desynchronization and counterfeiting.[6]

2.1.4.1 Removal

Removal attacks directly prevent watermark detection, by deleting some or all of the watermark from the watermarked image. They are often combined with other removal or desynchronization attacks to minimize the chance that the watermark is still detected.

The most easily applied attacks in this category are *non-geometric signal processing* operations, such as smoothing, sharpening, contrast enhancement and compression. These operations alter the values of the image pixels or coefficients, which may damage the watermark elements that are embedded therein. If this damage is sufficiently severe and widespread, it may be sufficient to remove the watermark. Non-geometric processing is often applied to an image non-maliciously, as a legitimate part of its use. For deliberate

[5]Indeed in certain watermarking applications, such as fingerprinting and proof of ownership, the existence of a watermark within the content is publicized in order to act as a deterrent against unauthorized use.

[6]Note that although several important attacks in each category are provided, this list is not intended to be exhaustive.

watermark removal, it typically forms part of, or is combined with a more sophisticated method.

Estimation and removal attacks use an estimate of the watermark to guide the removal process. This estimate can be obtained from a single watermarked image, using statistical analysis of local image properties [197]. If a watermark detector is available, a *sensitivity attack* [31] can be applied, which repeatedly uses the detector to inform a pixel-wise estimation process. If multiple images (or image blocks) containing the same watermark are available, an *averaging attack* may be applied. This attack uses the average of the images (or blocks) as the watermark estimate.

The *collusion attack* [15] requires a number of users with copies of the same original image, each containing a different watermark. These copies are combined (e.g. by averaging or interlacing) to produce an approximation of the original image in which none of these watermarks can be detected. A conceptually identical attack is applicable to video sequences of very similar frames that contain different watermarks.

2.1.4.2 Desynchronization

Desynchronization attacks do not remove the watermark from the image, but instead render it unreadable.

The simplest of these attacks are *geometric signal processing* operations, such as resizing, rotation and cropping. These attack alter the position of the watermark so that it does not match the position expected by the detector, causing a loss of synchronization. Like their non-geometric counterparts, geometric signal processing operations are often applied legitimately, but may also form part of a deliberate attack.

For global geometric changes it is often possible to identify the correct alignment through exhaustive search or to design a watermark that is invariant to those changes. Thus attacks have been developed which involve many random, local, geometric changes, such as line and column removal [95], or warping attacks[36], making them difficult to identify or anticipate.

To counter these attacks some methods established correct synchronization using an embedded reference template [138]. The removal of the embedded template thus becomes another desynchronization attack, called the *template attack* [69].

The *mosaic attack* [142] is a desynchronization attack which crops the image into a collection of pieces (so that no watermark can be detected in any single piece), which is assembled in the correct configuration on a web page. When the page is viewed, this collection appears identical to the watermarked image in every respect, but a web-crawling detector perceives it as multiple images, each containing no watermark.

2.1.4.3 Counterfeiting

Unlike removal and desynchronization attacks, which attack the watermarked image directly, counterfeiting attacks do not prevent the detection of legitimate watermarks. Instead, they cause the detection of an illicit watermark. Many counterfeiting attacks are employed directly, against watermarking systems for image authentication. However, by attacking the association between the image and the watermark rather than the watermark itself, counterfeiting attacks may also constitute a *protocol attack* on robust schemes.

In an *inversion attack* [34], the attacker takes a legitimate watermarked image (belonging to somebody else) and creates a counterfeit original (e.g. by subtracting his own watermark). When paired with the counterfeit original, the watermarked image will appear to contain the attacker's watermark, When paired with the true original, the watermarked image will appear to contain the legitimate owner's watermark. This forms a protocol attack in proof of ownership scenarios because, with no way to distinguish between the true and counterfeit originals, the watermarking system cannot prove ownership of a disputed image.

In the *copy attack* [96] a watermark is estimated from a watermarked image by subtracting a de-noised version of that image. The estimate is then perceptually shaped and embedded in an arbitrary target image, producing a counterfeit watermarked image that appears to contain a legitimate watermark. The ability to associate another person's watermark with an image is a protocol attack in certain ownership identification and fingerprinting scenarios. This is primarily because, once such an attack has been demonstrated, it becomes plausible to deny any association with an image containing one's watermark.

The copy attack can be considered to be a specific example of a *mark transfer attack*. A mark transfer attack exploits a lack of dependence between a watermarked image and its watermark to dissociate the watermark from the watermarked image and copy it onto a target image. Unlike the copy attack, however, mark transfer attacks are typically applied against fragile watermarks and use knowledge of the watermarking algorithm to ensure that the watermark is transferred undamaged.

The *Holliman-Memon attack* [72] exploits a lack of dependence between blocks of a watermarked image, which allows the blocks to be repositioned without affecting watermark detection. The attacker replaces each block of the watermarked image, with the equivalently watermarked block (from the watermarked image) that most resembles the corresponding block in a target image. Schemes which guard against the basic Holliman-Memon attack by including neighbouring blocks or positional information may still be vulnerable to more sophisticated versions; respectively, the *transplantation attack* [10] and *collage attack* [54]. Mark transfer and Holliman-Memon style attacks are discussed in more detail in section 5.2.3 (page 192) on image tampering.

2.1.5 Watermarking Techniques

Although the details of watermarking algorithms may vary substantially, giving rise to quite different properties, the vast majority of image watermarking algorithms embed directly into either the spatial or a transform domain, using either quantization or spread-spectrum methods.

2.1.5.1 Spatial Domain

A spatial domain watermarking algorithm is one in which the embedding and detection algorithms operate directly on the pixel values of the image. The spatial domain is in some sense the 'natural' domain for images[7]. Many early watermarking algorithms, e.g. [94, 190, 198, 11], were applied in the spatial domain. This practice may still be preferable in cases where compression will not be applied, as it avoids the complexity of converting the image to the transform domain and back.

2.1.5.2 Transform Domain

A transform domain watermarking algorithm is one in which the embedding and detection algorithms operate on coefficients which have been obtained by applying a transform such as a fast Fourier transform (FFT), discrete cosine transform (DCT) or discrete wavelet transform (DWT). It is often chosen over the spatial domain to provide better integration with or improved robustness to image compression algorithms [90, 211, 179], particularly when the domain chosen for watermarking is the same as that used in compression [206, 66]. Furthermore, many transform domains allow easier identification of the perceptually significant image regions, so transform domain embedding has been used to enhance watermark robustness or imperceptibility [30, 93].

The distinction between spatial and transform domains is not always clear-cut; an algorithm which is conceptually described in a transform domain might be applied in the spatial domain to reduce embedding time for uncompressed images, or vice-versa for compressed images. Indeed Hartung and Girod [66] suggested that both spatial and transform domain versions of the embedding algorithm should be produced, to allow interoperability between uncompressed and compressed images.

2.1.5.3 Quantization Watermarking

Quantization watermarking is used for both robust and fragile watermarks, and includes an early method of digital watermarking proposed by Turner [191]. This method, commonly known as *quantize-and-replace*, involves quantizing selected data words to remove

[7] For other types of content the 'natural' domain will be different. For audio, the natural domain is the temporal domain, while video belongs to both spatial and temporal domains.

one or more of the least significant bits and replacing the removed bits with watermark bits. At the (typically blind) detector, the watermark bits can then simply be read, provided the selection procedure used during embedding is known. A simplified version of quantize-and-replace embedding, in which only the least significant bit in each selected word is replaced, is known as *LSB embedding*.[8] One of the earliest image authentication watermarking algorithms [198] is of this type. Because information is contained in the least significant bits, quantize-and-replace embedding is fragile to most forms of processing. It has been widely used, particularly in the areas of image authentication [198, 207, 19] and in steganography [94, 51, 176, 209]. It is also known for its high capacity, and its simplicity allows fast embedding and detection.

Work by Costa [29], in communications theory, proved that if the channel noise is known at the encoder, then the transmission capacity through a noiseless channel is the same as that through a channel with random multivariate Gaussian noise. Although the concept of watermarking as communications with side information [33], specifically using knowledge of the content to modify the embedded watermark for improved detection, had already been considered, Costa's result suggested that it was possible to use this knowledge to eliminate the original content as a source of watermark interference without sacrificing capacity or requiring information about the original content at the detector.

This became more widely known to the watermarking community with the publication of Chen and Wornell's paper [22], proposing *Quantization Index Modulation* (QIM). Chen and Wornell presented a set of algorithms, based on Costa's concept, in which each sample is quantized using a different quantizer depending on the associated watermark bit to be embedded, and the detection algorithm decodes the index of the quantizer that contains the closest match to the quantized value. The simplest of these algorithms was *dither modulation*, which used a single quantizer shifted by a different dither value depending on the value of the watermark bit. QIM offers improved imperceptibility over quantize-and-replace methods while remaining reasonably computationally inexpensive.

A refinement of this technique, named the *scalar Costa scheme* (SCS), was proposed by Eggers et al. [44], in which a fraction of the difference between the original and dither-quantized value is added to the original value, thereby reducing the embedding-induced distortion. The same refinement was proposed by Chen and Wornell [23] as distortion compensated quantization index modulation (DC-QIM).

Chou et al. [26] considered the watermarking problem as a dual to distributed source coding, and thus suggested it could be solved by selecting a channel code and partitioning the codewords into cosets of source codes. While this method is identical to QIM if lattice

[8] Typical examples of LSB embedding replace only the least significant bit; however, some authors also use the term LSB embedding to refer to the more general quantize-and-replace method.

codes are used for the channel (each coset corresponds to a different quantizer), Chou et al. observed that the use of other error correcting codes, such as trellis codes, offered improved robustness, although presumably at the cost of some computational efficiency.

2.1.5.4 Spread Spectrum Watermarking

Spread spectrum watermarking was developed based on spread spectrum methods in telecommunication [58, 157, 143], which had long been used as a means of achieving signal detectability despite high levels of noise. It appears that the first application of spread spectrum methods to watermarking is in the paper by Tirkel et al. [190] who propose a watermark based on the addition of m-sequences, due to their random appearance and autocorrelation properties. Although they also include a similar algorithm using LSB embedding, Tirkel et al. note that the direct addition of the m-sequences to the image pixels results in higher security than the quantization based algorithm.

The most well known version of spread spectrum watermarking, however, is undoubtedly the transform domain algorithm described in the paper by Cox et al. [30]. In the embedding algorithm, a zero-mean, unit-variance, Gaussian pseudorandom sequence X of dimensionality[9] n is multiplied by a global strength α and added to the n highest magnitude AC DCT coefficients in a greyscale image. The detection algorithm uses the original and received images and the inverse of the embedding formula to extract a sequence Y, which is then compared to the candidate sequence X using a similarity measure $SIM(X,Y) = \frac{X \cdot Y}{\sqrt{Y \cdot Y}}$. If the similarity between the candidate and extracted vectors exceeds a threshold T, the watermark is detected.

A large proportion of all watermarking algorithms are spread spectrum based, e.g. [25, 84, 121, 123, 6]. Spread spectrum watermarking algorithms often favour non-blind detection, as do both the Tirkel et al. [190] and Cox et al. [30] algorithms, as this avoids detection problems caused by interference from the host content. Although techniques such as QIM are superior in terms of host-interference, spread spectrum techniques offer superior robustness to very high levels of processing induced distortion [44]. Furthermore, work by Koval et al. [92] suggests that correctly adapting the spread spectrum watermark to the host content will result in superior robustness at more moderate distortion levels, although QIM remains superior at low levels of processing induced distortion.

Finally, there are watermarking algorithms which draw from both the quantization based and spread spectrum techniques. These include Chen and Wornell's spread transform dither modulation (STDM) [23], a hybrid between spread spectrum and QIM, in which dither modulation is applied to a pseudorandom projection of multiple coefficients;

[9] While Cox et al. refer to n as the *length* of the watermark, the term *dimensionality* is preferred in this thesis as it avoids ambiguity between this and the watermark vector's magnitude, for which the term length is also used.

and the "dirty paper trellis watermarking" [126] of Miller et al., a hybrid between spread spectrum and trellis code techniques, which uses a modified trellis to construct multiple spread spectrum coded versions of the message, selecting the version which best matches the host content.

2.2 Scalable Compression

2.2.1 Compression

2.2.1.1 Definition

A compression system X consists of a compression algorithm $\mathsf{Compress}_{\mathrm{X}}$ and a corresponding decompression algorithm $\mathsf{Decompress}_{\mathrm{X}}$.

The *compression* algorithm $\mathsf{Compress}_{\mathrm{X}}$ takes an image (or other content) I, with perhaps some additional parameters Γ, and represents it in a compressed form I^C that uses fewer bits[10]

$$I^C = \mathsf{Compress}_{\mathrm{X}}(I, \Gamma). \tag{2.8}$$

The corresponding *decompression* algorithm $\mathsf{Decompress}_{\mathrm{X}}$ takes the compressed image I^C and changes it back to its uncompressed form, producing the decompressed image I^D, which is either the uncompressed image I or an approximation thereof

$$I^D = \mathsf{Deompress}_{\mathrm{X}}(I^C) \approx I. \tag{2.9}$$

Compression allows a reduction in requirements for both transmission and storage, while decompression allows the image to be restored to a form suitable for use. Compression may also be referred to as *coding* and decompression as *decoding*, with the combined system being termed a *codec*.

In subsequent chapters, the images referred to are generally in a state of partial compression (or partial decompression); where this is not the case, the state of the image is either irrelevant or clear from context. Thus, for clarity or notation, superscripts indicating the state of compression of an image will typically be omitted.

2.2.1.2 Lossy/Lossless

A compression algorithm may be lossless or lossy. A compression algorithm is *lossless* if its output can be decompressed to produce an exact copy of the uncompressed image.

[10]Note that this need only apply in general, there may exist particular combinations of compression algorithm and content for which the "compressed" version uses more bits than the uncompressed version. This situation is unlikely to occur for image content, however, due to the high level of redundancy in typical images.

That is,

$$\forall I, \mathsf{Decompress}_{\mathrm{X}}(\mathrm{Compress}_{\mathrm{X}}(I, \Gamma)) = I \tag{2.10}$$

Lossless compression only alters the format of the image and has no effect on the image itself.

A compression algorithm is *lossy* if it is not lossless. Lossy compression algorithms typically offer higher compression than lossless algorithms; however, this is achieved by not only altering the format of the content but also discarding some parts, resulting in a certain loss of quality. They are designed, using implicit or explicit models of the human visual system, so that the discarded parts are of minimal visual importance and the resulting perceptual distortion is minimized.

For types of image where the accurate depiction of small or faint details is important, such as medical or astronomical images, lossless compression is preferred. For the vast majority of images, however, a certain amount distortion can be tolerated without affecting the practical or aesthetic value of the image, and lossy compression is preferred. Some compression systems provide either lossy or lossless compression, with the desired type being specified via the input parameters.

2.2.1.3 Rate Distortion

An important problem in lossy compression is the tradeoff between the gain in compression and the loss of quality, which both result from discarding information. Most lossy image compression algorithms allow this tradeoff to be controlled via the input parameters. This can be done by specifying the amount of compression, for example using a target *compression ratio*, which is the number of bits in the uncompressed image relative to the number of bits in the compressed image, or a target *bit rate*, which is the number of bits in the compressed image relative to the number of pixels or samples. Alternatively, it can be done by specifying a target image quality, typically using a numerical or categorical indicator.

Ideally, the compressed image will be as small as possible, have minimal bit rate, given a particular level of allowed distortion. *Rate distortion theory* attempts to determine, based on a statistical model of the source data, the minimum possible bit rate as a function of the allowed distortion, thus providing a lower bound on the practical compression performance of a lossy system.

Rate control is the process of determining, for a given compression system, which parts of the image should be included, and which discarded, in an attempt to approach this lower bound at the target bit rate (or quality). While a number of ad hoc rate control methods exist, *rate distortion optimization* techniques are used to systematically

determine a solution to this problem, based on the number of bits required to encode each part and the reduction in distortion obtained by its inclusion. An interesting review of rate distortion theory and rate distortion optimization, along with a description of the Lagrange multiplier and dynamic programming techniques, can be found in the paper by Ortega and Ramchandran[133].

2.2.1.4 Image Compression Techniques

Images typically contain high amounts of redundancy in their uncompressed form, so compression is particularly effective for this type of content. The techniques used in image compression are divided here into three main areas: entropy coding, decorrelation and quantization, which can be characterized by the type of redundancy that they typically remove.

2.2.1.4.1 Entropy Coding

Entropy coding techniques replace the symbols used to represent the content in such a way that more frequently occurring symbols, or groups of symbols, are represented using shorter codewords, while those which occur less frequently are assigned longer codewords. They are lossless compression methods which remove non-semantic (lexicographic and/or syntactic) redundancy, dealing specifically with representation rather than meaning, and thus may be applied to any type of data.

The name "entropy coding" derives from the paper [165] by Shannon, which laid the foundation for the field of information theory, in which he defines the concept of the *entropy* $H = -\sum_{i=1}^{n} p_i \log p_i$ of a set of probabilities $\{p_1, ..., p_n\}$ and presents "the fundamental theorem for a noiseless channel"[11]. The proof of this theorem includes an entropy coding method developed by Shannon, in which it is explicitly stated that messages of high probability are represented by short codes and those of low probability are represented by long codes. It also includes a description of a similar method, which was independently developed by Fano and later published in a technical report [46], that is known as Shannon-Fano coding.

Perhaps the first widely used entropy code was Morse code, which was used for telegraphy and predates Shannon's paper by over a century. It encodes the letters a–z and the digits 0–9, into a series of dots, dashes and spaces. The length of the codeword assigned to each symbol was determined based on its frequency of occurrence within English text,

[11]The fundamental theorem for a noiseless channel states that it is possible to encode the output of a source with entropy H(bits per symbol) so as to achieve an average transmission rate arbitrarily close to, but not greater than, $\frac{C}{H}$(symbols per second), over a noiseless channel with capacity C(bits per second).

which was estimated using the relative number of pieces representing that symbol in a printer's type box [144].

Huffman coding is a well known entropy coding method that is guaranteed to be optimal [74], in the sense that no other encoding of individual input symbols to binary strings can achieve better average codeword length, provided the symbol probabilities are known. Adaptive variants may be used if probabilities are poorly estimated or non-stationary. However, because the minimum length of any Huffman codeword is a single bit, it cannot achieve compression rates of less than one bit per symbol without appropriately grouping the symbols and determining the multi-symbol probabilities.

Thus, for highly skewed probability distributions, methods that ordinarily encode a variable number of input symbols to a single codeword are often preferred. These include simple methods which replace contiguous areas of the same symbol, such as run length encoding and constant area coding; dictionary based methods such as Lempel-Ziv [214] (LZ), and Lempel-Ziv-Welch [205] (LZW) and also arithmetic coding [151] and its more efficient and context-adaptive[12] successors such as MQ coding [80], used in JPEG2000.

Although, for simple compression algorithms, entropy coding may be applied directly to each pixel of an image, it is more usually applied as the final stage of compression.

2.2.1.4.2 Decorrelation

Decorrelation techniques transform the input data samples so that output coefficients are, as nearly as possible, statistically independent. They may be lossless or lossy, depending on the specifics of the transform used and the arithmetic precision of the implementation. Because they require ordinal or numerical input, unlike entropy coding, they cannot be applied to arbitrary data; however, they are still widely used in image, audio and video compression. While they may potentially target any type of inter-sample correlation, in image compression they are primarily used to remove spatial redundancy, amongst pixels within a region, and also colour redundancy, amongst the red, green and blue (RGB) components of each pixel.

Some decorrelation techniques, such as predictive coding, in which a prediction is made based on previous samples and only the (smaller) error between the current sample and its predicted value is recorded, can achieve compression directly. However, because entropy coding is simplified when the interdependencies amongst the symbols to be coded are low, decorrelation techniques are often used to reduce these interdependencies with a view to subsequent entropy coding.

[12] In *context-adaptive* arithmetic coders, the probability model of a given symbol is selected based on the values of nearby symbols (context) and is adjusted according to the frequencies of previously encoded symbols (adaptive).

A wide variety of transforms have been proposed to remove spatial redundancy.[13] They trace back to the work of Fourier in 1807, where he found that any periodic function could be represented as a sum of sine and cosine functions at various frequencies. In the early 1900s, mathematicians including Haar, Rademacher and Walsh designed other families of functions that could be used in a similar manner [4]. However, it was not until 1965 that an efficient algorithm for applying a discrete Fourier transform, the fast Fourier transform (FFT), was popularized by Cooley and Tukey [27], giving rise to research on many other transforms, such as the highly-decorrelating but expensive Karhunen-Loeve (KLT), the discrete cosine (DCT), and the Walsh-Hadamard (WHT) transforms. This was followed by the fundamental work on wavelet analysis and the discrete wavelet transform (DWT), which better captures localized signal features, beginning with Morlet et al. [128] in 1982, through Mallat [119], Meyer [125] and Daubechies [39]. Fractal compression [83], which uses portions of the image itself to form the basis functions, is a relatively recent technique that is also of this type; it suffers a high encoding cost due to having to determine the basis functions individually for each image.

The red, green and blue components of a single pixel are generally similar in value, so transforms are often applied to RGB images to remove this colour redundancy. While some spatial transforms have been used to remove colour redundancy [64], and many colour spaces have been developed for reasons other than compression, the majority of colour transforms used in compression are luminance-chrominance (opponent) colour transforms, designed specifically for compressed video or images. The earliest such transforms were YUV and YIQ , used for PAL/SECAM and NTSC colour television respectively. The Y component represents the grey-scale, or luminance information[14] and provided backward compatibility with black-and-white television. The chrominance components U and V have the advantage of being easily computed from the gamma-adjusted R and B components (by subtracting Y and multiplying by a constant). The chrominance components I and Q represent orange and purple, which are less computationally efficient, for a better representation of flesh tones [82]. International standards BT.601 and BT.709 followed YUV, providing two different YC_bC_r transforms, optimized for different viewing conditions. The BT.709 YCbCr transform, as well as a lossless integer approximation thereof, forms part of the JPEG2000 standard.

Effective decorrelation is not the only factor in designing a decorrelating transform; indeed many of the above transforms have focused on other factors. The cost of encoding

[13] For further information on many of these transforms and their history, the reader may refer to books on image compression or processing such as that of Shi and Sun [168] or of Gonzalez and Woods [59].

[14] The luminance Y in YUV and YIQ is subtly different from the luminance defined in the 1931 XYZ colour model, developed by the Commission Internationale de l'Eclairage (CIE) to describe human colour perception[148]. The term luma and notation Y' are sometimes used for YUV, and similar colour spaces, in order to emphasize this.

and/or decoding is of great importance, as many applications for image compression are time-sensitive or power-limited. Furthermore, transforms which better match the human visual system are preferred as they facilitate subsequent quantization, either by separating the signal into visually important and negligible parts, or ensuring that similar quantization errors will result in similar perceptual errors.

2.2.1.4.3 Quantization

Because the human visual system is insensitive to slight changes in pixel values, it is possible to discard some information and still obtain an image that is a 'visually lossless' copy of the original. In applications where certain details of the image are not relevant to its meaning or use, even more information may be discarded. Quantization techniques exploit this perceptual and semantic redundancy by reducing the precision of the data, providing a shorter representation at the cost of some distortion. They are inherently lossy techniques, and are responsible for the vast majority of information loss in image compression systems. Quantization may be applied directly, to pixel values, or as part of a more complex compression system (e.g. to transformed coefficient values before entropy coding).

Quantization is the process of mapping a continuous domain to a discrete range (or a finer-granularity discrete domain to a coarser-granularity discrete range). The simplest quantization technique is *scalar quantization*, in which the real line is divided into disjoint intervals and every data value belonging to each interval is transmitted as an integer representing that interval and is reconstructed as a fixed value that is chosen from within the interval. The chosen reconstruction values may be optimized to minimize some distortion measure [113, 122], such as mean squared error (MSE), given the distribution of the data[15]. The compression rate, and the associated distortion, depends largely on the number and lengths of the intervals. In the case of a *uniform scalar quantizer*, consisting of infinitely many intervals of equal length, this length the *quantization step size* parameter may be specified to easily control the amount of distortion.

As with entropy coding, data values need not be quantized individually. *Vector quantization* (VQ) views each set of data values as a single vector in a multi-dimensional space. This space is partitioned into regions, and every vector belonging to each region is represented by a single vector within that region. The compression performance improves with the dimensionality of the vector; however, identifying the region to which a given input vector should be assigned becomes increasingly time consuming. This can be alleviated

[15]It is assumed during optimization that the reconstruction errors, between the data values and corresponding reconstruction values, are independent and identically distributed. Thus transforms for which this assumption is more nearly true are preferred. Similarly, transforms where reconstruction errors are more nearly perceptually uniform allow for simpler distortion measures and simpler quantizers.

by first defining coarse regions and then subdividing those into successively finer regions, as in tree-structured quantization (TSVQ) [55], or by finding the minimum reconstruction error using the Viterbi algorithm, as in trellis-coded quantization [120].

With an appropriate choice of regions, it is possible to discard data values entirely — for example by quantizing all vectors of the form (a, b, c, d) to $(a, 0, c, 0)$. Thus the technique of subsampling, which is often applied to the chrominance components of a colour-transformed image, may be considered a quantization technique (although it is unlikely to be implemented as such). Similarly, completely discarding perceptually unimportant data, such as the highest frequency coefficients after a spatial transformation, may be considered to be quantization.

2.2.2 Scalable Compression

2.2.2.1 Definition

A scalable compression algorithm is one which compresses the image in such a way that the single compressed output facilitates the extraction of multiple different versions of the image. This provides flexibility in that the different versions may be used by devices with differing resources, such as available bandwidth, processing, or display capabilities, or by users with different preferences. Furthemore, such flexibility is provided without any significant overhead (such as recompression or having to decode data which will then be discarded) and without requiring the storage and/or publication of each separate version.

A compression algorithm $\mathsf{Compress}_\mathrm{X}$ is *scalable* if it is possible to produce at least two different decompressed images I_A^D and I_B^D with the aid of some computationally trivial scaling algorithm $\mathsf{Scale}_\mathrm{X}(I^C, \Xi)$

$$I^C = \mathsf{Compress}_\mathrm{X}(I, \Gamma) \tag{2.11a}$$

$$I_A^D = \mathsf{Decompress}_\mathrm{X}(\mathsf{Scale}_\mathrm{X}(I^C, \Xi_A), \Gamma) \approx I \tag{2.11b}$$

$$I_B^D = \mathsf{Decompress}_\mathrm{X}(\mathrm{Scale}_\mathrm{X}(I^C, \Xi_B), \Gamma) \approx I \tag{2.11c}$$

$$I_A^D \neq I_B^D, \tag{2.11d}$$

where Ξ_A and Ξ_B represent different sets of scaling parameters. Each such image, in either compressed or decompressed form, is known as a *scaled image* or *subimage*, $I^{\mathcal{F}}$. One of these subimages is invariably I^C itself (or I^D if decompressed), which will be referred to as the *full image*. Note that in this work the term *scaling* is used to refer to the process of producing a subimage, and not, as is more commonly the case, the resizing of an already decompressed image.

2.2.2.2 Resolution/Quality Scalability

Although the definition of scalability allows for arbitrary subimages, in most algorithms the subimages can be ordered to give successive improvements in one or more aspects. In scalable image compression there are two main aspects: resolution and quality.

A compression algorithm is *resolution scalable* if it facilitates the production of subimages which differ in resolution. Usually each subimage will have twice the horizontal and twice the vertical resolution of the previous subimage. Resolution scalability is particularly suited to devices which differ in terms of display resolution or processing power and to image browsing applications, in which many small images may be viewed but only a few will be required at high resolution. Resolution scalability is also known as spatial scalability.

A compression algorithm is *quality scalable* if it facilitates the production of subimages which differ in quality. Typically the lowest quality subimage will be a coarsely quantized version of the original image, with each subsequent subimage being a more finely quantized version. However, content based algorithms, where, for example, the lowest quality subimage contains only edge features, may also be considered to be quality scalable. Quality scalability is most useful in networks with heterogeneous or varying bandwidth and where the quality of the data is less important than its timely receipt. It also supports progressive transmission for interactive viewing over low-bandwidth channels and dynamic downgrading of image quality for fitting more images in limited storage. Quality scalability is known variously as SNR (signal to noise ratio) scalability, rate scalability or distortion scalability.

2.2.2.3 Scalable Compression Techniques

2.2.2.3.1 Layered Coding

The basis of all scalable image compression techniques is the division of the image into a number of layers. The lowest layer or *base layer* forms a rough approximation of the original image and is also the lowest subimage. The remaining *refinement layers* contain the details of the image; they provide successive improvements to the rough approximation given by the lowest layer but are not particularly useful alone, requiring at least the base layer to form a subimage. Indeed, a strict ordering on the layers is often imposed, and it may not be possible to even decode a particular refinement layer until the base layer and all preceding refinement layers have been decoded. Once the layers have been encoded, the scaling algorithm need only select the layers necessary to achieve the desired resolution, image quality or bit rate and discard the rest.

Specific scalable compression techniques differ primarily in how the layers are defined. Early techniques, by Sloan and Tanimoto [171] and Knowlton [89], use a pyramid structure to define the layers, resulting in resolution scalability. The base of the pyramid structure contains every pixel in the image. At the next level of the pyramid, each $m \times n$ group of pixels is replaced by a single pixel whose value is a function of that group. This continues until the apex of the pyramid, which contains only a single pixel, representing the entire image. Thus the apex of the pyramid is the lowest subimage, the next level of the pyramid corresponds to the next subimage and so on, with base of the pyramid being the full image.

An important characteristic of scalable compression is that each layer should contain only the extra information required to construct the associated subimage. This provides a more compact representation compared to multiple description coding, in which a reasonable approximation of the image can be obtained from any layer. For example, in the method [171], if each pixel in subimage $r-1$ is the average of four pixels in subimage r, then only three of the four pixels in subimage r will be included in layer r, as the value of the fourth pixel can be calculated (from layer r and subimage $r-1$).

Note that finer layer divisions are also used, in which each layer contains the refinements only for a certain image region at a certain level of the pyramid. The resulting layers are less strictly ordered, so a layer corresponding to *any* region in resolution $r+1$ can be decoded provided the layers corresponding to that same region in resolution r have already been decoded. This dramatically increases the number of possible subimages and allows prioritized enhancement of either user- or encoder-selected image regions.

If each sample in an image is considered as a sequence of bits, then the image-sized grid consisting of the κth bit from each sample is called a *bit plane*. Dividing the image into layers based on bit planes and compressing each separately, e.g. [156], results in quality scalability. The most significant bit plane provides a rough approximation of the image at the original resolution, with the addition of each subsequent bit plane providing more precise colour and/or shading[16].

Adaptations of traditional compression techniques have also been applied to achieve scalability. The concept of predictive coding can be adapted to generate layers from a sequence of subimages, by considering each decompressed subimage as a prediction of the following decompressed subimage. Thus the refinement layer required to produce a given subimage is an encoding of the difference between that subimage and the preceding one. This technique appears to have been first applied by Burt and Adelson [18], using subimages formed by repeated applications of a Gaussian filter and is also used in the quality scalable JPEG [77] hierarchical refinement mode. Either resolution or quality

[16] If the precision of each pixel is $\mathcal{P}^c$ bits, then decompressing the most significant bit plane $\mathcal{P}^c - 1$ only is equivalent to decompressing the full image and then quantizing image using a uniform scalar quantizer with step size $2^{\mathcal{P}^c-1}$. Each subsequent bit plane has the effect of halving the quantizer step size.

scalability may be achieved through this technique, depending on how the subimages are formed.

Any frequency transform technique, in which coarse (low-frequency) image features are captured in particular coefficients while finer (higher-frequency) details are captured in others, can be adapted for scalability. This is done by simply grouping the coefficients into separate layers of similar frequency, which allows resolution scalable compression. JPEG spectral selection mode is exactly this technique applied to the block based DCT transform coefficients of ordinary JPEG; yet it is used for quality scalable compression, as the image area is not reduced.

Bit plane based layer formation can be combined with any compression technique in which increasing similarity between unencoded values typically results in an increasing number of identical most significant bits in their encoded values. JPEG successive approximation mode, in which the transform coefficients are divided into layers based on bit planes rather than frequency, is of this type. Successive approximation is an extension of the bit plane method to include any technique in which values are quantized coarsely and the granularity of the quantization is then repeatedly refined. It can be applied not only to transform coding but also to TSVQ and even full search VQ with specially constructed ordered codebooks [21].

2.2.2.3.2 Embedded Coding

A potential problem with a layered approach is that subimages are constructed using an integer number of layers. If the image is divided into only a few layers, it may be difficult to construct a subimage that matches the desired resolution, quality or bit rate. Dividing the image into a greater number of layers allows a larger set of possible subimages and thus a greater degree of control when scaling. However, many of the preceding scalable compression techniques specify only a limited number of layers. Furthermore, the cost of using a layered structure often increases with additional layers, so a scalably compressed subimage with many layers may have a substantially higher bit rate than a non-scalably compressed image at the same level of distortion.

Embedded coding techniques are those for which any prefix of a compressed subimage is also a valid subimage, of lesser or equal quality. They can be thought of as layered coding techniques with fine granularity, so that each bit in the output may be treated as a separate layer. Ideally, as in layered coding, these single-bit layers would be ordered with decreasing visual importance, thereby achieving optimal rate-distortion performance, equivalent to that attainable through non-embedded techniques. In practice, while each additional bit does produce a valid subimage, the division is not quite as fine as one bit per layer, and not every received bit will improve the image more than the following bit.

The concept of an embedded image coder was popularized by Shapiro [166], with his development of the quality scalable EZW (Embedded Zerotree Wavelet) algorithm. EZW uses a wavelet transform but, unlike the subband coders before it, considers the importance of each wavelet coefficient individually rather than considering an entire subband at a time. The coefficients are encoded using successive approximation; however, the coding at each level of precision is further divided into two passes, *dominant* and *subordinate*. The dominant pass describes which coefficients are *significant* at that level of precision, i.e. have magnitudes greater than the current quantizer step size. The subordinate pass provides refinement bits (thereby halving the quantizer step size) for the significant coefficients only. The significance information is coded efficiently by exploiting a hierarchical relationship amongst wavelet coefficients, specifically that if a coefficient is insignificant then its children are also likely to be insignificant, by using only a single symbol to encode an entire *zerotree*, which is an insignificant wavelet coefficient whose entire set of descendants are also insignificant. The most popular embedded coders are also wavelet based, and also compress by entropy coding successive approximations of transform coefficients. They differ from EZW primarily in how the significance and refinement information at various precisions is prioritized.

The hierarchical relationship amongst the wavelet coefficients was further exploited in Said and Pearlman's quality scalable SPIHT (Set Partitioning In Hierarchical Trees) algorithm [155], by modifying the encoding order according to the significance status of coefficients in other subbands. Specifically, the SPIHT encoder, immediately after finding a significant coefficient, encodes the significance status of its four children and the encoder only examines the significance of a coefficient in a detail subband if its parent or grandparent has already been coded as significant. Furthermore, the significance information at a given precision is prioritized over refinement data at the same precision, the reverse of the order used in EZW, providing better rate-distortion characteristics.

The EBCOT (Embedded Block Coding with Optimized Truncation) algorithm [187], by Taubman, further divides the dominant pass into three passes, forward significance, reverse significance and normalization. The forward and reverse significance passes encode the significance of a coefficient only if it is expected to be significant, based on the already-encoded significance of particular neighbouring coefficients, while the normalization pass encodes the significance of all other coefficients. The passes are ordered: forward significance, reverse significance, refinement, normalization, based on their expected importance to the image quality. Unlike earlier coders, EBCOT does not use the hierarchical relationships between wavelet coefficients to improve coding efficiency. Instead, wavelet coefficients within individual subbands are grouped spatially into blocks which are coded

independently using successive approximation. This enables a critical feature: both quality scalability *and* resolution scalability, depending on the arrangement of the passes. Each pass from each block is assigned to a resolution layer, according to its subband, and a quality layer, according to its rate-distortion contribution. The passes may then be ordered by resolution layer and subordered by quality layer, to obtain a resolution scalable image in which each resolution layer is quality scalable, or vice versa, to obtain a quality scalable image in which each quality layer is resolution scalable. A slight variation of EBCOT is used in the JPEG2000 standard.

2.2.3 JPEG2000

2.2.3.1 Overview

The JPEG2000 compression system [78] was intended to be visually superior to the existing JPEG system at low bit rates and to be applicable across a wide variety of applications (including digital photography, medical, remote sensing and archival applications), and heterogeneous devices (including mobile, internet, printing, scanning and facsimile devices) [79]. To achieve this, it was designed to be both resolution and quality scalable.

JPEG2000 is used throughout this thesis, as a current example of a scalable compression system, for testing the scalability of watermarking algorithms and in the design of new scalable watermarking algorithms. Specifically, the JasPer [1] implementation of baseline JPEG2000 is used.

Compression is performed by a JPEG2000 encoder. The baseline encoding process is only informatively described in the standard but can be considered as the sequential application of six distinct algorithms with the addition of a rate control mechanism, which interacts with the three final algorithms (figure 2.1) to adapt the compressed image to the target compression ratio(s).

Decompression is performed by a JPEG2000 decoder. Although it is the decoding process that is normatively specified by the standard, it is conceptually simpler to consider it as the inverse of the encoding process, with either the exact or approximate inverse of each of the algorithms in the encoder being applied in reverse order (figure 2.2).

The following sections contain a description of each of the main algorithms used in JPEG2000; however, many technical details that are not relevant to watermarking have been omitted. A detailed technical description can be obtained from the JPEG2000 standard itself [81] or from the book [188] by Taubman and Marcellin.

To support very large images, JPEG2000 allows the spatial division of an image into rectangular regions called tiles. In the description of the algorithms in the following

sections, the image is assumed to consist of a single tile. For tiled images the algorithms described are applied separately to each tile.

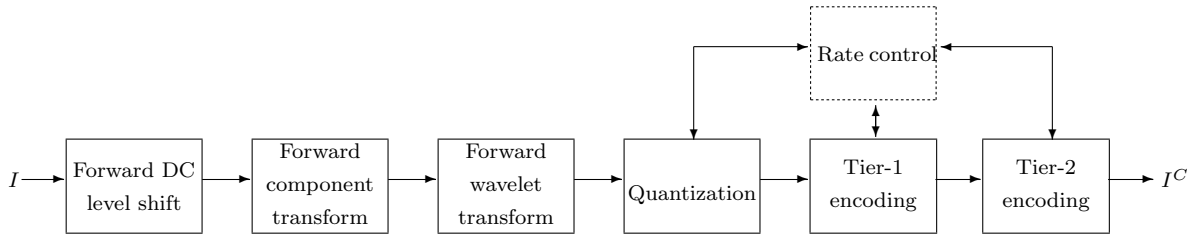


Figure 2.1: Structure of the encoder for the JPEG2000 system. The original image I is compressed to produce I^C.

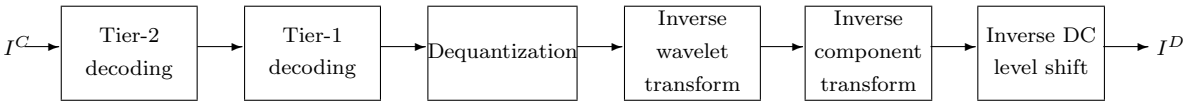


Figure 2.2: Structure of the decoder for the JPEG2000 compression system. The compressed image I^C is decompressed to produce I^D.

2.2.3.2 DC Level Shift

Many image representations use unsigned, rather than signed, sample values (for example, each component of a 24-bpp RGB image is typically in the range 0–255). In such cases, a forward DC level shift is applied to convert the values from unsigned to signed by subtracting $2^{\mathcal{P}_c-1}$ from each sample $I_c(x,y)$ of component c

$$I_c(x,y) = I_c(x,y) - 2^{\mathcal{P}_c-1}, \tag{2.12}$$

where $\mathcal{P}_c$ is the precision in bits of the unsigned values in component c of the uncompressed image I.

The original signedness and precision of each component is recorded in the SIZ marker segment of the encoded image. The inverse DC level shift is applied to each sample in any component I_c^D indicated to have originally been unsigned, and is precisely the inverse of the forward DC level shift

$$I_c^D(x,y) = I_c^D(x,y) + 2^{\mathcal{P}_c-1}. \tag{2.13}$$

2.2.3.3 Component Transform

Each image consists of C components; for typical RGB colour images $C = 3$ but JPEG2000 allows up to $2^{16} - 1$ components. The component transform is a decorrelating transform that is optionally applied to the first three image components. Both an irreversible component transform (ICT) and a reversible component transform (RCT) are defined in baseline JPEG2000, for use with the irreversible and reversible wavelet transforms respectively. The forward ICT can be seen as an RGB $\rightarrow$ $\mathrm{YC_bC_r}$ transform and the inverse ICT as the corresponding $\mathrm{YC_bC_r}$ $\rightarrow$ RGB transform.

The forward and inverse RCTs are approximations, of the forward and inverse ICTs, that have been designed to allow lossless operation.

2.2.3.4 Wavelet Transform

Each component is decomposed into R distinct resolutions, through $R - 1$ applications of a two-dimensional discrete wavelet transform (2D-DWT). This decomposition process is known as a *forward wavelet transform* or *analysis*, while the corresponding reconstruction process is known as an *inverse wavelet transform* or *synthesis*. The number of resolution layers R is chosen by the encoder in the range 1 to 33 inclusive and is recorded[17] in the file header.

Each application of the 2D-DWT decomposes the input into four subbands by applying a lowpass and a highpass analysis filter in the horizontal direction (to the rows) and again in the vertical direction (to the columns), downsampling by a factor of two in each case. Two different sets of forward and inverse wavelet filters are specified in the baseline standard, depending on whether lossy or lossless coding is desired. The Daubechies biorthogonal 9/7 filters [7] are used for the irreversible wavelet transform, while the Le Gall 5/3 filters [99] are used for the reversible wavelet transform.

At the first stage of decomposition, the wavelet transform is applied to the original image I. The resulting subbands are labelled $(R-2)$LL, $(R-2)$HL, $(R-2)$LH and $(R-2)$HH, according to the number of decomposition levels yet to be applied and which filter (lowpass or highpass) has been applied in which direction (horizontal or vertical).

The $(R-2)$LL subband is then further decomposed, using the same transform, to form the $(R-3)$LL, $(R-3)$HL, $(R-3)$LH and $(R-3)$HH subbands, with $(R-3)$LL being an approximation of the full image at one quarter the horizontal and vertical resolution. The process is repeated until the lowest resolution $0LL$ is obtained. The resulting subbands can be seen in figure 2.3. For some images, the number of bit planes required for the wavelet representation may exceed the precision of the original representation, so a small

[17] It is in fact the number of decomposition levels $N_L = R - 1$ that is recorded, in the COD or COC marker segments of the file header.

number of *guard bits* are assigned to each coefficient to catch any magnitude overflow. A different number of guard bits may be used for each component, and is recorded in the QCC or QCD marker segments of the JPEG2000 header.

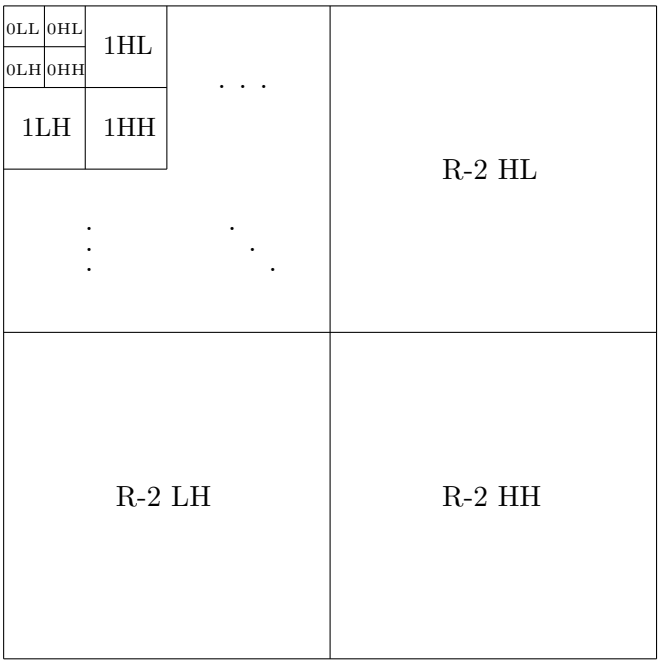


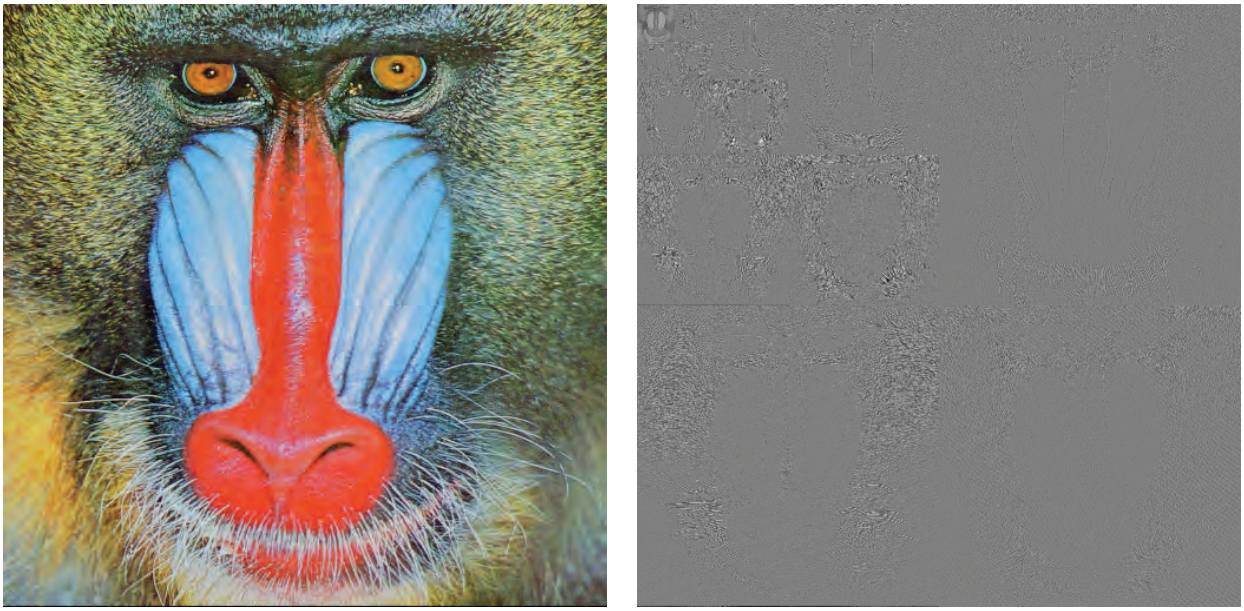
Figure 2.3: Subbands in an R resolution layer wavelet decomposition

Note that, depending on the orientation of the subband, different image features are captured (figure 2.4). The HL subband is horizontally highpass and vertically lowpass filtered, and thus contains the vertical edge information. Similarly, the LH subband contains the horizontal edges and the HH subband contains the diagonal edges. The LL subband is both horizontally and vertically lowpass filtered and is an approximation of the full image at low resolution.

To reconstruct the image the inverse of the above process is performed. Subband 0LL is combined with subbands 0HL, 0LH and 0HH and inverse wavelet transformed to form subband 1LL, which is combined with subbands 1HL, 1LH and 1HH and inverse wavelet transformed to form subband 2LL and so on until the full image $I = (R-1)\text{LL}$ has been reconstructed.

It is this structure that provides the resolution scalability in JPEG2000. To form a resolution r subimage, the wavelet reconstruction process is halted after forming the rLL subband. We say that a subband is in resolution layer r, $0 \leq r < R$ if it is required to form the resolution r subimage, but is not required to form the resolution $r-1$ subimage.

In the resolution r subimage, all coefficients contained within resolution layers greater than r are *completely lost*, while the remaining coefficients are entirely unchanged. This



(a) (b)

Figure 2.4: The Mandrill image (a) and the 5 resolution level wavelet transformed coefficients for the greyscale component of the same image (b)

behaviour is common to other resolution scalable compression techniques, such as pyramid coding and also to JPEG spectral selection.

We use $s = (r, o)$ to denote a specific subband in resolution layer $r \in \{0, 1, \cdots, R-1\}$ with orientation $o \in \{0, 1, 2\}$. The lowest resolution layer contains only the single LL subband $o = 0$; in all other resolutions there are three subbands: the HL subband $o = 0$, the LH subband $o = 1$ and the HH subband $o = 2$. Note that, unlike quality scaling, resolution scaling changes the dimensions of the image. If each component of the original image has dimensions $X \times Y$, then each component of the resolution r subimage has dimensions $X[r] \times Y[r]$ where

$$\begin{aligned} X[r] &= \left\lceil \frac{X}{2^{R-r-1}} \right\rceil \\ Y[r] &= \left\lceil \frac{Y}{2^{R-r-1}} \right\rceil . \end{aligned} \tag{2.14}$$

2.2.3.5 Quantization

In the baseline standard, each coefficient x in each subband s of the wavelet transformed image is quantized to an integer value using a deadzone uniform scalar quantizer according to the formula

$$v = sign(x) \left\lfloor \frac{|x|}{\Delta_s} \right\rfloor, \tag{2.15}$$

yielding a quantized coefficient v.

The sign function is simply plus or minus one, depending on whether x is positive or negative

$$sign() : \mathbb{R} \rightarrow \pm 1, x \mapsto \frac{x}{|x|}. \tag{2.16}$$

The step size Δ_s controls the granularity of the quantizer. For lossless compression, $\Delta_s = 1$, effectively bypassing quantization. For lossy compression, a different value of Δ_s may be chosen by the encoder for each subband s and is recorded in the file header. Each step size Δ_s is specified via an exponent and mantissa (termed ϵ_b and μ_b in [78]), which are recorded, in the QCD or QCC marker segments, either explicitly, for all subbands, or implicitly, based on the recorded exponent and mantissa values of the 0LL subband. For lossless compression only ϵ_b need be specified.

The corresponding reconstruction process converts the quantized coefficient v into a reconstructed wavelet coefficient x^D, which is an approximation of the original wavelet coefficient x, according to the formula

$$x^D = \begin{cases} sign(v)\left(|v| + \mathrm{r}\right)\Delta_s & v \neq 0 \\ 0 & v = 0 \end{cases} \tag{2.17}$$

for lossy reconstruction and, because the above formula is not guaranteed to produce integer values, the formula

$$x^D = \begin{cases} sign(v)\left\lfloor\left(|v| + \mathrm{r}\right)\Delta_s\right\rfloor & v \neq 0 \\ 0 & v = 0 \end{cases} \tag{2.18}$$

for lossless reconstruction, where $\mathrm{r} : 0 \leq \mathrm{r} < 1$ is the coefficient reconstruction parameter chosen by the decoder. The principal advantage of this style of quantization is that the set of quantization interval boundaries for a quantizer with step size $2^m\Delta_s$, where $m \in \mathbb{N}$, is contained entirely within the set of quantization interval boundaries for the quantizer with step size Δ_s (figure 2.5).

This embedded quantizer structure provides quality scalability; the effective quantization step size can be increased by a factor of 2^m, for any coefficient, simply by discarding the m least significant bits of the quantized coefficient magnitude and reconstructing using

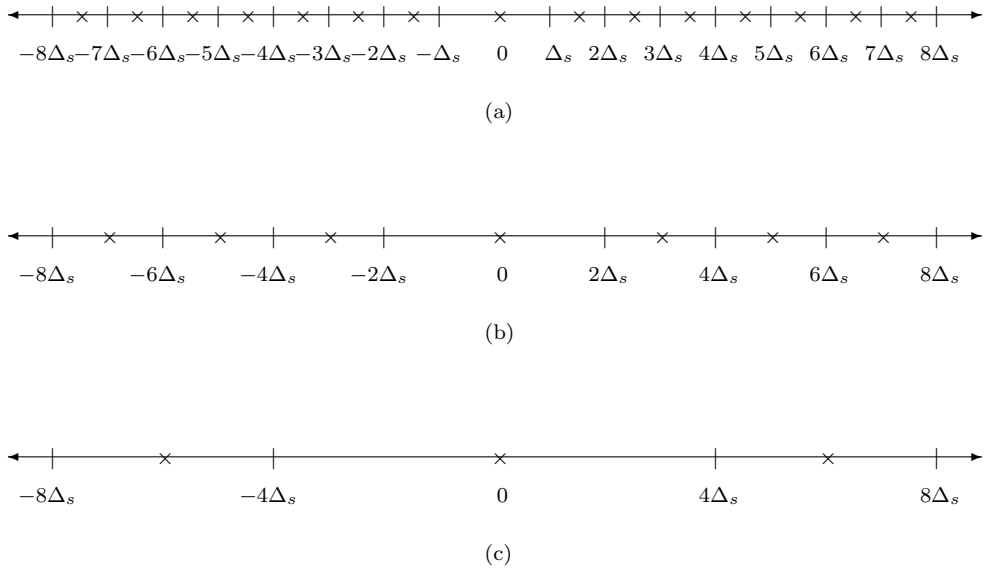


Figure 2.5: Interval boundaries | and reconstruction points × for deadzone uniform scalar quantizers with step sizes (a) Δ_s, (b) $2\Delta_s$ and (c) $4\Delta_s$. The boundaries of the quantization intervals align such that these quantizers form part of an embedded family of quantizers.

the step size $2^m\Delta_s$.

$$x^{\mathcal{Q}} = \begin{cases} sign(v)\left(\left\lfloor \frac{|v|}{2^m} \right\rfloor + \mathrm{r}\right) 2^m \Delta_s & \left\lfloor \frac{|v|}{2^m} \right\rfloor \neq 0 \\ 0 & \left\lfloor \frac{|v|}{2^m} \right\rfloor = 0 \end{cases} \tag{2.19}$$

Thus the step size Δ_s may be set to a small value, to produce high quality images, while the tradeoff between image quality and coding efficiency is managed by selectively discarding different numbers of least significant bits from the quantized coefficients.

2.2.3.5.1 Quantized Coefficients in JasPer

Although the quantized coefficient bits available from a quality scaled subimage are $sign(v)$ and $\left\lfloor \frac{|v|}{2^m} \right\rfloor$, not all JPEG2000 implementations store the quantized coefficient in the form $v^{\mathcal{Q}} = sign(v)\left\lfloor \frac{|v|}{2^m} \right\rfloor$. For the purposes of this thesis, the quality scaled quantized coefficient $v^{\mathcal{Q}}$ is considered to be of the form most readily available within the JasPer decoder [1].

In particular, if $v \in \mathbb{Z}$ is the unscaled coefficient, and the m least significant bits have been discarded due to quality scaling, the corresponding quality scaled coefficient at the

JasPer decoder is

$$v^{\mathcal{Q}} = sign(v) \times \begin{cases} \left\lfloor \frac{|v|}{2^m} \right\rfloor 2^m + \lfloor 2^m \mathrm{r} \rfloor & \left\lfloor \frac{|v|}{2^m} \right\rfloor \neq 0 \\ 0 & \left\lfloor \frac{|v|}{2^m} \right\rfloor = 0. \end{cases} \tag{2.20a}$$

where

$$\mathrm{r} : 0 \leq \mathrm{r} < 1 \tag{2.20b}$$

is the JPEG2000 coefficient reconstruction parameter; r = 1/2 is used in the JasPer decoder. The resulting reconstructed[18] value $x^{\mathcal{Q}}$ is

$$x^{\mathcal{Q}} = v^{\mathcal{Q}} \Delta_s. \tag{2.22}$$

For decoders using other forms some minor adaptation of the watermarking algorithms in chapters 5 and 6 would be necessary.

2.2.3.6 Tier-1 coding

Each resolution subimage is divided into a number of disjoint rectangular blocks called *precincts*, to allow the independent transmission of data relating to particular spatial regions of the image. The nominal width and height of the precincts is chosen by the encoder, and recorded in the file header,[19] but they must be integer powers of two in the range $[2^0, 2^{15}]$ and the width or height of any given precinct p may be less than the nominal width or height, specifically when the nominal precinct extends over the boundaries of the resolution.

The quantized coefficients in each subband are then divided into rectangular regions called *codeblocks*, the nominal codeblock width and height are powers of two in the range $[2^2, 2^{10}]$ and are chosen by the encoder. The same nominal width and height applies for every subband in a given component and is recorded in the file header[20]. The actual width or height of a given codeblock b is restricted by both the subband boundaries and the precinct boundaries. The coefficients in each codeblock are independently coded, using

[18] In the draft standard for JPEG2000 part 1 [78], the reconstructed coefficient value for an irreversible transform is said to be:

$$x^{\mathcal{Q}} = sign(v) \times \begin{cases} (\left\lfloor \frac{|v|}{2^m} \right\rfloor 2^m + 2^m \mathrm{r}) \Delta_s & \left\lfloor \frac{|v|}{2^m} \right\rfloor \neq 0 \\ 0 & \left\lfloor \frac{|v|}{2^m} \right\rfloor = 0, \end{cases} \tag{2.21}$$

where Δ_s is the quantization step size. The reconstruction value r = 1/2 gives the same reconstructed coefficient as the draft standard when $m > 0$, and a reconstructed coefficient which is $\frac{\Delta_s}{2}$ lower in magnitude when $m = 0$.

[19] Only the precinct width and height exponents, denoted PPx and PPy, are recorded in the COD or COC marker segments of the file header.

[20] As with precincts, only the codeblock width and height exponents, denoted xcb and ycb are recorded, in the COD or COC marker segments.

a variation of the fractional bit plane encoding method used in the EBCOT algorithm, to generate a number of passes.

Each bit plane of the coefficient magnitudes for that codeblock is encoded in turn, beginning with the most significant bit plane and proceeding to the least significant bit plane. The sign bit for each coefficient is encoded immediately after its most significant magnitude bit is encoded. The *number of coefficient magnitude bit planes in subband s* will be denoted M_s. In any particular codeblock b, the Z_b most significant of these bit planes consist entirely of zeros. To encode these *all-zero bit planes*, the number Z_b is simply recorded in the header for codeblock b, thus the 'most significant bit plane' referred to above is in fact the most significant magnitude bit plane that is not all zero.

Each of the remaining $M_s - Z_b$ bit planes is coded using three passes: significance, refinement and cleanup. The bit in that bit plane for any given coefficient must be encoded in one of these three passes, with the particular pass depending on the current and predicted significance status of the coefficient.

A coefficient is considered *significant* once its most significant bit has been encoded (or decoded, in the case of the decoder). Initially all coefficients within the codeblock are considered insignificant; coefficients outside the boundaries of the current codeblock are always taken to be insignificant.

The *significance pass* encodes a bit for each coefficient that is currently insignificant but is predicted to become significant, based on the known significance values of its 8-connected neighbours. The *refinement pass* encodes a bit for each coefficient that is significant but did not just become significant during the preceding significance pass. The *cleanup pass* encodes a bit for each coefficient that is neither significant nor predicted to become so. Note that the most significant bit plane is encoded using only the cleanup pass, as all coefficients are initially neither significant nor predicted to become so.

Generally, a context-adaptive arithmetic coder similar to that used in the JBIG standard [80], though with more flexible termination options, is applied to encode each of the passes. Arithmetic coding may also be bypassed (replaced with raw coding) for the significance and refinement passes of all but the first four most significant bit planes.

The passes for a given codeblock are always transmitted in order, starting from the first pass of the most significant bit plane. Thus, when a bit from a given bit plane in a codeblock has been received, all more significant bit planes in that codeblock have already been received.

2.2.3.7 Tier-2 coding

The coded passes are assigned to L quality layers, where L is chosen by the encoder and recorded in the file header. For lossy compression, some of the least important passes may

not be assigned to a quality layer and are instead discarded. A number of consecutive passes from each codeblock is assigned to each quality layer, with this number varying with both the quality layer l and the codeblock b. For any given codeblock, as few as zero or as many as all passes may be assigned to a particular layer l; however, the order of the passes within each codeblock is maintained. That is, if the nth pass of codeblock b appears in quality layer l then the $(n-1)$th pass of codeblock b must have already appeared, in either layer l or one of the preceding layers $0 \ldots l-1$. The particular assignment of coding passes to quality layers, for any given image, is determined using a rate control mechanism chosen by the encoder.

Generally rate-distortion optimization is performed to achieve rate control, using the method of Lagrange multipliers. For each pass of each codeblock, the length (eg. in bytes) of the coded pass and the reduction in distortion (eg. in terms of MSE) is recorded. The set of passes from each codeblock is then divided into as many subsets as possible, such that each subset contains a number of consecutive passes, earlier subsets contain earlier passes, and the *rate-distortion slope* (the average distortion reduction per unit length) for each subset is less than that of preceding subsets. The rate-distortion slope indicates the relative importance of each subset in terms of maximising image quality while minimising the compressed bit-rate. Thus all coding passes belonging to a subset with slope greater than or equal to some value $\frac{1}{\lambda_l}$, i.e. above some level of importance, are assigned to layer l. The value of λ_l is typically determined such that the total length of all passes assigned to layers 0 to l inclusive satisfies, as closely as possible, the target compressed length (often specified in terms of bit-rate or compression ratio) for the quality l subimage, although distortion based targets could also be used.

This process allows the quality scalability of JPEG2000. To obtain a quality l subimage, all quality layers greater than l are discarded, and each coefficient may lose a (different) number of least significant magnitude bits, depending on the number of passes in the discarded layers that involved the coefficient. If the quality scaling is sufficiently severe, all magnitude bits belonging to a given quantized coefficient may be lost. If this occurs, the coefficient will be termed *completely lost*[21] Unlike its resolution scaled counterpart, a completely lost quality scaled coefficient will still be present in the image, but will have the value zero. If fewer than all magnitude bits are lost, the coefficient will be termed *partially lost*.

It is also possible to set the quantization step size for each subband directly, based on the same rate-distortion slope information, to achieve rate control. However, adjusting the quantization step size requires the relatively expensive tier-1 coding stage to be repeated,

[21] Because the sign bit of a coefficient is encoded immediately after the most significant magnitude bit, the sign bit is only transmitted when the most significant magnitude bit becomes known. Thus as soon as all magnitude bits are lost, the sign bit is also lost.

so rate control is best performed by the allocation of passes to layers for subsequent discarding.

Once all layers have been assigned, the passes belonging to all codeblocks and subbands of each tuple (l, r, c, p) of quality layer l, resolution layer r, component c and precinct p are collected into contiguous sequence of data called a *packet*. The passes belonging to an individual packet are arranged in a specific order according to the subband and codeblock to which they belong; however, the packets themselves may be arranged in several different orders.

A particular ordering of packets is known as a *progression*. The baseline standard defines five different progressions: LRCP, RLCP, RPCL, PCRL and CPRL. Only the LRCP progression is described here, although the other progressions are organized in a similar manner. In LRCP progression the packets are arranged primarily according to quality layer, so that all packets in layer $l-1$ appear in the codestream before any packet in layer l, then by resolution, so that all packets in resolution $r-1$ of a given quality layer appear before any packet in resolution r of the same quality layer, then by component and, finally, by precinct (precincts are numbered in raster order).

The LRCP progression facilitates quality scalability, as the quality l subimage can be trivially obtained by truncating the codestream, or halting the transmission, after quality layer l. Similarly, the RLCP and RPCL progressions facilitate resolution scalability. However, header information is sufficient to enable the selective discarding of packets required scalability, regardless of the particular progression chosen during encoding. Thus any JPEG2000 coded image containing multiple resolution and quality layers may be considered both resolution and quality scalable.

Chapter 3

Definition and Evaluation of Scalable Watermarking

This chapter proposes a definition for a scalable watermarking algorithm along with suitable measures of watermark scalability. It also explains the experimental methodology used throughout the thesis, and discusses the need for statistical methods in ensuring reliable comparisons between watermarking algorithms, as in the comparison of section 4.3.2.

3.1 Definition of Scalable Watermarking

The desire to integrate scalable compression and watermarking gives rise to the concept of a scalable watermark.

Consider a scalably compressed image; the full image may be scaled to form a number of subimages.[1] Each of these subimages may itself be of sufficient value to warrant watermark protection. Applying a watermarking algorithm to each individual subimage would provide this protection; however, such an approach would require the separate watermarking and storage and/or publication of each subimage, which defeats the purpose of using scalable compression.

Rather than embedding in all subimages separately using traditional watermarking algorithms, we might use a scalable watermarking algorithm, embedding only in the full image and thus allowing the functionality of scalable compression to be retained. To successfully limit embedding to the full image, the protection provided by that scalable watermark must extend to all sufficiently valuable subimages.

As scalable watermarking currently lacks a formal definition, a qualitative definition is proposed in section 3.1.1. Section 3.1.2 builds upon this definition, developing measures to quantify scalability. Finally, section 3.1.3 describes other works that consider scalable watermarking and their concepts of its definition.

[1] A definition for many of the terms used in this chapter, such as full image, subimage and refinement layer, can be found in the description of scalable compression (section 2.2, page 29).

3.1.1 Proposed Definition

A scalable watermarking algorithm is defined here as follows:

A *scalable watermarking* algorithm consists of combined watermark embedding and detection algorithms that, when used with scalable compression, possess the following two properties:

1. **Detectability**
   The watermark is detectable in any subimage which is of acceptable quality.

2. **Graceful Improvement**
   Additional layers of the scaled content provide reduced error in watermark detection, appropriate to the improved content quality.

Property 1 ensures that the protection provided by a scalable watermark extends to all subimages, assuming those subimages are of 'acceptable quality'. What constitutes acceptable quality is highly subjective, and not merely because the perceived image quality differs according to the observer. For the purposes of this definition, the concept of acceptable quality is related to the commercial value or artistic merit of the subimage, as determined by the watermarker, because it determines when a given subimage is of sufficient value to warrant protection. As a result, the precise definition of 'acceptable quality' will vary, not only across application domains but also for individual authors and distributors.

Furthermore, property 1 suggests that a scalable watermark should be detectable in the lowest layer, as this is the smallest subimage defined during scalable compression. Of course, this assumes that the lowest layer is of acceptable quality—but this assumption is reasonable, because if the lowest layer were not of acceptable quality there would be little purpose in producing it (since all users would then require at least two layers).

Because all subimages contain at least the lowest layer, placing all of the watermark in the lowest layer would satisfy property 1. One may be inclined to conclude, therefore, that placing all of the watermark in the lowest layer, so that the watermark is fully detectable in all subimages, will provide the best scalable watermark; however, this is not the case. For a robust watermark, placing the entire watermark in the lowest layer allows an attacker to completely remove the watermark with relatively little damage to the image, by constructing a replacement lowest layer from the refinement layers as in the paper by Uehara et al. [192]. For a fragile watermark, placing the entire watermark

in the lowest layer allows[2] an attacker to freely use or modify all refinement layers, as is discussed in section 6.3 (page 239).

While this is not a problem for a subimage that contains only the lowest layer, or where the other layers provide little value, the refinement layers of a scalably compressed image can comprise a substantial proportion of its total value and should not go unprotected. Thus, as has been concluded by many previous works (section 3.1.3, page 60), the addition of each refinement layer warrants an increase in watermark protection. This additional protection is ensured by property 2, in requiring that the detection error decrease with each additional layer.

The remaining part of property 2 describes how this additional protection should be related to each layer. It has been suggested, by Lin et al. [110], that the watermark protection should increase in proportion to the increase in the compressed rate caused by each layer. Although that approach would produce similar results, it seems more conceptually appropriate that the protection of each subimage be based on its value to the watermarker, which we may reasonably equate with its quality.

While the level of detectability required to ensure that a watermark retains its functionality will vary with application, the concepts of detectability and graceful improvement are sufficient to define scalability, for at least the types of watermark considered in this thesis. In particular, neither tailoring to match the compression algorithm nor distribution scalability are necessary for this definition.

The concept that the watermark should be specifically tailored to match the compression algorithm has occurred frequently in the literature. This includes techniques such as using the same transform domain, the same coefficient selection method and the same block sizes as the compression algorithm. The concept is not included in the definition provided here. Although such tailoring may well be necessary to achieve the above properties, it does not seem to define a scalable watermarking algorithm. Any watermarking algorithm able to achieve detectability and graceful improvement, without being specifically tailored to match the compression algorithm, would still be suited to the watermarking of scalably compressed content.

The literature also includes the concept of distribution scalability, that a watermark should be able to support increasingly large numbers of concurrent users. This is a desirable property, but it is not in any way related to scalably compressed content. It is scalability in the same sense as in a scalable network, which is an entirely different sense

[2] One might, in fact, restrict the embedding of a fragile watermark to the lowest layer, while still protecting the refinement layers. However, this could only be achieved by using all layers to construct the watermark, meaning that each additional layer must contribute positively to the detection of the watermark. This would still result in reduced detection error with increasing portions of content, and thus property 2 would still hold.

than the one being defined here and so is not included in this definition. Various alternative concepts of what it means for a watermark to be 'scalable' are described in section 3.1.3, page 60.

Note that, as there are different types of scalability in compression, there will also be different types of scalability in watermarking. Thus a *quality scalable* watermark will satisfy the detectability and graceful improvement properties in quality scaled content, while a *resolution scalable* watermark will satisfy them in resolution scaled content.

Similarly, we can consider different types of scalability for each specific scalable compression algorithm, so one may speak of a 'resolution scalable watermarking algorithm for SPIHT', which would satisfy resolution scalability for SPIHT compressed images but may not necessarily be resolution scalable under JPEG compression.

The above definition of scalable watermarking is, of necessity, a qualitative one. The point at which a watermark is considered detectable will depend on the expected functionality of the watermark, and the allowable error rates for the application. The notion of acceptable quality depends entirely upon the individual judgement of each watermarker as to the value of an image. Similarly, the improvement in quality provided by each layer is also subjective and, although objective methods to measure image quality do exist, there is no consensus on which of these methods is best. However, issues much the same as these are confronted in the evaluation of watermarking algorithms in general, and thus the majority of the related discussion is delayed until section 3.2.3 (page 68), on fair evaluation. Instead, the following section focuses on the derivation of concrete measures of the scalability properties: detectability and graceful improvement.

3.1.2 Measuring Scalability Properties

In order to perform a meaningful evaluation of watermark scalability in terms of the definition in the previous section, it is necessary to convert the qualitative terms into quantitative ones, so that watermark scalability can be measured and compared. As is the case with any such conversion, the appropriateness of the selections made will be substantially application dependent. A consequence of this is that in any general study, such as this one, the choices made will always be somewhat arbitrary. However, they will at least allow the definition of reasonable measures; $\mathcal{D}$ for detectability and $\mathcal{G}$ for graceful improvement.

Although the detectability and graceful improvement properties should hold across all images, messages and keys for a watermark to be considered scalable, the detectability and graceful improvement measures defined here are based on a single watermarked image $I' = \mathsf{Embed}_{\mathrm{X}}(I, sk_e, \Lambda)$. The use of these measures to evaluate the scalability of an algorithm X for all images, messages and keys is considered in section 3.2.

3.1.2.1 Detectability

The detectability property states that

1. **Detectability**
   The watermark is detectable in any subimage which is of acceptable quality.

Before we can measure detectability, it is necessary to first define *acceptable quality*.

3.1.2.1.1 Defining Acceptable Quality

Precisely what constitutes acceptable quality is highly subjective. The author or distributor of the content will generally make some assessment as to the level of degradation that the content can sustain before it no longer holds either commercial value or artistic merit.

One method for determining whether a scaled image is of acceptable quality would be to use a perceptual distortion measure $D_{\mathcal{F}}$ to measure the distortion between the original image and the scaled version, and compare it to a threshold $T_{\mathcal{F}}$, that represents the amount of distortion at which the scaled image is considered to be of unacceptable quality. That is, a subimage $I^{\mathcal{F}} = \mathsf{Scale}(I', \Xi)$, produced from an original image I' is considered to be of acceptable quality if

$$D_{\mathcal{F}}(I, I^{\mathcal{F}}) < T_{\mathcal{F}} \tag{3.1}$$

Although this is a reasonable definition of acceptable quality, it suffers from much the same drawbacks of perceptual distortion equalization for processing (section 3.2.3.2, page 72). That is, it may take many scaling attempts with each watermarked image before the threshold of least acceptability is reached,[3] which will cause problems if the measure $D_{\mathcal{F}}$ is subjective (section 3.2.3.1.1, page 69). Resolution scaling also results in changes to the image geometry, which will cause problems if the measure $D_{\mathcal{F}}$ is objective (section 3.2.3.1.2, page 71).

Any application using scalable compression will require the selection of the base layer, which forms the lowest resolution or quality subimage. We can reasonably assume that the lowest resolution or quality layer will be chosen to be the most highly scaled version that is useful to at least some proportion of users, and therefore is at the threshold of acceptable quality. Thus, an alternative approach is to choose a fixed level of scaling $\Xi_{\mathcal{F}}$, that will produce images at roughly the desired level of perceptual quality but which can be specified using the parameters of the scalable compression algorithm. This fixed level of scaling is then used to define the lowest resolution or quality layer, and the subimage

[3] This occurs because the level of scaling is typically set according to compression-focused concepts, such as the compression rate or the included frequency bands, rather than according to perceptual quality.

composed solely of the lowest resolution or quality layer is considered to be the most highly scaled image that is of acceptable quality.

Let the base layer subimage $I^{\mathcal{F}_0}$ of the watermarked image I', correspond to scaling parameters $\Xi_{\mathcal{F}}$

$$I^{\mathcal{F}_0} = Scale(I', \Xi_{\mathcal{F}}). \tag{3.2}$$

Then, any subimage $I^{\mathcal{F}_l}$ containing layers 0 through l, where $l \geq 0$, of the watermarked image I' is deemed to be of *acceptable quality*.

This method, despite not resulting in a constant perceptual quality for each tested image, is more practical experimentally, and also fits more closely with the expected use of scalably compressed images.

For the JasPer implementation of JPEG2000, the level of scaling $\Xi_{\mathcal{F}}$ would correspond to either a compression rate or a number of resolution layers, depending on the particular type of scaling. For convenience of notation, while $\mathcal{F}$ may still be used to represent scaling in general, quality scaling will be represented by$\mathcal{Q}$ and resolution scaling by $\mathcal{R}$.

In the majority of this thesis, the lowest quality layer is defined to be at compression rate of 0.01

$$I^{\mathcal{Q}_0} = \mathcal{Q}(I', 0.01), \tag{3.3}$$

which corresponds to a subimage with $\frac{1}{100}$th the file size of the original image, and the lowest resolution layer is defined to be the smallest of six resolution layers, which corresponds to a subimage with $\frac{1}{1024}$th the area of the original image.

$$I^{\mathcal{R}_0} = \mathcal{R}(I', \frac{1}{6}), \tag{3.4}$$

3.1.2.1.2 Measuring Detectability

The point at which a watermark is considered detectable will also depend on the application requirements. As was noted in section 2.1.1 (page 13), the detection output, $\{True\}$ or $\{False\}$, is often computed by calculating a detection statistic γ and comparing it to an application dependent threshold T. If the value of the detection statistic exceeds the threshold, $\gamma > T$, then the watermark is detected.

If the threshold T (or some other detection criterion appropriate to the particular application) is known, then detectability can be measured by applying the detection algorithm to each subimage $I^{\mathcal{F}} = \mathcal{F}(I')$ that is of acceptable quality and calculating the number of subimages for which a true positive occurs

$$\frac{|I^{\mathcal{F}} : \{\mathsf{Detect}(I^{\mathcal{F}}, I, sk_d) = \{True, M\} \wedge I^{\mathcal{F}} \text{ is of acceptable quality}\}|}{|\{I^{\mathcal{F}} \text{ is of acceptable quality}\}|}. \tag{3.5}$$

For a general evaluation, in which no specific application scenario is known, this true positive rate could be estimated using an arbitrary detection threshold. However, a better approach would be to use the detection statistic[4] γ directly, so that the results could later be compared to an appropriate threshold.

The calculation of the mean detection statistic across all subimages of acceptable quality would provide a general indication of detectability; however, it would not allow a determination of whether, for a given threshold, the watermark is detectable in all subimages. That information can be obtained if we instead use the minimum value of the detection statistic

$$\mathcal{D} = \mathsf{min}\left(\{\gamma(I^{\mathcal{F}}) : I^{\mathcal{F}} \text{ is of acceptable quality}\}\right). \tag{3.6}$$

For the vast majority of scalable compression techniques, the subimages are ordered $\{I^{\mathcal{F}_0}, I^{\mathcal{F}_1}, \ldots, I^{\mathcal{F}_{L-1}}\}$ so that each subimage consists of the previous subimage and a layer of additional data (section 2.2.2.3, page 36). In this case, the minimum value of the detection statistic should occur for the subimage containing the least data, which is the subimage consisting only of the lowest layer, and thus the detectability measure is simply the detection statistic for the base layer subimage

$$\mathcal{D} = \gamma(I^{\mathcal{F}_0}). \tag{3.7}$$

In this thesis, quality detectability will be measured using the detection statistic at a compression rate of 0.01

$$\mathcal{D}^{\mathcal{Q}} = \gamma(\mathcal{Q}(I', 0.01)), \tag{3.8}$$

and resolution detectability will be measured at the lowest of six[5] resolution layers or $\frac{1}{1024}$th the original image area

$$\mathcal{D}^{\mathcal{R}} = \gamma(\mathcal{R}(I', \frac{1}{6})). \tag{3.9}$$

3.1.2.2 Graceful Improvement

The graceful improvement property states that

2. Graceful Improvement

Increased portions of the scaled content provide reduced error in watermark detection, appropriate to the improved content quality.

[4] Although the detection statistic γ is typically used for zero-bit watermarking algorithms, for other algorithms a similar performance measure, such as the bit error rate of the extracted message, may be substituted for γ.

[5] In the evaluations of sections 4.2 and 4.3.2 (pages 100 and 122), detectability is instead measured using the second-lowest resolution layer, $\Xi^{\mathcal{R}} = 2/6$, to avoid disadvantaging algorithms which avoid lowest resolution coefficients. This is further explained in these sections.

Thus, to measure graceful improvement, it is necessary to measure the improvement in content quality provided by each layer and determine what constitutes an appropriate reduction in error for any given improvement.

For simplicity of discussion, it is assumed that the subimages are ordered so that each consists of the previous subimage and an additional layer. This is the case for most scalable compression algorithms; for those algorithms for which this is not the case we can impose an ordering based on the perceptual quality of each subimage and still obtain a reasonable measure of graceful improvement. Also, for simplicity of notation, the superscript indicating the type of scaling ($\mathcal{Q}$ for quality, $\mathcal{R}$ for resolution, $\mathcal{F}$ any type) will be dropped, so, for example, a subimage consisting of the layers 0 to l inclusive will be denoted I^l rather than $I^{\mathcal{F}_l}$.

3.1.2.2.1 Improved Content Quality

Particular layers may contribute far more to the quality[6] of the image than others. Rather than attempt to measure the improvement in quality provided by a given layer in an absolute sense, it seems appropriate to measure the contribution of each layer relative to the full image.

If no layers of the image have been received, then there has been a total improvement of 0%, and the user has no information about the image. The most plausible image to represent this situation is a mid-grey, empty image I^e.

The full scalably compressed image I^{L-1}, containing all L layers, is, although of lower quality than the uncompressed image, the highest quality image that any user can expect to obtain and should thus be considered to be full quality. Thus, if I^{L-1} has been received, then there has been a total improvement in quality of 100%, as the user, who was initially unable to reconstruct any part of the image, can reconstruct the full image,

We can measure the perceptual quality P_l of each scaled subimage I^l, relative to the original image I, using an image quality measure. In this thesis, *peak signal to noise ratio* (PSNR) is used as the measure of perceptual quality when calculating P_l, because it is a well known measure of image quality which still gives reasonable results when the distortions are far above the threshold of perceptibility.[7] For the purposes of measurement, because PSNR measurements require the two compared images to be of the same size, each resolution scaled subimage I^r is decompressed as though the discarded resolution layers

[6] Note that the term quality is used here in the generic sense of the visual quality or value of the content, as it is in the definition of graceful improvement. The layers themselves need not be quality layers; resolution layers also unarguably contribute to the quality of an image.

[7] The major concern with PSNR (section 3.2.3.1.2, page 71) is its inaccuracy when the distortion to the image is localized, which is unlikely to occur with scalable compression.

are present but all zero, so that all decompressed subimages are the same size as the original.

This allows us to describe the improvement in quality of a subimage I^l over the empty image I^e, as a proportion of the improvement provided by the full image I^{L-1}:

$$\frac{P^l - P^e}{P^{L-1} - P^e} \tag{3.10}$$

Thus, the improvement in image quality provided by layer l, as a proportion of the improvement provided by the full image I^{L-1} is

$$\mathrm{iq}^l = \frac{P^l - P^{l-1}}{P^{L-1} - P^e} \tag{3.11}$$

3.1.2.2.2 Appropriate Error Reduction

To appropriately protect all layers of the image, the detectability of the watermark should improve with the addition of each layer. However, it is not immediately clear how much of reduction in detection error is appropriate given a quality improvement of iq^l.

Consider a watermark that consists of N elements $W = \{w_1, w_2, \ldots, w_N\}$ and assume that all elements are equally valuable to the detectability of the watermark, regardless of their position in the watermark sequence. Given an L-layer image, in which each layer is of equal value $\mathrm{iq}^l = \frac{1}{L}$, the most logical approach is to divide the N watermark elements equally between the L layers. With any other division, an attacker could, for example, destroy more than $\frac{1}{L}$ of the watermark by damaging a layer worth only $\frac{1}{L}$ of the image. Thus the reduction in detection error appropriate to each layer should be exactly that gained by the addition of $\frac{1}{L}N$ watermark elements.

Similarly, when layers do not contribute equally to the image quality, the best overall protection will be obtained if the N watermark elements are distributed according to the image quality contribution of each layer, i.e. if $\mathrm{iq}^l N$ elements are embedded in layer l. Thus, an appropriate reduction in detection error for the addition of the lth layer is the reduction gained by the addition of $\mathrm{iq}^l N$ watermark elements. This represents the ideal behaviour for graceful improvement in a scalable watermark.

3.1.2.2.3 Measuring Graceful Improvement

The graceful improvement of a watermarking algorithm X can be measured according to how closely the actual division of the watermark amongst the image layers matches the ideal division. The ideal number of watermark elements in layer l was discussed in the previous section, and is denoted

$$\iota^l = \mathrm{iq}^l N, \tag{3.12}$$

for an N element watermark. The number of watermark elements that can be extracted from layer l is denoted ϵ^l.

If each image layer contains a set of whole watermark elements, the calculation of ϵ^l is straightforward: ϵ^l is simply the number of watermark elements extracted from the layer l. However, it may be the case (in fact it is the case with quality scaling for the algorithms in chapter 4) that each layer contains only partial watermark elements, making this straightforward approach impossible. Such a problem can be overcome by measuring the detection statistic at subimages l and $l-1$ considering the value ϵ^l to represent the equivalent number of whole watermark elements necessary to produce the observed increase in the detection statistic of $\gamma(I^l) - \gamma(I^{l-1})$.

How greatly the extracted values differ from the ideal values, across all layers can be measured using

$$\Delta = \sum_l \frac{(\epsilon^l - \iota^l)^2}{\iota^l}. \tag{3.13}$$

The use of the squared difference between extracted and ideal values results in a preference for more evenly distributed watermarks, which obtain values close to the ideal for all layers, over those which are very close to ideal on most layers but far from ideal on other layers. Division by the ideal ensures that the differences between extracted and ideal are considered in proportion to the layer value.

A more convenient measure of how closely the extracted values match the ideal values can be formed by normalizing Δ so that the possible values range between 0 and 1 and subtracting the normalized value from 1, so that a better distribution of the watermark amongst the layers will result in a larger graceful improvement value. This process is shown in appendix B.1, and gives the following graceful improvement measure:

$$\mathcal{G} = 1 - \frac{\Delta}{N(\frac{N}{\iota^{\mathfrak{m}}} - 1)}, \tag{3.14}$$

where $\mathfrak{m}$ is the non-empty layer which contributes least to the image quality (i.e. $0 < \iota_{\mathfrak{m}} \leq \iota_l$ for all layers l such that $\iota^l > 0$). A graceful improvement value of 1 represents a watermark which perfectly matches our ideal division, which is to embed according to the value of each layer $\epsilon^I = \iota_I$. A graceful improvement value of 0 represents the worst possible choice for dividing the watermark into layers, which is to embed the entire watermark in the least valuable (non-empty) layer $\mathfrak{m}$.

3.1.3 Scalable Watermarking Literature

Although the existing works on scalable watermarking are primarily focused on the development of new scalable watermarking algorithms, each makes some judgement (either

explicitly or implicitly) about what properties define a scalable watermarking algorithm. Three such properties repeatedly occur in the literature. Two of these, that the watermark be detectable even in highly scaled content and that detection ability should improve as more content is received, are essentially identical to the detectability and graceful improvement properties used in the definition given in section 3.1.1 (page 52). The third is that the watermarking algorithm be tailored to match the compression algorithm,[8] typically including, though not necessarily limited to, embedding in the same transform domain as that of the compression algorithm.

Early work on the integration of watermarking and scalable compression algorithms concentrated on two of the three properties: compression tailoring and graceful improvement. The detectability property was typically not considered or, in the case of [178], was specifically rejected.

The concept of scalable watermarking was first introduced by Wang and Kuo [201] as the watermarking component of an "integrated progressive image coding and watermarking system", allowing simultaneous image copyright protection and image compression with progressive display. Wang and Kuo stated that in order for such integration to be successful it is important to have the progressive property both in compression and watermark, so that "the retrieved watermark strength increases with more coded bitstreams received". They also suggested that the same transform domain should be used for both compression and watermarking. Indeed they use not only the same domain but also the same significant coefficient selection scheme for both their scalable compression algorithm [200] and their watermarking algorithm. However, their paper contains nothing to suggest they considered a detectability property as necessary for a scalable watermark.

Their further work with Su [177, 178], proposing a blind robust watermark for the EBCOT [187] compression algorithm, did not extend their definitions of either the graceful improvement property or the compression tailored property; however, it does shed light on their opinion of detectability. They suggest that, even when the detection statistic exceeds the threshold, detection in highly scaled subimages should be delayed until enough of the image has been received that the number of extracted watermark elements exceeds some threshold.

The same properties (graceful improvement and compression tailored) were identified by Chen and Chen [25] for their non-blind watermark for the copyright protection of spectral-selection mode JPEG images. Chen and Chen explicitly emphasized that upon receipt of "more information of the watermarked image, the bit error rate (BER) of the retrieved watermark image decreases". They added that such a watermarking algorithm

[8] This third property is not included in the definition as it does not appear necessary for the definition of a scalable watermark (section 3.1.1, page 52).

must "take into consideration the way the image is transmitted", tailoring the watermarking algorithm to the scalable compression algorithm rather than simply progressively transmitting a watermarked image; Although no mention was made of detectability, a restriction was added to allow watermark detection at low resolutions: coefficients were selected only from the first N AC coefficients in each block.

Some other early papers, despite using compression algorithms with the capacity for scalability, appear interested in watermarking and compression, rather than watermarking and scalable compression. They provide valuable results on the advantages of compression tailoring, but mention neither the detectability nor the graceful improvement property.

While Hartung and Girod [66] considered watermarking of MPEG-2 video they focused entirely on integration with compression and did not consider watermark scalability. They suggested that watermarking in the same transform domain as the compression scheme is preferable, particularly when transmitting differently watermarked versions of the same compressed video. The reason for this is that, when the compression and watermarking domains are identical, watermarking an already compressed video requires only partial decoding and reencoding, which they show requires a fraction of the computation, and offers a better quality result than full decompression and recompression. They also stated that, for fixed bit-rate applications, watermarking should not increase the bit-rate of the video.

Wolfgang et al. [206] also focused entirely on compression tailoring, and concluded that watermarking in the same transform domain as the compression scheme was preferable, although for a different reason than Hartung and Girod. Their experiments, with the detection of DCT and DWT domain spread spectrum watermarks after DCT-based (JPEG) or DWT-based (CEZW) compression, showed that in highly compressed images watermark detectability was improved by matching the watermarking and compression domains.

Later work considered detectability as an important property for a scalable watermark; sometimes alone but usually in addition to graceful improvement. Descriptions of desirable properties for a scalable watermark ceased to emphasize compression tailoring, yet, despite this, matching the embedding and compression domains remains almost universal.

Steinder et al. [174] provided a blind watermarking algorithm for image authentication, designed to be compatible with SPIHT compression. They were the first to introduce a notion similar to the detectability property, stating that for an authentication watermarking to be successfully combined with 'network conscious' (i.e. scalable) image compression, the watermarking algorithm should not require the entire image before authentication can begin but should instead allow authentication as soon as possible. Indeed their algorithm used only the LL subband, for both embedding and feature generation, which allowed

authentication to occur immediately upon receipt of the lowest resolution layer. No concept of graceful improvement is to be found in the paper and their watermark was not improved by the receipt of higher layers, despite the title "Progressively Authenticated Transmission". The lack of such a concept resulted in a security issue, which Steinder et al. themselves observed: given a small LL subband, it is possible to tamper with the image without affecting the LL subband and thus still pass authentication.

Also of note is the paper on the progressive detection of watermarks by Tefas and Pitas [189], which does not involve any integration with compression but does include concepts that can be identified as detectability and graceful improvement. They use the term 'progressive watermark detection' to denote a detection algorithm that may be successfully applied to a small region of the image, which is a detectability property, but also increase the detection threshold as the size of the region increases, suggesting the desirability of graceful improvement.

The preceding works used the term 'progressive' rather than 'scalable', emphasizing the gradual receipt of data over the expectation that limited quantities will be received. This preference for early detection during transmission as opposed to detection under a variety of possible rate constraints may be the reason that it was not until the discussion by Lin et al. [110] that specific mention was made of the requirement (implicit in the algorithm of Steinder et al.) that the watermark be "detectable when only the base-layer is decoded". Lin et al. examined scalable video compression, outlined its likely effects on an embedded watermark, and proposed a number of related research issues, many of which are appropriate for image watermarking. They suggested that image loss during transmission is not subject to the same quality constraints that exist for most attacks, that watermark embedding may take place at the source, receiver or within the network and that attacks using error concealment techniques and on the watermark synchronisation should be resisted. More importantly, they stated that both the base layer and the enhancement layers should be protected, and that the detectability of the watermark should improve uniformly with the compression rate of the decoded video.

Guo and Georganas [61] added resolution scalability to a DCT domain watermarking algorithm for JPEG images by grouping coefficients into resolution layers similar to those used in DWT domain algorithms. Although the desirability of various properties was not explicitly discussed, the reported experimental results (for resolution scaling to $\frac{1}{64}$th the area of the original and quality scaling to JPEG quality 10, suggest that both graceful improvement and detectability were considered. Unlike most previous authors, they did not exclusively embed in the same domain as the compression algorithm; they added a DFT domain watermark to improve robustness against geometric attacks.

Detectability was the scalability property that most interested Lu et al. [114], who

considered both resolution and quality scalability, down to $\frac{1}{64}$th the area of the original and JPEG quality 10. Some level of graceful improvement was shown, but the watermark was restricted to the 15 lowest frequency DCT coefficients in each JPEG compressed block, thereby limiting the possible resolution graceful improvement. Compression tailoring was not explicitly discussed, although the embedding and compression domains were matched.

Danyali and Amiri [37] focused entirely on detectability, producing a blind watermark that is detectable under resolution scaling to $\frac{1}{16}$th the area of the original and quality scaling to $\frac{1}{80}$th the original file size. Compression tailoring was not discussed, although the wavelet domain used for embedding matches with the HS-SPIHT algorithm used for compression. Although quality graceful improvement was achieved, graceful improvement does not appear to have been actively considered, and the higher resolution layers contain no watermark. This has little impact on the already poor security, however, as the downsampled logo is embedded directly using quantization, allowing the watermark to be easily identified and removed or replaced.

Li et al. [105] ensured detectability for their *audio* watermarking algorithm, even at the expense of watermark imperceptibility. However, rather than gracefully improve the detectability of the watermark by embedding in the refinement layers, they modified the refinement layers to remove the watermark. This unusual approach results in a graceful improvement in imperceptibility rather than robustness. The resulting watermark is therefore detectable in the lowest layer, but the imperceptibility of the watermark improves with each additional layer so that the full watermarked content is identical to the original. Although the fact that the watermark is self-removing, upon decompression of the full content, is problematic for applications where security is important, this approach may be ideal for data enrichment of content where lossless compression is desired.

Lin et al. [112] followed the more usual concepts of scalability in their work on collusion resistant fingerprinting of *temporally scalable video*. However, unlike most algorithms, in which detection results from all layers are weighted evenly and the graceful improvement arises purely from an increase in the available watermark data, their algorithm uses the detection statistics from individual layers to actively improve detection of a colluder's watermark, by excluding certain layers from the detection process.

Huang et al. [73] consider both detectability and graceful improvement, placing roughly equal emphasis on each property. Their genetic algorithm provides both properties for (quality scalable) JPEG spectral selection mode, down to the 13 lowest frequency coefficients in each 64-coefficient block; however, they do not examine resolution scalability.

Meerwald and Uhl [123, 124] considered both detectability and graceful improvement, but categorize quality scalability as separate from resolution and temporal scalability, as it is more easily achieved by traditional watermarking algorithms. They also proposed

other scalability categories: complexity scalability, a concept very similar to detectability, describes the ability to halt detection early and thereby save much computational effort; and distribution scalability is used to describe the ability to accommodate increasingly large numbers of concurrent users in a fingerprinting system. Despite these new categories, the experimental results for their algorithms focus primarily on detectability and graceful improvement, with the DWT version of their image watermarking algorithm providing both properties at $\frac{1}{16}$th the area of the original and $\frac{1}{80}$th the original file size.

3.2 Evaluation of Scalable Watermarking

Having established a definition of watermark scalability and constructed appropriate measures for its properties, it becomes possible to evaluate the scalability of a watermarking algorithm. This thesis uses an experimental evaluation method, as described in section 3.2.1.

This method is not specific to scalable watermarking; it is by far the most popular method for watermark evaluation. Related work on watermarking evaluation, including the origins and variations of the procedure used here but also work on alternative methods of evaluation, are discussed in section 3.2.2.

While the method itself is simple, it is important to set the parameters so that the evaluation is, as far as possible, 'fair'. That is, the evaluation should facilitate comparison among watermarking algorithms, not merely through clear reporting of results but by allowing repeatable experiments with well-specified parameters. In particular, the embedding and attacking stages should be designed so that all watermarks are equally strong, to ensure that the evaluation will not be inherently biased towards a particular algorithm. The problem of fairness, in both the embedding and attacking procedures, is discussed in section 3.2.3.

While most of the evaluations in this thesis are simply to demonstrate that both resolution and quality scalability have been achieved, the evaluation of section 4.3 (page 113) is intended to show the superiority of one watermarking algorithm over another. This requires a sufficiently accurate evaluation of scalability to be able to distinguish differences in performance between algorithms. Section 3.2.4 highlights the need for a sufficient number of test images in such comparisons, a factor that has been largely neglected in the watermarking literature, and describes the statistical framework developed for the comparison of watermarking algorithms.

3.2.1 Experimental Method

All the evaluations in this thesis (sections 4.2, 4.3.2, 5.2 and 6.4, pages 100, 122, 158 and 243) use the following experimental procedure, which evaluates the performance of a given watermarking algorithm X on a set of test images I.

Embedding

The embedding algorithm $\mathsf{Embed}_\mathrm{X}$ is applied to a set of images $\mathcal{I}$ using a set of keys $\mathcal{K}$ and, if applicable, a set of messages $\mathcal{M}$ to generate a set of watermarked images $\mathcal{I}' = \{I' = \mathsf{Embed}_\mathrm{X}(I, M, sk_e, \Lambda) : I \in \mathcal{I}, M \in \mathcal{M}, sk_e \in \mathcal{K}\}$ with a fixed level of perceptual distortion.

Attacking

Each image is then subjected to some consistent form of processing (or attack) F, to generate a set of attacked images ${\mathcal{I}'}^F = \{I^F = F(I'), I' \in \mathcal{I}'\}$

Detection

The detection algorithm is applied, $\mathsf{Detect}_\mathrm{X}(I^F, sk_d, \Lambda)$, to each attacked image and the detectability of the watermark, or some other property of interest, is measured.

The specifics of each evaluation, including the embedding strength, attack type and measurement process, vary and are reported individually for each section. Because most of the evaluations are of watermark scalability, the attacks are typically various levels of resolution and quality scaling, which are applied using the JasPer implementation of JPEG2000. The properties of interest at the detection stage are detectability and graceful improvement (section 3.1.1, page 52), which are calculated using the measures developed in section 3.1.2 (page 54).

3.2.2 Benchmarking Literature

While there are no specific works on the evaluation of scalable watermarking algorithms, the issues to be considered are largely the same as in watermark evaluation in general. This section contains a brief overview of the evaluation of watermarking algorithms as it appears in the literature.

Although some work has been done towards theoretical algorithm evaluation of watermarking algorithms, the focus of the vast majority of the literature is on experimental evaluation, which is the type of evaluation used in this thesis.

Su et al. [175] used theoretical models for both the image and watermark, in their work comparing the robustness of watermarking algorithms to white noise addition and linear filtering. However, it is not clear how applicable such theoretical models would be for real images nor that it is necessarily feasible to apply this approach for complex types of processing, such as scalable compression.

Chen and Wornell [23] used theoretical models of the watermark signal to noise ratio and the embedding-induced distortion in their demonstration of the superiority of QIM over both additive spread spectrum and quantize-and-replace embedding.

Adelsbach et al. [2] worked towards formal definitions of watermark robustness. They suggested that further work may, eventually, lead to reductionist proofs of watermark performance based on hard problems as can be found in cryptography. Such proofs would provide an excellent basis for algorithm evaluation; however, it is clear that more development is needed before this theoretical approach will be viable.

The fundamental work on the evaluation of watermarking algorithms was the paper by Kutter and Petitcolas [95], which describes an experimental structure, applying a set of attacks to each of a set of images watermarked with a fixed payload at a fixed visual quality and measuring the detection outcome, that is used essentially unchanged today. In addition to this, Kutter and Petitcolas proposed a benchmarking program, Stirmark, which allowed authors to evaluate their algorithms against a set of standard attacks in a consistent manner. Much emphasis was placed on controlling the image quality (i.e. the embedding-induced distortion) when measuring robustness, which is critical to the success of the method. The composition of the set of test images was also discussed, specifically that the test set should include a large range of image types.

Fridrich and Goljan [53] presented a similar method for watermark evaluation, but allowed minor modifications to watermarking algorithms so that they could be better compared. Algorithms were compared using both a zero-bit and a 60-bit payload, so the algorithms which did not support both payloads were modified to allow direct comparison.

Pereira et al. [139], in their paper describing the Checkmark benchmarking system, focused primarily on the inclusion of additional attacks, including new geometric and protocol attacks as well as more sophisticated watermark estimation and removal. They also noted that not all attacks were equally important in all applications and suggested that, in summarizing results, different types and strengths of attack should be weighted according to the intended application.

The same solution was proposed by Solachidis et al. [172], who also developed a benchmarking program, called Optimark. This paper was the first to discuss the impact of the number of keys used in the experiments on the reliability of the estimated false positive and false negative rates. Also of note is that, contrary to all of the previous work on

experimental comparison, Solachidis et al. suggested that PSNR should be used as the measure of image quality despite its well known drawbacks, stating that PSNR was still the only globally applicable and acceptable quality measure.

Macq, Dittmann and Delp [115], discussed watermark requirements in some application scenarios and modularity and accessibility in benchmarking tools in their work on the CERTIMARK project. They restated the importance of effective objective image quality measures beyond PSNR, in particular the need for image quality measures which give reasonable results for the distortion induced by geometric attacks such as rotation, for which PSNR is particularly poor. Although they noted the difficulty of experimentally obtaining accurate false positive results at extremely low false positive rates, no potential solutions were offered.

Kim et al. [87] used the same experimental structure as previous algorithms, in their Watermark Evaluation Testbed, but, rather than weighted averages, they used Taguchi loss functions to summarize image quality and robustness results and the area under the ROC curve to summarize false positive and false negative results across all thresholds. They proposed a partial solution to the problem of inaccurate measurement at extremely low false positive rates, suggesting that such rates can be modelled by assuming the detection statistic is independent and normally distributed.

A variety of attack-specific measures of image quality were presented by Kim and Delp [86], also for the Watermark Evaluation Testbed. These included measures for image quality after geometric attacks such as rotation and column dropping and were based on estimating and inverting each attack to allow PSNR based distortion measurement. They suggested that future work should involve the extension of the results to include confidence levels.

Dittman et al. [41] detailed their system for the comparison of audio watermarking algorithms. However, their approach differs substantially from other papers on experimental evaluation in that it eschews payload and embedding-induced distortion standardisation in order to plot the tradeoff between payload, robustness and imperceptibility obtained by each particular algorithm and choice of parameters. Also unlike previous authors, who typically focus on a few preferred measurement techniques, Dittman et al. describe a large number of possible measures for capacity, robustness and imperceptibility.

3.2.3 Fair Evaluation

An evaluation between watermarking algorithms is fair if no algorithm is disadvantaged simply as a result of the evaluation procedure. In practical terms, this means that all aspects of evaluation procedure should be adjusted so that all properties of the algorithms, besides those being measured, are the same. For example, the same payload should be

embedded and the detection process should give the same false positive rate. In cases where the property is image independent this adjustment is relatively straightforward; however, many watermarking algorithms are image dependent, as are the effects of attacks such as compression. To obtain a fair evaluation the embedding and attack strengths should be adjusted so that the perceptibility of the watermark and the severity of the attacks are the same for all compared algorithms.

During embedding, the embedding strength is used to increase the detectability of a watermark at the cost of increasing the distortion caused by embedding. Thus, to obtain meaningful measurements with the above procedure, the embedding strength must be adjusted so that there is a fixed level of perceptual distortion to each image. This is not straightforward due to the inherent subjectivity of perceptual distortion and the lack of commonly agreed standards for objective distortion measurement, which are discussed in section 3.2.3.1.

Many attacks and types of processing, including JPEG2000 scaling, may be applied at different strengths. The strength of the applied attack will also affect the measured performance of the algorithm. In any experiment where an attack is applied, the both the attack type and strength are reported. Possible approaches to attacking that will allow a fair evaluation of watermarking algorithms are discussed in section section 3.2.3.2.

3.2.3.1 Fair Embedding

The watermark embedding strength should be the same for all algorithms. However, this is not simply a matter of setting the embedding strength to the same value for all algorithms, as most algorithms have different strength parameters, which affect the watermark differently. Instead, we recognize that in all cases, the embedding strength governs the image distortion caused by embedding, so if two watermarks equally distort an image they are embedded at equal strength. For a fair comparison between watermarking algorithms, the embedding stage for each algorithm must result in the same amount of perceptual distortion to each watermarked image.

This is made more difficult by the lack of agreement on the best method to measure perceptual distortion. There are two ways of measuring perceptual distortion: subjective measurement and objective measurement.

3.2.3.1.1 Subjective Quality Measurement

The imperceptibility of a watermark (section 2.1.2.1, page 15) is defined in terms of a human observer. In subjective measurement, the imperceptibility of the watermark (or, equivalently, the quality of the watermarked image) is measured directly using a human observer. Since human perception is not exactly the same for all observers, in practice a

group of observers is used to obtain a result that reflects the watermark perceptibility for the 'average' observer.

Subjective quality measurement can be divided into two main categories: forced choice and ordinal scale. Forced choice is more suitable for small quality differences, while the ordinal scale is more suitable for large quality differences. This makes forced choice the better method for measuring embedding-induced distortion.

In forced choice techniques, the observer is shown two images and asked to choose which is better. This pair may be a distorted image and the original from which it was derived, or perhaps two distorted images. By repeating the choice over many trials (either by using a large number of observers or a smaller number of observers and repeated tests), the ratio of trials in which each image was preferred can be determined; two images of identical quality should result in a ratio close to 1:1. The ISO standard [76] provides a method to convert this ratio to a 'just noticeable difference' or JND value, which represents the quality difference between the two images. The number of images displayed in a single trial is not, in fact, limited to two; the observer may be asked to rank multiple images from lowest to highest quality. The rankings may then be used to simulate multiple two-alternative choices by assuming that a higher ranked image is always preferred to a lower ranked image.

In ordinal scale techniques, rather than choose which image is better, the observer is asked to rate a single image on a quality scale. This scale may either be numeric or non-numeric (e.g. a 5-point scale: bad, poor, fair, good, excellent). A set of reference images may be provided to the observer, one for each point on the scale. The use of reference images, with known JNDs, both improves the consistency between trials and allows the scale results to be converted to JND values.

Subjective measurement is arguably the most effective method of achieving equal watermark-induced distortion, as it measures perceptibility directly using real observer responses, rather than estimating it using calculations from the image. However, the measurement process is extremely time consuming. Many trials may be required to set an appropriate level of distortion, for even a single test image I', and observers may tire over long sessions, leading to less accurate results. This limits the size of the test image set $\mathcal{I}$. Furthermore a number of willing and available observers must be selected and their visual acuity recorded; the instructions to observers must be consistent and recorded and the viewing conditions (illumination, distance from stimulus) must be measured and controlled, which may necessitate special equipment.

3.2.3.1.2 Objective Quality Measurement

A computed perceptibility measure is not only faster but allows a larger set of test images and easy reproducibility for any researchers wishing to verify the results. In objective quality measurement, perceptual quality is estimated from the image itself, often also with reference to the original image, using a model of the human visual system.

The human visual system (HVS) is complex and difficult to model. It includes not only the eye, but also the areas of the brain responsible for interpreting light received by the eye. Many HVS characteristics, such as luminance sensitivity, spatial masking and colour perception, have been well studied and modelled using psychovisual experiments. These models have been developed in isolation using simple stimuli, and it is not always clear how to best combine them, to represent perception of the complex stimuli found in images. Higher level characteristics, such as grouping and scene interpretation, have substantial effects on human perception; however, the development of models that include these characteristics is formidable task. This complexity has resulted in a variety of different models, emphasizing the aspects of the HVS that are most relevant to particular applications. Unfortunately, there is still no method for objective measurement that has been agreed upon by the watermarking community.

The *peak signal to noise ratio* (PSNR) is a simple and well known measure of image quality. The poor correspondence between PSNR values and perceived image quality, for images with strong localized distortion, has been well documented (e.g. [32, 139]) and is common to all MSE based measures. However, because this measure is commonly used and well known to the watermarking community, PSNR results may be more easily interpreted than those of other, less well known, measures, in cases where the intensity of the distortion is fairly uniform across the image.

More sophisticated measures, such as the Watson metric [204], the Noise Visibility Function [196] and the S-CIELAB metric [213], use models of the HVS and are thus better correlated with the perceptibility evaluations of human observers. Although a controlled viewing environment is not required, because the measurements are derived directly from the image, more sophisticated perceptual models do require the viewing conditions to be specified as input. Thus, for any experiment using HVS models, the relevant viewing condition information must be reported in addition to the target level of distortion for the embedding procedure.

An often unmentioned drawback of objective measures is that they may bias the results towards a particular watermarking technique. Watermarking algorithms use perceptual models to determine relative embedding strengths in different regions of the image; the more closely aligned the chosen perceptual distortion measure is with the perceptual model

of a watermarking algorithm, the better that algorithm is likely to perform. In the extreme case, when the watermarking algorithm and the embedding-induced distortion measure use the same model, the watermarking algorithm will embed most strongly in precisely the manner judged least perceptible by the distortion measure and so will perform better compared to a second algorithm that uses a different (yet perhaps significantly better) perceptual model. This problem cannot be easily resolved as it is natural that the models used in watermarking will align with those used in image quality evaluation. The best that can be done is to ensure that the perceptual distortion measures used for evaluation are well correlated with subjective measures of perceptibility.

The majority of evaluations in this thesis use the S-CIELAB perceptual distortion metric to determine an embedding strength that results in a fixed level of perceptual distortion to each image, although occasionally a fixed embedding strength is used for all images. In all cases, the method used to determine an appropriate embedding strength, along with any relevant viewing condition information, is provided in the description of the embedding stage for each experiment.

3.2.3.2 Fair Attacking

Typically, the image is subjected to some attack or other processing before the performance of the watermark is measured. This processing often reduces the measured performance of the watermarking algorithm, with stronger processing resulting in a greater reduction. For an unbiased comparison, each image that has been watermarked with algorithm A should undergo the same level of processing-induced distortion as the corresponding image that has been watermarked algorithm B.

This suggests that the technique chosen to measure and equalize embedding-induced distortion should also be applied to equalize processing-induced distortion. However, finding an appropriate perceptual distortion measure for attacking can be even more difficult than finding one for embedding. For this reason, evaluations of watermarking algorithms consistently avoid the use of a fixed level of perceptual distortion in the attacking step. Instead, a fixed processing strength is used as a far more efficient, and arguably more realistic, substitute.

3.2.3.2.1 Problems with Perceptual Distortion Equalization

Having decided on a perceptual distortion measure for embedding it seems reasonable to use the same measure for processing induced perceptual distortion. However, a measure selected for embedding-induced distortion is not necessarily a good choice for processing-induced distortion.

Many perceptual distortion measures, objective and even subjective, are focused on distortion at the threshold of perceptibility. These are useful for measuring embedding-induced distortion, which is intended to be imperceptible. However, such measures tend to lose their accuracy when the distortion is large and may thus be unsuitable for measuring the perceptual distortion of a processed image. This is likely to be the case in scalable watermarking, as high levels of processing-induced perceptual distortion are more widely acceptable in scalable watermarking scenarios than in those where the image is routinely received at full resolution and full quality.

The changes to the image that occur during processing may cause further problems for objective measures of perceptual distortion. These measures often rely on comparisons between the original and altered image on a pixel-by-pixel basis. Any form of processing which changes the geometry of the image, such as rotation, cropping or (importantly) resolution scaling, causes problems for many of these measures. While it is possible, as shown in [86], to restore the processed image to its original geometry[9], measurements on the restored image do not necessarily reflect the subjective distortion in the (unrestored) processed image.

Finally, determining the level of processing required to reach the processing-induced distortion target may be quite difficult. If either the distortion measure or the processing is complex, it may be necessary to repeat the attacking step several times on each watermarked image before the target level of perceptual distortion is achieved.

3.2.3.2.2 Processing Strength Equalization

A far less difficult and time consuming method is to use the same processing strength for all watermarked images. For example, if an image I'_A (watermarked with algorithm A) is cropped by 25% we assume that the image I'_B, (watermarked with algorithm B), will have undergone the same level of distortion if it, too, is cropped by 25% in the same manner.

This method avoids all of the problems associated with perceptual quality measurement, but assumes that the same level of perceptual distortion will result when the watermarked images are subjected to the same level of processing. Although this cannot be assumed in general, because the same level of processing may have dramatically different effects depending on image content and structure, it *can* reasonably be assumed for a pair of processed watermarked images I^F_A and I^F_B that are derived from the same original image I.

[9]The restoration of the image geometry after processing is used in this thesis, not in the context of attack strength adjustment but in determining the relative contribution of each resolution layer to the final image quality (section 3.1.2.2, page 57).

Because the differences between each watermarked image and the corresponding original image I are already constrained to be slight, as both watermarks must be imperceptible, a pair of watermarked images I'_A and I'_B should be similar, not only in quality but also in content and structure. Thus, if processing F is applied at a fixed strength, the processing should distort both I'_A and I'_B in much the same manner. As a result, the perceptual difference between the processed watermarked images I^F_A and I^F_B will be small. That is, this strength-based distortion equalization will allow a valid comparison between algorithms on the basis of pairs of processed, watermarked images that are derived from the same original.

Measuring watermark performance at a particular level of processing, rather than a particular level of perceptual quality, may also be preferred beyond the practicalities of avoiding the hassles of perceptual distortion measurement. For applications in which all images are expected to undergo particular forms and strengths of processing as part of a standard procedure, for example when specific levels of compression are used consistently in distribution or archiving, it becomes more important to measure the watermark performance at these set processing strengths rather than at any particular perceptual quality.

3.2.4 Comparing Algorithms

In section 4.3 (page 113), algorithms are compared to determine their relative performance in terms of resolution and quality scalability and, more importantly, to verify the claim that the developed HVS adaptive algorithm has superior scalability to its non-adaptive counterpart. The fact that there is a specific claim to be 'proven' in this comparison raises an issue that is important to the evaluation of watermarking algorithms but has been largely absent from the watermarking literature: the effect of the number of images on the reliability of the results.

In any comparison of image watermarking algorithms, the aim is to compare the relative performance of the algorithms A and B for a population of images $\mathcal{I}_{\mathrm{u}}$, which contains all images of interest, using experimentally obtained performance measurements (such as the detectability measurements) for each algorithm.

Let $\Omega_{A\mathrm{u}}$ and $\Omega_{B\mathrm{u}}$ denote the sets of performance measurements for watermarking algorithms A and B on the entire population $\mathcal{I}_{\mathrm{u}}$, then $\mu_A = \sum_{\mathfrak{a}\in\Omega_{A\mathrm{u}}} \frac{\mathfrak{a}}{|\Omega_{A\mathrm{u}}|}$ and $\mu_B = \sum_{\mathfrak{b}\in\Omega_{B\mathrm{u}}} \frac{\mathfrak{b}}{|\Omega_{B\mathrm{u}}|}$ are the *population means*, which represent the average performance A and B, respectively, on the population of interest. We say that algorithm A performs better[10]

[10]Comparing means is not always sufficient to determine the better algorithm. For example, we may prefer an algorithm that performs reasonably well on all images over an algorithm which performs extremely well on some images but not at all well on others. If the distributions of $\Omega_{A\mathrm{u}}$ and $\Omega_{B\mathrm{u}}$ have similar shape and spread we can expect their means μ_A and μ_B to provide a fair comparison of overall performance.

than algorithm B in the population of interest if $\mu_A - \mu_B > 0$.

Direct comparison of the population means μ_A and μ_B would require the (potentially infinite) sets $\Omega_{A\mathrm{u}}$ and $\Omega_{B\mathrm{u}}$ to be generated. Instead, performance is measured using a set of test images $\mathcal{I} \subseteq \mathcal{I}_{\mathrm{u}}$ chosen randomly from the population. For each image in $\mathcal{I}$, embedding, attacking and detection are performed using algorithm A, resulting in a set of measurements Ω_A. The same process is applied using algorithm B to yield a set of measurements Ω_B. The performance of algorithms A and B can then be compared using the *sample means* $m_A = \sum_{\mathfrak{a}\in\Omega_A} \frac{\mathfrak{a}}{|\Omega_A|}$ and $m_B = \sum_{\mathfrak{b}\in\Omega_B} \frac{\mathfrak{b}}{|\Omega_B|}$, rather than the population means. If $m_A - m_B > 0$, we can conclude that algorithm A performs better than algorithm B (and vice versa).

This is essentially the method used in all the benchmarking systems described in section 3.2.2 (page 66). However, since the set of images used for the comparison is only a subset of all possible images to which the watermarking algorithms could be applied, there is a chance that the average performance of an algorithm over the sample set of images $\mathcal{I}$, is not sufficiently close to the average performance over the population $\mathcal{I}_{\mathrm{u}}$. This method of comparison does not provide any assurance that the observed difference in means is reliable, and is not simply a chance result of the particular images used in the sample. The number and types of images used in the test set will affect the reliability of the result; however, it is not immediately clear how many images should be used. If a small number of images is used, there is a higher probability that the outcome does not reflect the real performance difference and is influenced by the properties of the test images. On the other hand, it is desirable to use the smallest possible set to minimize the required computation.

This problem has remained essentially unaddressed in the watermarking literature. While Kutter and Petitcolas [95] note that the types of image used in the test set should be representative of the types of images in the population of interest, and Kim and Delp [86] suggest the reliability of results may form part of their future work, the number of images chosen for watermarking comparisons remains ad hoc. Some papers use as few as two or three test images [24, 130] and others use hundreds [8, 149] without any justification in either case.

Benchmarking systems are often accompanied by a worked example or an image database, so that number of images could serve as a guide, but this number also varies: [53] obtain their results from 4 images, [41] uses 16 audio files, [95] provide 29 images and, at the other extreme, the system in [87] has a public version with 1301 images, and an internal research version comprising at least 5000.

One cannot expect to determine a 'right' number of images that would be appropriate for all comparisons, as differences between algorithms will be more or less difficult to discern depending on the attacks, performance measures and the algorithms themselves.

However, a more systematic approach is required than simply following either the choice made in an existing paper or our own intuition.

A framework is proposed, in section 3.2.4.1, as a means of solving this problem. The data-generation phase of this framework essentially consists of the experimental procedure that has already been described in section 3.2.1 (page 66). The remainder of the framework is developed to guide the appropriate use of statistical methods in designing the experiment (section 3.2.4.2) and in analysing the results (section 3.2.4.3),

3.2.4.1 Comparison Framework

The comparison framework proposed in this section uses the well established, statistical methods of hypothesis testing and power analysis to decide upon the 'right' number of images and to make a reliable and meaningful comparison between watermarking algorithms.

Hypothesis testing is a formal method for drawing conclusions by deciding between two competing claims (or hypotheses), based on measurements on a sample taken from the whole population, in this case measurements of algorithm robustness taken from he set of test images. By selecting an appropriate hypothesis test and applying it to the sample set, it is possible to estimate not only which algorithm performed better but also the probability of the observed difference between algorithms occurring by chance. If this probability is low, we can reliably conclude that one algorithm has better performance than the other.

Requiring this probability to be low, limits the chance of erroneously concluding that a difference in performance exists. However, this may cause us to erroneously conclude that no difference in performance exists. Increasing the size of the image set can overcome this problem, allowing both types of errors to be reduced; however, the use of too large a sample can waste considerable amounts of time and resources. *Power analysis* is a method for determining the size of the sample set required for a hypothesis test in which both types of error are limited to specified levels.

Once the 'right' number of images is determined, experiments can be performed on the test image set to produce a set of measurements.

These measurements form the sample set for the hypothesis test. It is important to use an appropriate test. The test that was originally planned may have to be replaced with a different one. This may occur, for example, when the assumptions required by the test are not satisfied by the sample data.

Finally, the results of statistical tests need to be interpreted; a positive test result does not necessarily mean that one algorithm will greatly improve robustness or improve

robustness in all possible applications. Raw results may be made more meaningful by relating them back to the original problem of watermark detection.

Thus, to achieve a reliable comparison between watermarking algorithms several additional steps must be added outside the experimental method described in section 3.2. It is necessary to choose a hypothesis test and determine the required number of images, source the images, test any assumptions required for the test, apply the hypothesis test and interpret the results. The entire process can conceptually be divided into three phases: design, data generation and analysis.

The comparison procedure is as follows:

Design

In the design phase, the high level decisions required for data generation and analysis are made. This includes the choice of relevant attacks and performance measures, the selection of the hypothesis test and the calculation of the number of images to be used.

Data Generation

The data generation phase involves the establishment of image and key sets, the application of the watermarking algorithms and attacks and the gathering of the measurements.

Analysis

The analysis phase begins once all the raw data have been generated. It involves the verification of any assumptions required by the chosen hypothesis test, application of the hypothesis test and interpretation of the results.

The data generation phase has already been described (section 3.2.1, page 66), thus only the steps relating to the design and analysis phases are discussed in the following sections.

3.2.4.2 Experiment Design

3.2.4.2.1 Choice of Attacks and Performance Measures

The conclusions reached in a given comparison will only be applicable when the conditions of the experiments are a close match to the intended application for the watermarking system. Experiments using a single type of processing will be inadequate if we require robustness to many types of processing. Similarly, robustness tests against a wide range of processing types may not apply if only a few specific types of processing are expected.

In the same manner, experiments that measure only robustness, if fragility or graceful improvement are required by the application will be insufficient.

This said, in most comparisons, the choice of attacks and measurement statistics will be relatively straightforward. Typically the purpose of the experiment will be to evaluate the relative fitness of the watermarking algorithms for some particular purpose, and thus certain attacks and measures will present themselves as particularly appropriate.

As the comparison in this thesis (section 4.3, page 113) focuses on scalability, the relevant attacks are various amounts of resolution and quality scaling, which are implemented using the JPEG2000 program JasPer. The amounts of scaling for different images are made fairly consistent by using the same scaling parameters on each image, rather than adjusting each attack on each image to obtain a precise image quality (see the discussion on fair attacking—section 3.2.3.2, page 72). Because the comparison is not related to a specific application, the particular settings for the resolution scaling attacks are chosen according to the typical number of resolution levels in a JPEG2000 image, as suggested by Adams [1]. The settings for the quality scaling attacks are, likewise, not obtained from any specific application but are chosen to provide a small yet noticeable increase in quality with each additional layer.

The performance measures used are the detectability $\mathcal{D}$ and graceful improvement $\mathcal{G}$ measures of scalability, which were derived in section 3.1.2 (page 54). Both measures are based on the watermark detection statistic γ rather than the true positive rate, as this requires far fewer images to obtain a reasonable estimate of performance. However, as a result, direct comparison using these measures requires that the watermarking algorithms use the same detection statistic, so the algorithms must be adapted to use the same detection statistic if they do not already do so.

3.2.4.2.2 Hypothesis Testing

Statistical hypothesis testing[13] is a method for deciding between two competing hypotheses: the null hypothesis and the alternative hypothesis, on the basis of a *test statistic* t.

- The null hypothesis H_0, conventionally states that there is no difference between the populations under study, that is the observed result t is entirely due to chance. For a comparison between means, the null hypothesis states that the average performances of algorithm A and algorithm B across all images $\mathcal{I}_\mathrm{u}$ are identical

$$\mu_A - \mu_B = 0.$$

- The alternative hypothesis H_1, conventionally states there is a real difference between the populations under study, that the observed result t is due to an underlying

difference in the populations. For a comparison between means, the alternative hypothesis states that the average performance of algorithm A across all images $\mathcal{I}_{\mathrm{u}}$ differs from that of algorithm B by some nonzero amount

$$\mu_A - \mu_B \neq 0.$$

Hypothesis testing requires the selection of a hypothesis test and the evaluation of the test statistic t, calculated from the sample measurements. If the value of t obtained is sufficiently unlikely under H_0, we entertain the alternative hypothesis H_1.

Provided the population measurements $\Omega_{A\mathrm{u}}$ and $\Omega_{B\mathrm{u}}$ satisfy a set of assumptions, t will have a known probability distribution when H_0 is true. The particular assumptions and the associated distribution of the test statistic will depend on chosen hypothesis test. This distribution is used to calculate a p-value, which is the probability p of obtaining a value of t that is less supportive of H_0 than the value that was calculated using the sample measurements, assuming H_0 is true. Thus, the smaller the value of p, the less likely it is that H_0 is true.

If this p-value is below some *significance level* α, H_0 is rejected in favour of H_1. The significance level α is the probability of erroneously rejecting H_0. If H_0 is rejected, we say that the observed difference is significant, which suggests it is not due to chance. The choice of significance level α establishes the *confidence level*, $1 - \alpha$. The confidence level is the probability of (correctly) accepting H_0 when H_0 is true. The calculation of the test statistic t and its associated p-values depends upon the chosen hypothesis test.

3.2.4.2.3 Choosing a Hypothesis Test

For comparisons, we are typically interested in which algorithm has the best average performance, which suggests a comparison between means. The most widely known test for comparing means is the two sample t-test (see appendix B.2). However, in watermarking experiments, we can watermark identical copies of the image I_i with algorithms A and B. This allows the use of the more powerful test, known as the paired t-test, which takes advantage of the natural pairing between the performance measurements $\mathfrak{a}_i$ and $\mathfrak{b}_i$ of the respective algorithms applied to the same image I_i.

3.2.4.2.3.1 The Paired t-Test

The paired t-test requires the calculation of the m_d and standard error SE_d of all the*paired differences* $d_i = \mathfrak{a}_i - \mathfrak{b}_i$ between the sample measurements to generate the test statistic

$$t_d = \frac{m_d}{SE_d} \tag{3.15}$$

which, if H_0 is true, will have student's-t distribution with $\nu = |\{d_i\}| - 1$ degrees of freedom. The further t_d is from zero, the less it supports the null hypothesis.

The corresponding p-value is the probability[11] that a test statistic t drawn from a student's-t distribution with ν degrees of freedom, t_ν, is further from zero than t_d

$$\begin{aligned} p &= P(t \sim t_\nu > |t_d|) + P(t \sim t_\nu < -|t_d|) \\ &= 2 \times P(t \sim t_\nu > t_d). \end{aligned} \tag{3.16}$$

If $p < \alpha$, we reject H_0 with confidence $1 - \alpha$ and conclude that there is a significant difference between algorithms A and B. If $p \geq \alpha$ we conclude that neither algorithm is significantly better than the other.

It should be noted that the paired t-test involves only two algorithms, therefore if several algorithms are to be compared using this test, it must be applied repeatedly, which causes an increase in the probability of incorrectly rejecting H_0. The primary purpose of the comparison in section 4.3 is to test specific claims regarding two algorithms (`hvs` and `nohvs`, section 4.3.2.1, page 123), so we choose not to correct for this increase. However, if many algorithms or many processing types are examined it may be necessary to correct for the increased probability of incorrectly rejecting H_0, by using a test designed specifically for multiple comparisons or by directly lowering the significance level α, which will increase the number of images required to maintain the same error rates. Shaffer's paper [164] provides a good discussion of this problem and some of the available solutions.

3.2.4.2.4 Power Analysis

The probability of erroneously accepting the null hypothesis H_0 is denoted β. The *power* of the test, $1 - \beta$, is the probability of (correctly) rejecting H_0 when H_1 is true. However, while H_0 normally represents a single possibility, e.g. $\mu_A - \mu_B = 0$, the hypothesis H_1 represents many different possibilities, e.g. $\mu_A - \mu_B = 0.001$, $\mu_A - \mu_B = 12$, $\mu_A - \mu_B = 34861$. Thus, the probability of correctly rejecting H_0 depends on which specific alternative is true; the smaller the difference between the two populations, the lower the power of the test to correctly detect a difference. Rather than evaluate power over all possible alternative hypotheses, power is typically calculated for a point alternative hypothesis, in which the difference is assumed to be some specific amount δ. Provided the true difference is at least δ, the test will have at least the calculated power.

[11] The notation $P(s \sim D > v)$ represents the probability that a sample s, drawn randomly from distribution D, has a value greater than v.

The probabilities α and β are also known as the type-I and type-II error rates.[12] They quantify the likelihood that a decision based on $\mathcal{I}$, about which algorithm is superior, will fail to reflect the true performance of the algorithms over all images $\mathcal{I}_{\mathrm{u}}$. If α is small, we can be confident that any significant difference is not simply due to the test image set. However, *for a fixed number of images*, there is a trade-off between α and β. So small values of α will cause large values of β. To obtain low values for both α and β the number of images n may need to be increased. *Power analysis* is concerned with the relationship between the sample size n, and the error probabilities α and β and the smallest substantial difference in algorithm performance δ. The formulae it provides can be used to determine how many images are required to maintain acceptably low error probabilities in a comparison between watermarking algorithms.

3.2.4.2.5 Calculating the Required Number of Images

Power analysis is used to determine the appropriate sample size for the hypothesis test. That is, the smallest number n of test subjects, in this case original images, that will keep both the type-I error rate α and the type-II error rate β low. Sample size calculations depend on the particular hypothesis test being applied, but in general, the size of the sample n required for a hypothesis test is dependent on

- the allowable error rates α and β,
- the variability of the measurements
- the difference δ that we wish to be able to detect

For a paired t-test, if the difference in population means is δ and the standard deviation of the difference values is σ_d, then the number of images required to achieve error rates α and β is given by:

$$n \geq \frac{\sigma_d^2}{\delta^2}(t_{\alpha(2),n-1} + t_{\beta(1),n-1})^2 \tag{3.17}$$

where $t_{x(y),\nu}$ is the value such that the probability that a sample t, drawn randomly from a student's t-distribution with ν degrees of freedom, is greater than $t_{x(y),\nu}$ is $\frac{x}{y}$. That is, $P(t \sim t_\nu > t_{x(y),\nu}) = \frac{x}{y}$.

[12] The error rates α and β are further known as false positive and false negative error rates, respectively. These terms, although more descriptive, are not used in this thesis, in order to maintain a clearer distinction between the false positive (α) and false negative (β) error rates for performance evaluation and the false positive (false alarm) and false negative (missed detection) error rates for watermarking detection. The commonality of terms is not accidental, as any test in which a decision is made regarding the truth or falsity of a given proposition (be it H_0 or "a watermark is embedded") can be considered in terms of false positive and false negative error rates.

The larger the error rates α and β are allowed to be, the smaller the sample size required. Conventionally, hypothesis tests use a confidence level of 95% ($\alpha = 0.05$) and a power of 80% ($\beta = 0.02$).

The less the measurements of each algorithm's perfomance vary with the choice of subject, the smaller the sample size required. Note that the population standard deviation of the measurements is typically unknown, but can be estimated using the sample standard deviation s. However, in order to calculate the sample standard deviation, it is necessary to first have the measurements, and taking the measurements requires the determination of the sample size, which requires the standard deviation! To break this vicious circle, the sample standard deviation can be obtained using measurements from a small sample, which will presumably, though not necessarily, turn out to be smaller than the required sample size n.

The larger the difference in performance, the easier it is to detect, and hence the smaller the sample size required. For small differences in performance, an extremely large sample may be required to reach power $1 - \beta$. However, in most circumstances, small differences in performance are not substantial enough to cause us to favour one algorithm over another, as they will be outweighed by non–performance-based factors. That is, for differences below some threshold, maintaining a power of at least $1 - \beta$ is no longer important, because incorrectly concluding that the algorithms have identical performance (H_0) is no longer a costly error. The value δ is set to denote this threshold, the smallest difference that is considered 'substantial', and the sample size is set to ensure a power of at least $1 - \beta$ whenever the population performance difference is at least δ. Because there is no standard method for determining how big a difference in performance must be before it can be considered substantial, the choice of δ is best made with knowledge of the field and is often subjective and/or application dependent.

3.2.4.3 Analysis

The final stage of the comparison is the analysis of the performance measurements. While the application of a hypothesis test (section 3.2.4.2.2) is a vital part of the framework, for a comparison to be fully successful, the underlying assumptions must be verified and the results of the hypothesis test interpreted.

3.2.4.3.1 Testing Assumptions

In modelling the distribution of the sample data, a hypothesis test may use certain assumptions. These will vary depending on the specific test, but more powerful tests will typically require more stringent assumptions. Thus, in order to ensure the hypothesis test is reliable, its assumptions must be verified. If all assumptions hold then the test can be

carried out as planned. If any assumptions are violated, the results of the planned test will be questionable. The same applies to any other assumptions made when designing the experiment, such as those required for the chosen measurement.

When an assumption is violated it may still be possible to follow the planned analysis provided the violation is mild and the test is known to be robust to that particular violation [100]. If such is not the case, then any assumption violations will require us to redesign the experiment so that the assumptions will be satisfied or choose an alternative hypothesis test with less stringent assumptions.

3.2.4.3.1.1 Paired t-test Assumptions

The paired t-test (section 3.2.4.2.3.1, page 79) relies on three assumptions, each to be verified before the paired t-test can be applied.

A1 Pairing assumption: each $\mathfrak{a}_i$ and $\mathfrak{b}_i$ form a matched pair[13].

A2 Independence assumption: $d_i = \mathfrak{a}_i - \mathfrak{b}_i$ is independent of $d_j = \mathfrak{a}_j - \mathfrak{b}_j$

A3 Normality assumption: $d_i \sim N(\mu_d, \sigma_d)$

In watermarking experiments, the pairing and independence assumptions can typically be assured during the design of the experiment. However, the normality of the difference measurements between algorithms is less certain, and must be tested in the analysis phase. If the distribution of the paired differences d_i proves to be non-normal, an alternative test may be necessary.

3.2.4.3.1.2 The sign test

Parametric tests assume the data have a specific distribution (the paired t-test assumes a normal distribution). When this distributional assumption is violated, one solution is the use of a non-parametric test, which does not rely on the data having any particular distribution. The *sign test* [71] is an alternative to the paired t-test, suitable for a non-normal, non-symmetric distribution. The null hypothesis for the sign test is that the median[14] of the paired differences is zero. If H_0 holds, we expect $P(d_i > 0) = P(d_i < 0)$. Thus the test statistic t_s, formed by counting the number of positive and negative differences and taking the maximum,

$$t_s = \max(\#d_i > 0, \#d_i < 0) \tag{3.18}$$

[13] We use the term 'matched pair' to emphasize that the pair $\mathfrak{a}_i, \mathfrak{b}_i$ cannot be arbitrarily assigned but must reflect a real dependency between $\mathfrak{a}_i$ and $\mathfrak{b}_i$. This dependency typically arises because the measurements $\mathfrak{a}_i$ and $\mathfrak{b}_i$ are taken either from the same subject or from different subjects which have been matched so that they are are as similar as possible in the characteristics expected to influence the measurement.

[14] The sign test examines the *median* paired difference rather than the *mean* paired difference; however, the median difference, like the mean, is still a good indicator of the overall difference between watermarking algorithms.

is binomially distributed with trial success probability 0.5. The associated p-value is $p = 2 \times P(t \sim B(n, 0.5) \geq t_s)$.

3.2.4.3.2 Applying the Hypothesis Test

Applying the hypothesis test is the straightforward procedure of taking the procedure for the chosen hypothesis test, or an appropriate alternative if the assumptions of the originally chosen hypothesis test are violated, and the recorded measurements from both algorithms, performing the required calculations and either accepting or rejecting the null hypothesis.

For the paired t-test the differences $d_i = \mathfrak{a}_i - \mathfrak{b}_i$, between corresponding pairs of measurements for the algorithms A and B, are calculated. Their mean m_d and standard error SE_d are computed, and the test statistic $t_d = \frac{m_d}{SE_d}$ is generated. The test statistic is compared to a student's t distribution t_ν, where ν is one less than the number of images used for the experiment, yielding a p-value $p = 2 \times P(t \sim t_\nu > t_d)$. If $p < \alpha$, we reject H_0 with confidence $1 - \alpha$ and conclude that there is a significant difference in the mean performance between algorithms A and B. If $p \geq \alpha$ we conclude that there is no significant difference in the mean performance of both algorithms.

For the sign test the differences $d_i = \mathfrak{a}_i - \mathfrak{b}_i$, between corresponding pairs of measurements for the algorithms A and B, are calculated. The number of positive differences $\#d_i > 0$, and the number of negative differences $\#d_i < 0$ are counted, and the test statistic $t_s = \max(\#d_i > 0, \#d_i < 0)$ is generated. The test statistic is compared to a binomial distribution $B(n, 0.5)$, where the number of trials is equal to the number of images n, and the trial success probability is 0.5, yielding a p-value $p = 2 \times P(t \sim B(n, 0.5) \geq t_s)$. If $p < \alpha$, we reject H_0 with confidence $1-\alpha$ and conclude that there is a significant difference in the *median* performance between algorithms A and B. If $p \geq \alpha$ we conclude that there is no significant difference in the median performance of both algorithms.

3.2.4.3.3 Interpreting the Results

It is important to recall that hypothesis testing is a means to an end. Careful selection of the number of images and the application of an appropriate test is useful only so far as it allows a meaningful comparison between algorithms.

If the hypothesis test shows a significant difference exists, the estimated direction and size of the difference should be presented. The test results should be related back to the original problem so that we can state which (if any) algorithm is better and indicate how much better it is.

When using statistical hypothesis testing there is a risk of placing too much focus on the statistical significance of the results. A *significant* difference exists when the observed

difference does not appear to be due to chance (i.e. $p < \alpha$). A *substantial* difference exists when the observed difference is sufficiently large that it has a non-trivial impact on the system's performance in the intended application. While it is important to establish that the difference is significant, we are ultimately interested in the existence of a substantial difference.

The estimated difference between algorithms should be viewed with reference to the minimum substantial difference δ that was established during the design phase, and ideally the practical effect of the estimated difference should be examined. Of course, what constitutes a substantial or practical difference is both application dependent and subjective, and can only be determined by careful consideration of the application domain.

Chapter 4

Scalable Spread Spectrum Watermarking

Spread spectrum watermarking techniques (section 2.1.5, page 26) are highly robust to processing-induced distortion. In particular, they have been shown to survive the loss of a substantial portion of the watermark data as a result of resizing or compression [61, 30, 42, 145]. This suggests that spread spectrum techniques may be a good choice for scalable watermarking. This chapter examines the effects of resolution and quality scaling, using JPEG2000, on spread spectrum watermarks.

The spread spectrum watermarking algorithm used in this chapter is essentially the one developed by Cox et al. [30]. The embedding and detection algorithms are integrated (figures 4.1 and 4.2) with JPEG2000 such that the watermark is embedded into the wavelet coefficients[1] of the JPEG2000 compressed image. In particular, the irreversible Daubechies 9/7 wavelet transform is used, since there is no reason to employ a lossless transform given the watermark itself will cause some degradation to the image. The embedding routine is placed prior to the quantization step of JPEG2000. However, given that quantization may be modelled as an additive noise attack, robustness may be improved if the embedding routine is placed following the quantization step. This will be done for the algorithms of chapters 5 and 6.

There are several advantages to embedding in the wavelet domain, rather than the DCT domain. Firstly, because the image is already transformed into the wavelet domain as part of the JPEG2000 compression process, it eliminates the use of a transform specifically for watermarking, thereby reducing the computational costs [66]. Secondly, the multiresolution nature of the wavelet transform provides a good match for the human visual system [118], allowing improved imperceptibility. Finally, by matching the watermarking domain to the compression domain the robustness of the watermark is increased [206].

[1] Although the DCT transform is used in the experimental section of [30], Cox et al. also suggest embedding the wavelet transform domain as an option for their algorithm.

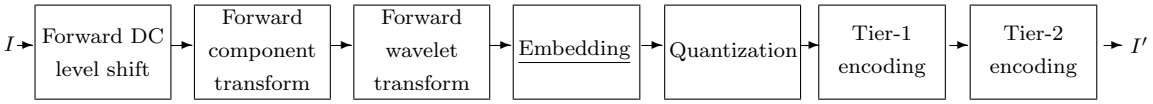


Figure 4.1: Watermark embedding is performed during JPEG2000 compression, immediately preceding the quantization step.

Figure 4.2: Watermark detection is performed during JPEG2000 decompression, immediately following the dequantization step.

The basic embedding and detection algorithms considered in this chapter are as follows:

Embed

Input: I, sk, N, α

Output: I'

- Transform I into the discrete wavelet domain using JPEG2000
- Use the secret key sk to generate a pseudorandom sequence $W = (w_1, w_2, ..., w_N)$, where $w_i \in \mathbb{R}$ is normally distributed with mean 0 and variance 1
- Set $I' = I$
- Select N coefficients $X = (x_1, x_2, ..., x_N)$ from I
- Select corresponding coefficients $X' = (x'_1, x'_2, ..., x'_N)$ from I'
- Modify I' so that $x'_i = x_i + \alpha w_i \quad 1 \leq i \leq N$

Detect

Input: I^*, I, sk, N, α

Output: $\{True\}$ or $\{False\}$

- Transform I and I^* as for Embed
- Use sk to generate $W = (w_1, w_2, ..., w_N)$ as for Embed
- Select $X = (x_1, x_2, ..., x_N)$ from I as for Embed
- Select corresponding coefficients, i.e. those having the same indices in the transformed image, $(x^*_1, x^*_2, ..., x^*_N)$ from I^*
- Extract $W^* = (w^*_1, w^*_2, ..., w^*_N)$ where $w^*_i = \frac{1}{\alpha}(\frac{x^*_i}{x_i} - 1)$
- Calculate $\gamma(I^*, I, sk) = \text{sim}(W, W^*) = \frac{W \cdot W^*}{\sqrt{W^* \cdot W^*}}$
- Compare $\gamma(I^*, I, sk)$ to a detection threshold T and output

$$\text{Output} \begin{cases} \{True\} & \text{if } \gamma(I^*, I, sk) > T \\ \{False\} & \text{otherwise} \end{cases}$$

Watermark detection is non-blind and occurs by comparing the similarity $\gamma = \text{sim}(W, W^*)$, between the candidate and extracted watermarks, to a detection threshold T, which is typically chosen according to a false positive model. In section 4.1 the impact of resolution and quality scaling on the detection process, and the possibility of adapting the false positive model to achieve better watermark detection in scaled images is examined.

Note that, in the watermarking algorithm presented above, the method of selecting the set of coefficients X, which are to be watermarked, has not been specified. While Cox et al. selected the highest-magnitude coefficients excluding the DC coefficient, many other selection methods have been proposed. Section 4.2 explores the effect of coefficient selection on watermark scalability, using experimental evaluation of spread spectrum watermarks with different coefficient selection schemes.

In section 4.3, the watermarking algorithm is extended to include human visual system adaptation, in an attempt to provide both resolution and quality scalability. The HVS adaptive algorithm includes the contrast sensitivity functions developed by Nadenau [131] and a new texture scoring algorithm, allowing the strength of the watermark to be increased where the HVS is less sensitive. The section includes an experimental comparison of scalability, between the HVS adaptive algorithm and the corresponding non-adaptive algorithm from the preceding section, as well as a wavelet implementation of the algorithm of Cox et al. [30] and one of the HVS adaptive spread spectrum algorithm of Xia et al. [210].

4.1 FalsePositive Modelling for Spread Spectrum Watermarks

4.1.0.4 False Positive Modelling

Detection of a spread spectrum watermark occurs by calculating a correlation based detection statistic γ such as similarity $\text{sim}(W, W^*)$, between the candidate watermark W and the extracted watermark W^*, and comparing it to a detection threshold T. If the detection statistic exceeds the threshold, then the extracted watermark is a sufficiently close match to the candidate watermark that watermark is deemed present.

As was discussed in section 2.1.1 (page 13), the watermark detection process may result in one of four possible outcomes. Two of these outcomes, true positive and true negative, represent correct decisions (that the watermark is present or absent, respectively). The remaining two outcomes represent detection errors. These are false positive error, where an absent watermark is deemed present, and false negative error, where a present watermark is deemed absent.

The choice of the detection threshold T determines the trade-off between these two types of error. A smaller threshold provides a lower rate of false negative error at the expense of a higher rate of false positive error, and vice-versa.

Ideally, for any given application the threshold selected minimizes the weighted sum of the two error rates, where the weights are determined according to the relative cost of each type of error occurring. However, such selection requires accurate estimates of both the expected false positive and false negative error rates over a range of potential threshold values.

To obtain accurate estimates of these error rates by experimental means, it would be necessary to perform the embedding and extraction procedures, and calculate the value of the detection statistic for each of a vast number of different images and different watermarks. This makes experimentally obtaining such estimates an expensive proposition, so analytical models of the error rates are generally employed.

Furthermore, it is substantially easier to model the false positive error rate than it is to model the false negative error rate and, for most applications, a false positive error is deemed more costly than a false negative error. The result of this is that, rather than model both types of error and select a threshold which minimizes the weighted sum, it is common to model only the false positive error rate and select the threshold which provides the maximum false positive error allowable for the application. This approach minimizes the false negative error rate of the algorithm, subject to the application-specific false positive error constraints.

A false positive is generated whenever the similarity between the candidate and extracted watermark exceeds the detection threshold $\mathrm{sim}(\mathrm{W}, \mathrm{W}^*) > \mathrm{T}$ but the candidate watermark was not embedded in the image I^*. In this section, two different false positive models for spread spectrum watermark detection are described: the Gaussian model (section 4.1.0.5) and the hypersphere model (section 4.1.0.6).

When modelling the detection process it is not necessary to directly involve the watermarking technique used, provided one can make some assumptions about the distributions of the candidate and extracted vectors. The candidate and extracted vectors assumed here are chosen such that they will satisfy the assumptions of both false positive models.

Let the candidate watermark be an N-dimensional vector

$$W = (w_1, w_2, \ldots, w_N) \tag{4.1}$$

and let the extracted watermark be an N-dimensional vector

$$Z = (z_1, z_2, \ldots, z_N) \tag{4.2}$$

with the following conditions:

- each element of W is independently drawn from a Gaussian distribution $\mathrm{P}(w)$ with zero mean and unit variance[2] and

- each element of Z is independently drawn from a Gaussian distribution $\mathrm{P}(z)$ with zero mean but unknown variance.

Because the candidate and extracted vectors are chosen independently, the extracted watermark Z is completely unrelated[3] to the candidate watermark W and thus any positive detection will be a false positive. The false positive rate will thus be the proportion of vector pairs W and Z for which the detection statistic exceeds the threshold.

4.1.0.5 The Gaussian Model

The Gaussian model is a common false positive model for spread spectrum watermarks, and was the model applied in the paper by Cox et al. [30]. This model hinges upon the assumption that the detection statistic $\mathrm{sim}(W, Z)$ has a Gaussian distribution.

To fully specify the distribution of the detection statistic, since a Gaussian shape is assumed, only the mean μ_{sim} and variance σ^2_{sim} are required. Once the distribution is specified, the probability of obtaining a detection statistic that exceeds a given threshold T can be determined.

To calculate the values of μ_{sim} and σ^2_{sim} we let

$$\begin{aligned} \hat{Z} &= \frac{Z}{||Z||} \\ &= \frac{Z}{\sqrt{Z \cdot Z}} \end{aligned} \tag{4.3}$$

and thus

$$\begin{aligned} \mathrm{sim(W, Z)} &= \frac{W \cdot Z}{\sqrt{Z \cdot Z}} \\ &= W \cdot \hat{Z} \\ &= \sum_{i=1}^{N} w_i \hat{z}_i. \end{aligned} \tag{4.4}$$

[2] A normal distribution is not strictly necessary as any radially symmetric distribution will satisfy the assumptions of both models; however, the Gaussian model will be more accurate if the distribution of candidate and extracted vectors is normal.

[3] The notation Z, rather than W^*, is used for the extracted watermark in this section to emphasize that the extracted watermark is certainly, rather than merely possibly, *not* derived from W.

As all the watermark elements w and z are independent and zero mean, calculation of the mean of the similarity distribution is straightforward:

$$\begin{aligned}\mu_{\text{sim}} &= \mu(\sum_{i=1}^{N} w_i \hat{z}_i) \\ &= N\mu_{w\hat{z}} \\ &= N\mu_w\mu_{\hat{z}} \\ &= 0.\end{aligned} \tag{4.5}$$

So all that remains is to calculate the variance:

$$\begin{aligned}\sigma^2_{\text{sim}} &= N\sigma^2_{w\hat{z}} \\ &= N(\mu_{w^2}\mu_{\hat{z}^2} - \mu^2_w\mu^2_{\hat{z}}).\end{aligned} \tag{4.6}$$

Given that w has zero mean and unit variance, we can easily obtain the value of μ_{w^2}, allowing the simplification of σ^2_{sim}:

$$\begin{aligned}\mu_{w^2} &= \sigma^2_w + \mu^2_w \\ &= 1^2 + 0^2 \\ &= 1\end{aligned} \tag{4.7}$$

$$\begin{aligned}\sigma^2_{\text{sim}} &= N(\mu_{w^2}\mu_{\hat{z}^2} - \mu^2_w\mu^2_{\hat{z}}) \\ &= N(1\mu_{\hat{z}^2} - 0^2\mu^2_{\hat{z}}) \\ &= N\mu_{\hat{z}^2}.\end{aligned} \tag{4.8}$$

We cannot use the same procedure to obtain the value of $\mu_{\hat{z}^2}$ because, while $\mu_{\hat{z}} = 0$, the variance $\sigma^2_{\hat{z}}$ is unknown. So we must use an estimate for the value of $\mu_{\hat{z}^2}$ in our calculation of σ^2_{sim}:

$$\begin{aligned}\sigma^2_{\text{sim}} &= N\mu_{\hat{z}^2} \\ &\approx N\frac{\sum_{i=0}^{N}\hat{z}_i^2}{N} \\ &= \sum_{i=0}^{N}\hat{z}_i^2 \\ &= \sum_{i=0}^{N}(\frac{z_i}{\sqrt{Z\cdot Z}})^2 \\ &= \frac{\sum_{i=0}^{N}{z_i}^2}{Z\cdot Z} \\ &= \frac{Z\cdot Z}{Z\cdot Z} \\ &= 1.\end{aligned} \tag{4.9}$$

Thus the distribution of $\mathrm{sim}(W, Z)$ is Gaussian with zero mean and unit variance (standard normal). To calculate the false positive rate FP, we must determine, as a function of T, the proportion of watermarks W, Z that will satisfy

$$\mathrm{sim}(W, Z) > T. \tag{4.10}$$

This will be the area beneath a Gaussian curve of zero mean and unit variance that lies to the right of the threshold value T. Thus,

$$\mathrm{FP}(T) = \int_T^\infty \frac{1}{\sqrt{2\pi}} e^{\frac{-\mathrm{u}^2}{2}}. \tag{4.11}$$

Note that the estimated false positive rate depends only on the value of the threshold T, as π and e are constants and u is a variable of integration.

4.1.0.6 The Hypersphere Model

An alternative false positive model, developed by Miller and Bloom [127] does not assume the shape of the distribution of the detection statistic. Instead, it attempts to model the detection statistic based on the geometric interpretation of normalized correlation.

The hypersphere model requires that the extracted watermark Z have a radially symmetric distribution, so the probability of obtaining a given value of Z depends only on the magnitude of Z and not its direction. The same definition for Z as previously, where each element z_i of Z is independently drawn from a zero-mean Gaussian distribution $\mathrm{P}(z)$, gives a distribution of Z that is radially symmetric.

The derivation presented in [127] is given for the normalized correlation detection statistic

$$R_{\mathrm{NC}}(W, Z) = \frac{W \cdot Z}{\sqrt{W \cdot W}\sqrt{Z \cdot Z}}; \tag{4.12}$$

for the similarity detection statistic some adjustments must be made.

For a given threshold T a false positive occurs when

$$\mathrm{sim}(W, Z) > T. \tag{4.13}$$

The random vector elements w_i have unit variance and the candidate and extracted vectors have dimension N, so the similarity detection statistic $\mathrm{sim}(W, Z)$ is related to normalized correlation by a factor of approximately $\sqrt{N}$.

$$\begin{aligned} \mathrm{sim}(W, Z) &= \frac{W \cdot Z}{\sqrt{Z \cdot Z}} \\ &= \sqrt{W \cdot W}\frac{W \cdot Z}{\sqrt{W \cdot W}\sqrt{Z \cdot Z}} \\ &\approx \sqrt{N}\frac{W \cdot Z}{\sqrt{W \cdot W}\sqrt{Z \cdot Z}} \\ &= \sqrt{N} R_{\mathrm{NC}}(W, Z) \end{aligned} \tag{4.14}$$

Thus a false positive occurs when

$$R_{\mathrm{NC}}(W, Z) > T_{\mathrm{NC}} \tag{4.15}$$

where

$$T_{\mathrm{NC}} = \frac{T}{\sqrt{N}}. \tag{4.16}$$

If we let $\hat{Z}$ be defined as before, then $\hat{Z}$ lies on the surface of an N-dimensional hypersphere of radius 1 and furthermore, because the distribution of Z is radially symmetric and the magnitude of $\hat{Z}$ is fixed, any particular value of $\hat{Z}$ is equally likely.

But, since Z and $\hat{Z}$ have identical directions, they form the same angle with W, so

$$R_{\mathrm{NC}}(W, Z) = R_{\mathrm{NC}}(W, \hat{Z}). \tag{4.17}$$

Thus a false positive occurs when

$$R_{\mathrm{NC}}(W, \hat{Z}) > T_{\mathrm{NC}}. \tag{4.18}$$

Now the geometric interpretation of the normalized correlation statistic is

$$R_{\mathrm{NC}}(W, \hat{Z}) = \cos(\theta) \tag{4.19}$$

where $\theta$is the angle between the vectors W and $\hat{Z}$. The watermark will be detected when the two vectors are sufficiently close that the angle between them is smaller than T_θ

$$R_{\mathrm{NC}}(W, \hat{Z}) > T_{\mathrm{NC}} \quad \Leftrightarrow \quad |\theta| < T_\theta \tag{4.20}$$

where

$$\begin{aligned} T_\theta &= \cos^{-1}(T_{\mathrm{NC}}) \\ &= cos^{-1}(\frac{T}{\sqrt{N}}) \end{aligned} \tag{4.21}$$

This occurs when $\hat{Z}$ lies on the intersection of the surface of the hypersphere and an N-dimensional hypercone about W with half-angle θ.

The false positive probability is the chance that $\hat{Z}$ lies on the cap of the described hypercone. Given that all values of $\hat{Z}$ on the hypersphere are equally likely, this probability is the area of the cap divided by the area of the hypersphere.

The formula for the area of the cap at radius 1 of an N-dimensional hypercone is

$$Cap(N, \theta) = S_{N-1} I_{N-2}(\theta) \tag{4.22}$$

where

$$S_N = \frac{N\pi^{\lfloor N/2 \rfloor}}{\lfloor N/2 \rfloor} \tag{4.23}$$

and

$$I_N(\theta) = \int_0^\theta \sin^n u \, du. \tag{4.24}$$

Finally, an N-hemisphere is an N-dimensional hypercone with half-angle $\frac{\pi}{2}$ so the area of the hypersphere is twice that, giving a false positive probability of:

$$\begin{aligned} \mathrm{FP}(T, N) &= \frac{Cap(N, \theta)}{2Cap(N, \pi/2)} \\ &= \frac{S_{N-1} I_{N-2}(\theta)}{2 S_{N-1} I_{N-2}(\pi/2)} \\ &= \frac{1}{2} \frac{I_{N-2}(\theta)}{I_{N-2}(\pi/2)} \\ &= \frac{1}{2} \frac{\int_0^\theta \sin^{N-2} u \, du}{\int_0^{\pi/2} \sin^{N-2} u \, du} \end{aligned} \tag{4.25}$$

where

$$\theta = cos^{-1}(\frac{T}{\sqrt{N}}). \tag{4.26}$$

In this case, the false positive rate depends on the threshold T and the dimensionality N of the watermark vectors.

4.1.0.7 The Effects of Resolution Scaling on a Spread Spectrum Watermark

When resolution scaling is applied to an image, all coefficients contained within the discarded resolutions are lost, while the remaining coefficients are unchanged. For most embedding schemes, in which each watermark element is embedded entirely within a single coefficient, this results in the loss of large numbers of whole watermark elements from the extracted vector W^*, leaving the remaining watermark elements untouched.

Any element w_i^*, of the extracted vector W^*, that was embedded in a portion of the content which is entirely unavailable is assigned the value zero, preventing both it and the corresponding candidate element w_i from contributing to the calculation of the similarity statistic. If we let k denote the number of watermark elements that have been lost, the expected maximum value of the similarity statistic, where the candidate and extracted vectors matched exactly, drops from $\sqrt{N}$ to $\sqrt{N-k}$. Thus valid watermarks are less likely to produce a similarity value that exceeds T, i.e. are more difficult to detect, after resolution scaling.

4.1.0.7.1 The Gaussian Model and Resolution Scaling

As the expected maximum of the similarity statistic is reduced, we would like a corresponding reduction in the threshold value so that the probability of detecting a valid

watermark is partially restored but the maximum allowable false positive rate is not exceeded.

However, this is not possible using the Gaussian model as the false positive rate is dependent only on the threshold value, and the initial threshold value is set to produce the maximum allowable false positive rate. Thus, going by the Gaussian model, any reduction in the initial threshold will cause the estimated false positive rate to exceed the maximum allowable rate.

4.1.0.7.2 A Variable Threshold for Resolution Scaling

Note that when k elements of W^* are assigned the value zero, the corresponding elements in the candidate watermark W have no effect on the similarity value. Because those k elements are not comparable, the dimensionality of both the extracted vector W^* and the candidate vector W are effectively reduced by k. In the hypersphere model, unlike the Gaussian model, the false positive rate depends on not only the threshold but also the watermark dimensionality. So a slightly reduced threshold can be used for resolution scaled images while maintaining an acceptable false positive rate.

Consider the hypersphere model for the watermark vectors W and Z in the case where only the element for the ith dimension is missing. In this case, the acceptance region includes not only those vectors which lie on the hypercone surface, but also those which would have done, were there not an overly large distance between w_i and z_i. This increases the size of the acceptance area, requiring a decrease in the angle θ to maintain the same false positive rate.

The area in question extends such that the proportion no longer diminishes with distance in dimension i. That is, that a cross section of the N-dimensional hypersphere at any value of z_i will show the same proportion of accepted vectors as did that through $z_i = w_i$. But this is simply that of an $(N-1)$-dimensional hypercone cap with half-angle alpha over an $(N-1)$-dimensional hypersphere. Thus, to retain the desired false positive rate, the threshold T_{NC} must be increased to the value which would be used had our original vector dimensionality been $N-1$ rather than N.

This appears to conflict with our earlier aim of a decrease in threshold value. However, although the threshold T_{NC} for a normalized correlation statistic increases, the similarity threshold T is determined by multiplying T_{NC} by the expected magnitude of W, and (because the ith element has been assigned the value 0) the expected magnitude of W is now $\sqrt{N-1}$ rather than $\sqrt{N}$. Multiplying the slightly increased value of T_{NC} by the reduced factor $\sqrt{N-1}$ produces a slight overall decrease in the threshold T while not exceeding the allowable maximum false positive rate.

If k of the N watermark elements of Z have been set to zero,

$$Z^{\mathcal{R}} = (z_1, z_2, \ldots, z_{N-k}, \underbrace{0, ..., 0}_{k}) \tag{4.27}$$

then the similarity calculation between W and $Z^{\mathcal{R}}$ is identical to that between two vectors

$$W_{N-k} = (w_1, w_2, \ldots, w_{N-k}) \tag{4.28}$$

and

$$Z_{N-k} = (z_1, z_2, \ldots, z_{N-k}), \tag{4.29}$$

reduced in dimensionality by deleting the k dimensions corresponding to the zeroed watermark elements

$$\mathrm{sim}(W, Z^{\mathcal{R}}) > T \quad \Leftrightarrow \quad \mathrm{sim}(W_{N-k}, Z_{N-k}) > T. \tag{4.30}$$

This is

$$\begin{aligned} \mathrm{sim}(W_{N-k}, W_{N-k}) &= \frac{W_{N-k} \cdot Z_{N-k}}{\sqrt{Z_{N-k} \cdot Z_{N-k}}} \\ &\approx \sqrt{N-k} \frac{W_{N-k} \cdot Z_{N-k}}{\sqrt{W_{N-k} \cdot W_{N-k}} \sqrt{Z_{N-k} \cdot Z_{N-k}}} \\ &= \sqrt{N-k} R_{\mathrm{NC}}(W_{N-k}, Z_{N-k}) \end{aligned} \tag{4.31}$$

The vector Z_{N-k} will occur with probability density function

$$\mathrm{P}_{N-k}(z_1, z_2, ..., z_{N-k}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} \mathrm{P}(z_1, z_2, ..., z_N)\, dz_{N-k+1}\, dz_{N-k+2} \ldots dz_N. \tag{4.32}$$

Because integration occurs over the entire range for each missing dimension, the function $\mathrm{P}_{N-k}(Z_{N-k})$ retains the radial symmetry property of $\mathrm{P}(Z)$, so the hypersphere model can be applied. Thus a false positive occurs when

$$\begin{aligned} \mathrm{FP}(T, N, k) &= \frac{Cap(N-k, \theta^{\mathcal{R}})}{2Cap(N-k, \pi/2)} \\ &= \frac{S_{N-k-1} I_{N-k-2}(\theta)}{2 S_{N-k-1} I_{N-k-2}(\pi/2)} \\ &= \frac{1}{2} \frac{I_{N-k-2}(\theta) I_{N-k-2}(\pi/2)}{} \\ &= \frac{1}{2} \frac{\int_0^{\theta} \sin^{N-k-2} u \, du}{\int_0^{\pi/2} \sin^{N-k-2} u \, du} \quad . \end{aligned} \tag{4.33}$$

where

$$\theta^{\mathcal{R}} = cos^{-1}(\frac{T}{\sqrt{N-k}}). \tag{4.34}$$

4.1.0.8 The Effects of Quality Scaling on the Spread Spectrum Watermark

When quality scaling is applied to an image I^*, from each coefficient v^* a number $m \geq 0$ of least significant magnitude bits are lost, giving the quality scaled coefficient

$$v^{\mathcal{Q}} = sign(v^*) \times \begin{cases} \left\lfloor \frac{|v^*|}{2^m} \right\rfloor 2^m + \lfloor 2^m \mathrm{r} \rfloor & \left\lfloor \frac{|v^*|}{2^m} \right\rfloor \neq 0 \\ 0 & \left\lfloor \frac{|v^*|}{2^m} \right\rfloor = 0. \end{cases} \tag{2.20a}$$

For our spread spectrum watermarking algorithm, in which each watermark element is embedded entirely within a single coefficient, this results in an extracted watermark vector

$$W^{\mathcal{Q}} = (w_1^{\mathcal{Q}}, ..., w_N^{\mathcal{Q}}) \tag{4.35}$$

in which each element is

$$w^{\mathcal{Q}} = \frac{1}{\alpha} \text{sign}(v^* + \alpha w^*) \left\lfloor \frac{|v^* + \alpha w^*|}{2^m} \right\rfloor 2^m - \frac{v^*}{\alpha} + \begin{cases} \lfloor \frac{2^m \mathrm{r}}{\alpha} \rfloor & \text{if } \left\lfloor \frac{|v^* + \alpha w^*|}{2^m} \right\rfloor \neq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{4.36}$$

Given the original image coefficients $v_1 \ldots v_N$ and a watermark strength parameter α, and provided the number of bits $m_1 \ldots m_N$ missing from each coefficient is known[4], the probability of obtaining an extracted watermark $Z^{\mathcal{Q}}$ from a quality scaled image can be expressed using the probability distribution of the extracted watermark Z from an unscaled image (as the integral of $\mathrm{P}(Z)$ over all watermarks Z that would result in $Z^{\mathcal{Q}}$ after quality scaling).

$$\mathrm{P}(z_1^{\mathcal{Q}}, z_2^{\mathcal{Q}}, ...z_N^{\mathcal{Q}}) = \int_{L_1}^{U_1} \int_{L_2}^{U_2} ... \int_{L_N}^{U_N} \mathrm{P}(z_1, z_2, ..., z_N)\, dz_1\, dz_2\, ...\, dz_N \tag{4.37a}$$

where

$$L_i = \begin{cases} \frac{-v_i + \lfloor \frac{|v_i + \alpha z_i^{\mathcal{Q}}|}{2^{m_i}} \rfloor 2^{m_i}}{\alpha} & z_i^{\mathcal{Q}} = \frac{-v_i + \left\lfloor \frac{|v_i + \alpha z_i^{\mathcal{Q}}|}{2^{m_i}} \right\rfloor 2^{m_i} + \lfloor 2^{m_i} \mathrm{r} \rfloor}{\alpha} \\ \frac{-v_i - 2^{m_i}}{\alpha} & z_i^{\mathcal{Q}} = -\frac{v_i}{\alpha} \\ \frac{-v_i - \left(\lfloor \frac{|v_i + \alpha z_i^{\mathcal{Q}}|}{2^{m_i}} \rfloor + 1 \right) 2^{m_i}}{\alpha} & z_i^{\mathcal{Q}} = \frac{-v_i - \left(\left\lfloor \frac{|v_i + \alpha z_i^{\mathcal{Q}}|}{2^{m_i}} \right\rfloor 2^{m_i} + \lfloor 2^{m_i} \mathrm{r} \rfloor \right)}{\alpha} \\ 0 & \text{otherwise} \end{cases} \tag{4.37b}$$

[4] It is possible to determine the number of bits missing from a given coefficient in a JPEG2000 image using information obtained during the decoding process, before watermark detection occurs (section 5.1.6, page 153).

and

$$U_i = \begin{cases} \frac{-v_i+\left(\lfloor\frac{|v_i+\alpha z_i^{\mathcal{Q}}|}{2^{m_i}}\rfloor+1\right)2^{m_i}}{\alpha} & z_i^{\mathcal{Q}} = \frac{-v_i+\left\lfloor\frac{|v_i+\alpha z_i^{\mathcal{Q}}|}{2^{m_i}}\right\rfloor 2^{m_i}+\lfloor 2^{m_i}\mathrm{r}\rfloor}{\alpha} \\ \frac{-v_i+2^{m_i}}{\alpha} & z_i^{\mathcal{Q}} = -\frac{v_i}{\alpha} \\ \frac{-v_i-\lfloor\frac{|v_i+\alpha z_i^{\mathcal{Q}}|}{2^{m_i}}\rfloor 2^{m_i}}{\alpha} & z_i^{\mathcal{Q}} = \frac{-v_i-\left(\left\lfloor\frac{|v_i+\alpha z_i^{\mathcal{Q}}|}{2^{m_i}}\right\rfloor 2^{m_i}+\lfloor 2^{m_i}\mathrm{r}\rfloor\right)}{\alpha} \\ 0 & \text{otherwise.} \end{cases} \tag{4.37c}$$

It is important to note that the number of missing bits m_i and the original image coefficient v_i may change substantially as i is changed so, although the probability distribution of z_i is the same for all $1 \leq i \leq N$, the probability distribution of $z_i^{\mathcal{Q}}$ is likely to differ for different positions i in the watermark vector.

This means that, in contrast to that of the resolution scaled vector $Z^{\mathcal{R}}$, the probability distribution $\mathrm{P}(Z^{\mathcal{Q}})$ of the quality scaled watermark vector $Z^{\mathcal{Q}}$ is unlikely to retain the radial symmetry of the underlying probability distribution of the (unscaled) vector Z. Thus the transformation $\hat{Z}^{\mathcal{Q}} = \frac{Z^{\mathcal{Q}}}{||Z^{\mathcal{Q}}||}$ results in vectors on the unit hypersphere that do not necessarily occur with equal probability. This means the probability that $\mathrm{sim}(W, Z^{\mathcal{Q}}) > T$ can no longer be calculated as the area of the hypercone cap divided by the area of the hypersphere, as there is no guarantee that all points on the hypersphere are equally likely.

Let C be the set of all points $Z^{\mathcal{Q}}$ that lie within the infinite N-dimensional hypercone with half angle $\theta = cos^{-1}(\frac{T}{\sqrt{N}})$ about the vector W. The false positive rate for a given candidate vector W is

$$\mathrm{FP}(T, N, Z, \alpha, M, V) = \sum_{Z^{\mathcal{Q}} \in C} \mathrm{P}(Z^{\mathcal{Q}}) \,, \tag{4.38}$$

where $M = \{m_1, m_2, \ldots, m_N\}$, $V = \{v_1, v_2, \ldots, v_N\}$ and α represent the numbers of missing bits, the original coefficient values and the embedding strength respectively.

It may be possible to compute the proportion of points $Z^{\mathcal{Q}}$ in C with non-zero probabilities given W, M, V and α; however, the probabilities for these points need not be the same. Calculating the probability $\mathrm{P}(Z^{\mathcal{Q}})$ of occurrence for every point $Z^{\mathcal{Q}}$ in C, even if a specific probability distribution $\mathrm{P}(Z)$ for the unscaled extracted vector were given, would be prohibitive.

Despite this, it is possible to account for the worst effects of quality scaling by using a similar process to that used in resolution scaling, while ignoring any less-severe effects. For each dimension i of the watermark vector, the effects of quality scaling are classified as either extreme or not, depending on the number of missing bits m_i relative to the coefficient v_i.

- If $2^{m_i} > v_i$ the effect of quality scaling is considered extreme; all coefficient bits have been lost due to scaling, hence no watermark information is recoverable in that

dimension, so $z_i^{\mathcal{Q}}$ is set to 0 to prevent that dimension from contributing to the similarity calculation.

- If $2^{m_i} \leq v_i$ the effect of quality scaling is ignored; $z_i^{\mathcal{Q}}$ is unchanged in the similarity calculation, and the probability of the quality scaled watermark element is assumed to be identical to that of the unscaled element, $P^{\mathcal{Q}}(z_i^{\mathcal{Q}}) = P(z_i)$.

Thus, if exactly k of the N dimensions, $\{i_1, i_2, ..., i_k\}$, satisfy $2^{m_i} > v_i$ then

$$Z^{\mathcal{Q}} = (z_1^{\mathcal{Q}}, ..., z_N^{\mathcal{Q}}) \tag{4.39a}$$

where

$$z_i^{\mathcal{Q}} = \begin{cases} 0 & \text{if } i \in \{i_1, i_2, ..., i_k\} \\ z_i & \text{otherwise} \end{cases} \tag{4.39b}$$

then and $Z^{\mathcal{Q}}$ will occur with probability density function

$$\mathrm{P}(Z^{\mathcal{Q}}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} \mathrm{P}(z_1, z_2, ..., z_N)\, dz_{i_1}\, dz_{i_2}\, ...\, dz_{i_k}. \tag{4.40}$$

Because the integration is over the entire range for each missing dimension, the probability density function $\mathrm{P}(Z^{\mathcal{Q}})$ retains the radial symmetry property of $\mathrm{P}(Z)$ and, hence, the hypersphere model can be applied. Thus a false positive occurs when

$$\begin{aligned} \mathrm{FP}(T, N, k) &= \frac{Cap(N-k, \theta^{\mathcal{Q}})}{2Cap(N-k, \pi/2)} \\ &= \frac{S_{N-k-1} I_{N-k-2}(\theta)}{2 S_{N-k-1} I_{N-k-2}(\pi/2)} \\ &= \frac{1}{2}\, \frac{I_{N-k-2}(\theta) I_{N-k-2}(\pi/2)}{} \\ &= \frac{1}{2}\, \frac{\int_0^{\theta} \sin^{N-k-2} u\, du}{\int_0^{\pi/2} \sin^{N-k-2} u\, du} \quad . \end{aligned} \tag{4.41a}$$

where

$$\theta^{\mathcal{Q}} = cos^{-1}(\frac{T}{\sqrt{N-k}}). \tag{4.41b}$$

4.2 The Effect of Coefficient Selection on Watermark Scalability

An important component of any spread spectrum watermarking algorithm is its coefficient selection method. The coefficient selection method in the paper [30] on which the

watermarks in this chapter are based, is to use the 1000 highest-magnitude non-DC coefficients in the Y component of a YIQ transformed image. The selection of high-magnitude coefficients is designed to place the watermark into the perceptually significant image data, thereby increasing its robustness. The DC coefficient, which represents the lowest frequency image data, is excluded from selection as even small modifications to the DC coefficient may cause unacceptable distortion. Finally, the selection of coefficients exclusively from the Y component ensures robustness to colour-to-greyscale conversion.

Numerous other coefficient selection schemes have been implemented, indeed many proposals for new watermarking algorithms will also contain a new method for coefficient selection. For example, Piva et al. [145], selected DCT coefficients from a greyscale image solely according to frequency; a number of the lowest frequency coefficients were excluded and next lowest frequency coefficients selected for watermarking. Dugad et al. [42] used a threshold based approach, selecting the wavelet coefficients, excluding those in the LL subband, that exceeded one threshold during embedding, and those that exceeded a larger threshold during detection. Kim and Moon [88] also used a threshold based selection scheme, but they calculated a different threshold for each resolution layer and did not exclude the LL subband.

A large number of schemes select coefficients from the luminance component only, as this component is not only unaffected by colour-to-greyscale conversion, but also is typically less altered by compression, due to its perceptual importance. In contrast to this, Fleet and Heeger [50] avoided the luminance component, because the human visual system is sensitive to luminance at the frequencies used for watermarking; instead, they chose the yellow-blue component in order to achieve reduced watermark perceptibility. Similarly, Kutter et al. [97] selected wavelet coefficients from the blue component only of an RGB image, allowing a far larger embedding strength to be achieved without visible distortion to the image than in the luminance only approach. A drawback of this approach is that the perceptual impact of colour variation is less predictable than that of luminance variation, and may result in watermark perceptibility. Sayrol et al. [158] suggested that, to improve watermark detection, all colour channels, in either RGB or a luminance-chrominance colour model, should be used.

In this section, seven distinct coefficient selection schemes, all in the wavelet domain, are considered. These schemes include magnitude, frequency and threshold based approaches as well as colour-based and luminance-only methods. The scalability of each scheme is evaluated experimentally, and the implications of the results on how to best achieve scalability in a spread spectrum watermark are discussed.

4.2.1 Coefficient Selection Schemes

The coefficient selection schemes used in this section are an attempt to provide a wide variety of methods within the wavelet domain, so that the effects of the decisions made in coefficient selection may be examined. In particular, the list includes schemes that exclude the lowest frequency coefficients and those that do not, as well as one that always selects some coefficients from each frequency band. Similarly, it includes schemes that exclude the colour components and those that do not, as well as and one that always selects some coefficients from each component. Finally, a threshold based selection scheme is included.

The following is a list of the coefficient selection schemes, containing both the description of each method and its motivation.

`top`: Embed in the 1000 largest magnitude coefficients, regardless of any other considerations. This scheme has the greatest freedom of coefficient selection and thus the most intrinsic robustness that can be provided when embedding proportionally to coefficient magnitude. However, because there is no restriction on embedding in either the chrominance (C_b and C_r) components or the low-resolution subbands, it risks being perceptible unless the embedding strength α is low.

`nolow`: Embed in the 1000 largest magnitude coefficients, excluding the lowest resolution layer. The lowest frequency subband is often excluded from watermark embedding schemes due to the sensitivity of the human visual system to artifacts caused by the modification of low frequencies.

`lum`: Embed in the 1000 largest magnitude coefficients of the luminance component only. Many spread spectrum watermarking schemes are designed for greyscale images. Embedding in the luminance (Y) component only is a straightforward adaptation of these techniques to colour images. It places the watermark in areas which are less affected during compression and avoids the risk of watermark perceptibility through variations in colour. However, this approach restricts the selection space to one third of the available coefficients.

`lumnl`: Embed in the 1000 largest magnitude coefficients of the luminance component only, excluding the lowest resolution layer. This is the scheme closest to that recommended for colour images in [30] and can be expected to share the advantages and disadvantages of both `lum` and `nolow`.

`res`: Embed in each resolution layer proportionally to the number of coefficients in that resolution. The number of coefficients added by the second resolution is three times that available at the first resolution and each subsequent layer provides four times

that provided by the one before it. Furthermore, the sensitivity to modifications in each resolution is reduced as the resolution layer increases. Thus we can comfortably embed an increasing portion of the watermark in each additional resolution while maintaining quite a high embedding strength.

`comp`: Embed in each colour component proportionally to the number of coefficients in that component. This scheme allows embedding in colour components, which are commercially valuable and may warrant such protection, but it ensures that only one third of the watermark is embedded in any component in an attempt to avoid colour artifacts due to excessive embedding in a single component. However, in images where colour coefficients are not large, this is likely to embed more of the watermark in the colour components than does the unconstrained embedding.

`thresh`: Embed in those coefficients with magnitude greater than a threshold of two fifths of the maximum magnitude coefficient in their associated resolution layer. This selects coefficients which are fairly large for their resolution layer but has less emphasis on the lower resolution layers (which tend to have higher magnitude coefficients) than the unconstrained `top` scheme. Strategies involving coefficient selection thresholds such as this generally do not specify a set watermark dimensionality; however, for comparison purposes, a fixed dimensionality of 1000 is used and embedding is stopped once that length is reached. If less than 1000 coefficients satisfy the selection criterion, the remaining coefficients are selected from the highest magnitude coefficients in the highest resolution.

4.2.2 Experimental Evaluation of Coefficient Selection Schemes

The effect of these different coefficient selection schemes on the scalability of the resulting spread spectrum watermark, is examined using experiments on three classic 512×512 RGB test images: Lena, Mandrill and Peppers.

4.2.2.1 Experimental Framework

Embedding

For each coefficient selection scheme, a spread spectrum watermark using that selection scheme is embedded into a copy of each original image during JPEG2000 compression with 6 resolution layers and 5 quality layers with compression ratios of 0.01, 0.02, 0.04, 0.06 and 0.9999.[5] The embedding strength α is adjusted for each selection method and each image

[5]The first four compression ratios are intended to result in a series of compressed subimages with qualities ranging from 'barely acceptable' to 'good'; the compression ratio of 0.9999 is used to represent the full image.

in order to ensure that the mean squared error between each watermarked image and its corresponding original is 6.5.[6] The process is repeated using 100 different embedding keys, resulting in 300 watermarked images for each scheme.

Attacking

Each watermarked image undergoes JPEG2000 scaling, producing 6 resolution scaled subimages, with the lowest resolution subimage being of size 16×16, and 5 quality scaled subimages with compression ratios 0.01, 0.02, 0.04, 0.06 and 0.9999.

Detection

Watermark detection, using the correct detection key, is applied to each watermarked subimage and the resulting similarity value is recorded. For each coefficient selection scheme, detectability and graceful improvement, as described in section 3.1.2 (page 54), are calculated, for both resolution and quality scalability, using the mean similarity values across all 100 keys. The calculation details are as follows:

The detectability value is simply the average similarity value at the lowest resolution or quality subimage. However, this would result in resolution detectability values of zero for the `nolow` and `lumnl` schemes, both of which specifically exclude the lowest resolution. To avoid disadvantaging these schemes, resolution detectability $\mathcal{D}^{\mathcal{R}}$ for all schemes will be calculated using the second-lowest resolution subimage, $I^{\mathcal{R}_1}$,

$$\mathcal{D}^{\mathcal{R}} = \frac{\sum_{sk=1}^{100} \gamma(I^{\mathcal{R}_1}, I, sk)}{100} \tag{4.42a}$$

$$\mathcal{D}^{\mathcal{Q}} = \frac{\sum_{sk=1}^{100} \gamma(I^{\mathcal{Q}_0}, I, sk)}{100}. \tag{4.42b}$$

The ideal number of watermark elements for layer l is the portion of the 1000 elements that should be embedded in a given layer according to the increase in perceptual quality, measured using PSNR, provided by that layer, relative to the total improvement in perceptual quality of the full image over a mid-grey image I^e. As was the case with the detectability measure, the second-lowest resolution subimage is treated as though it were the lowest resolution subimage, so

$$\iota^{\mathcal{R}_1} = 1000 \frac{P^{\mathcal{R}_1} - P^e}{P^{\mathcal{R}_5} - P^e} \tag{4.43a}$$

$$\iota^{\mathcal{Q}_0} = 1000 \frac{P^{\mathcal{Q}_0} - P^e}{P^{\mathcal{Q}_4} - P^e} \tag{4.43b}$$

[6] A mean squared error of 6.5 corresponds to a peak signal to noise ratio of 40, which should ensure visual imperceptibility.

for the lowest layers, and

$$\iota^{\mathcal{R}_l} = 1000\frac{P^{\mathcal{R}_l} - P^{\mathcal{R}_{l-1}}}{P^{\mathcal{R}_5} - P^e} \tag{4.43c}$$

$$\iota^{\mathcal{Q}_l} = 1000\frac{P^{\mathcal{Q}_l} - P^{\mathcal{Q}_{l-1}}}{P^{\mathcal{Q}_4} - P^e} \tag{4.43d}$$

for all higher layers l.

Because the expected similarity value for an N-element watermark is $\sqrt{N}$, the mean similarity values of sequential, scaled subimages can be used to calculate an equivalent number of whole extracted watermark elements.

$$\epsilon^{\mathcal{R}_1} = \left(\frac{\sum_{sk=1}^{100}\gamma(I^{\mathcal{R}_1}, I, sk)}{100}\right)^2 \tag{4.44a}$$

$$\epsilon^{\mathcal{Q}_0} = \left(\frac{\sum_{sk=1}^{100}\gamma(I^{\mathcal{Q}_0}, I, sk)}{100}\right)^2 \tag{4.44b}$$

for the lowest layers, and

$$\epsilon^{\mathcal{R}_l} = \left(\frac{\sum_{sk=1}^{100}\gamma(I^{\mathcal{R}_l}, I, sk)}{100}\right)^2 - \left(\frac{\sum_{sk=1}^{100}\gamma(I^{\mathcal{R}_{l-1}}, I, sk)}{100}\right)^2 \tag{4.44c}$$

$$\epsilon^{\mathcal{Q}_l} = \left(\frac{\sum_{sk=1}^{100}\gamma(I^{\mathcal{Q}_l}, I, sk)}{100}\right)^2 - \left(\frac{\sum_{sk=1}^{100}\gamma(I^{\mathcal{Q}_{l-1}}, I, sk)}{100}\right)^2 \tag{4.44d}$$

for all higher layers l.

With the resulting graceful improvement values being

$$\mathcal{G}^{\mathcal{R}} = 1 - \frac{\sum_l \frac{(\epsilon^{\mathcal{R}_l} - \iota^{\mathcal{R}_l})^2}{\iota^{\mathcal{R}_l}}}{N(\frac{N}{\iota^{\mathcal{R}_\mathfrak{m}}} - 1)} \tag{4.45a}$$

and

$$\mathcal{G}^{\mathcal{Q}} = 1 - \frac{\sum_l \frac{(\epsilon^{\mathcal{Q}_l} - \iota^{\mathcal{Q}_l})^2}{\iota^{\mathcal{Q}_l}}}{N(\frac{N}{\iota^{\mathcal{Q}_\mathfrak{m}}} - 1)} \tag{4.45b}$$

where $\mathfrak{m}$ is the resolution layer or quality layer, respectively, with the smallest, non-zero ideal value.

4.2.2.2 Results

4.2.2.2.1 Embedding Strengths

During the embedding of each watermark, the embedding strengths were adjusted to ensure a mean squared error of 6.5 between the original and watermarked image. The average embedding strength of a watermarking scheme indicates to what extent that scheme

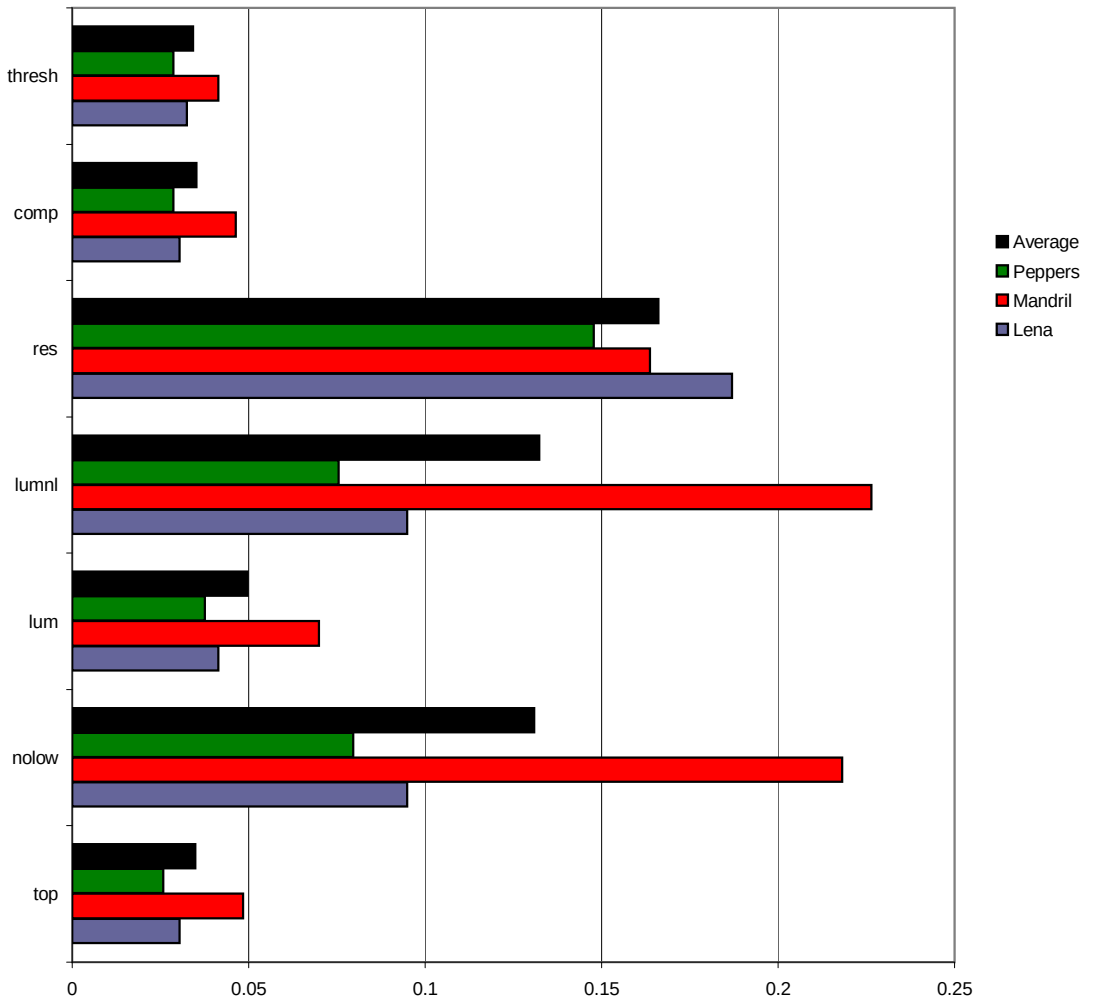


Figure 4.3: Average embedding strengths for the coefficient selection schemes. Schemes that tend to select low-resolution coefficients require lower embedding strengths.

selects perceptually significant coefficients, as schemes which select large numbers of perceptually significant coefficients will require a lower embedding strength than those which do not. The average embedding strength for each scheme is shown in figure 4.3.

As might be expected, this adjustment results in low embedding strengths for `top` and `comp`, a somewhat higher embedding strength for `lum`, and much higher strengths for `nolow`, `lumnl` and `res`. An explanation for this is that the primary factor governing the embedding strength for each scheme is its selection of lower resolution coefficients, as changes to these coefficients typically have a greater effect than do changes to higher resolution coefficients.

The `top` and `comp` schemes are completely unconstrained with regards to resolution and are therefore able to select a large number of coefficients from the lowest resolution layer,

thus requiring a low embedding strength to not exceed the target level of distortion. While `lum` is also unconstrained with regards to resolution, the restriction to a single component ensures that no more than one third of the coefficients in the lowest resolution layer are available for selection, thus it is impossible for the `lum` scheme to select as many of the lowest resolution coefficients as do `top` and `comp`, which results in reduced distortion and thus permits a higher embedding strength. The `nolow`, `lumnl` and `res` schemes, which are drastically restricted in their ability to select low-resolution coefficients, all allow a high embedding strength. Interestingly, the `thresh` scheme shows embedding strengths very near to those of `top` and `comp`, suggesting that this scheme still selects a large proportion of lower resolution coefficients.

4.2.2.2.2 Detectability

Resolution detectability (figure 4.4) is relatively high for the schemes `thresh`, `top` and `comp`, and to some extent `lum`. This confirms that these selection schemes do indeed select a higher proportion of low-resolution coefficients than the `res`, `lumnl` and `nolow` schemes, which all have average $\mathcal{D}^{\mathcal{R}}$ measurements below six[7].

Despite the use of subimage $I^{\mathcal{R}_1}$, rather than subimage $I^{\mathcal{R}_0}$, in the resolution detectability calculation, the performance of both the `lumnl` and `nolow` schemes is poor. This shows that when selection of lowest resolution coefficients is disallowed, the number of high-magnitude coefficients selected from the next-lowest resolution layer is insufficient to make up for the loss.

The mandrill image shows reduced detectability relative to the Lena and peppers images across all schemes but `res`. The mandrill image contains a large amount of fine texture, which results in many high-magnitude coefficients in higher resolutions. Because the number of coefficients to be selected is fixed, at $N = 1000$, the selection of these high-magnitude coefficients is often at the expense of coefficients in the two lowest resolution layers, which causes the reduction in resolution detectability. This does not occur for the `res` scheme because it selects an image-independent number of coefficients at each resolution layer, and thus has almost identical (though insufficient) $\mathcal{D}^{\mathcal{R}}$ values for all three images. A selection scheme similar to `res`, but with a greater emphasis on low-resolution coefficients, may ensure consistently high resolution detectability regardless of image content.

Because low quality layers are composed of the most visually significant parts of the image, one might expect the schemes that are most free to select the more perceptually significant, high-magnitude, low-resolution coefficients to have the highest $\mathcal{D}^{\mathcal{Q}}$ values. However, quality detectability is in fact highest for the `res`, `lumnl` and `nolow` schemes and

[7] A threshold of six will ensure a false positive rate of roughly 1×10^{-9}.

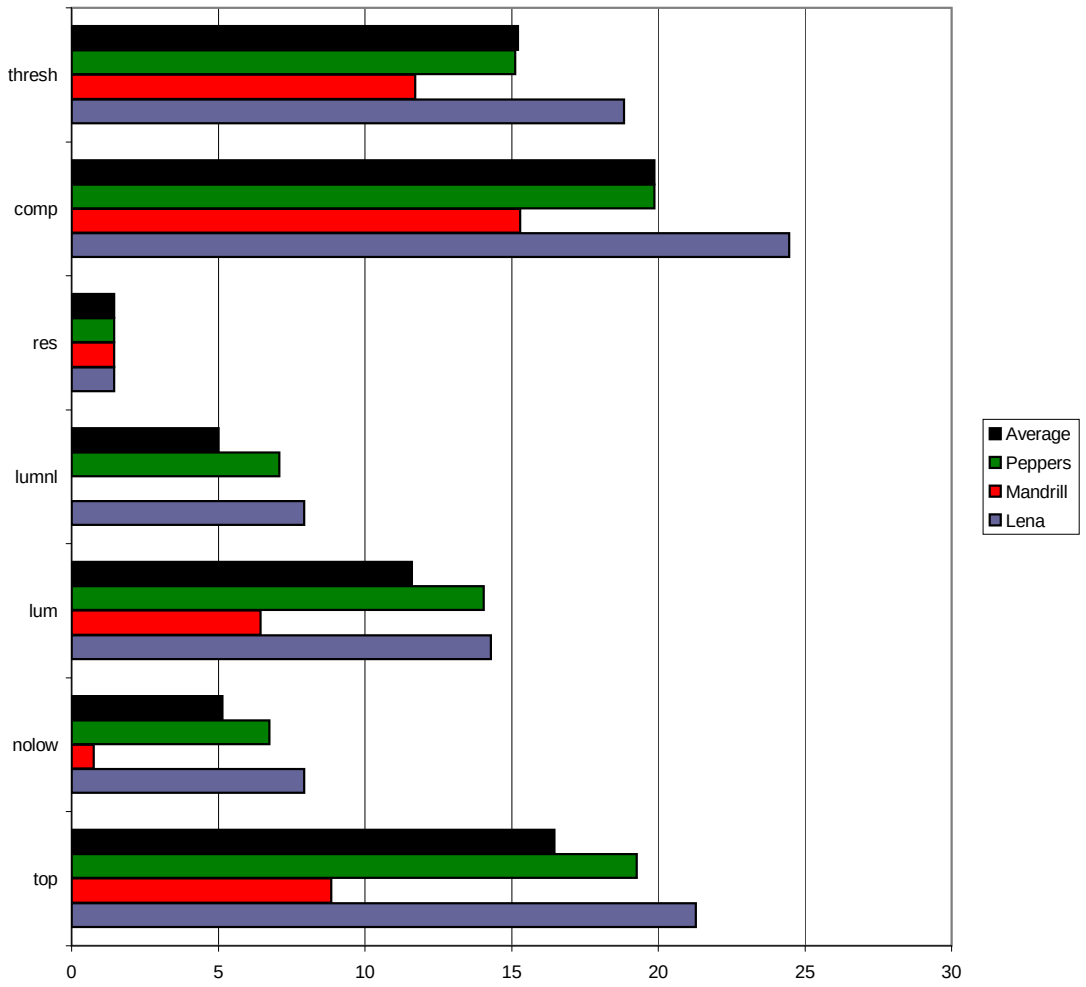


Figure 4.4: Resolution detectability results for the coefficient selection schemes.

relatively low for the `thresh`, `top`, `comp` and `lum` schemes. Indeed, the graph of quality detectability, in figure 4.5, is remarkably similar to the graph of embedding strength, in figure 4.3 (page 106).

This suggests that a high embedding strength, rather than the selection of perceptually significant coefficients, is a major factor in quality detectability. The reason for this is that, of the coefficients that are present in the low quality layers, only the more significant bit planes are included. If the watermark strength is low, the watermark will be largely confined to the less significant bit planes and will thus be largely absent from the lowest quality layer, resulting in low quality detectability.

This is not to say that the coefficient selection has no effect. Firstly, as was discussed previously, coefficient selection has a clear effect on embedding strength. Secondly, the schemes `res`, `lumnl`, and `lum` have lower quality detectability, relative to their unconstrained counterparts (`nolow` and `top`), than their embedding strengths would indicate.

This suggests that if similar embedding strengths could be achieved, a scheme that selects visually significant coefficients without other constraints would be preferable.

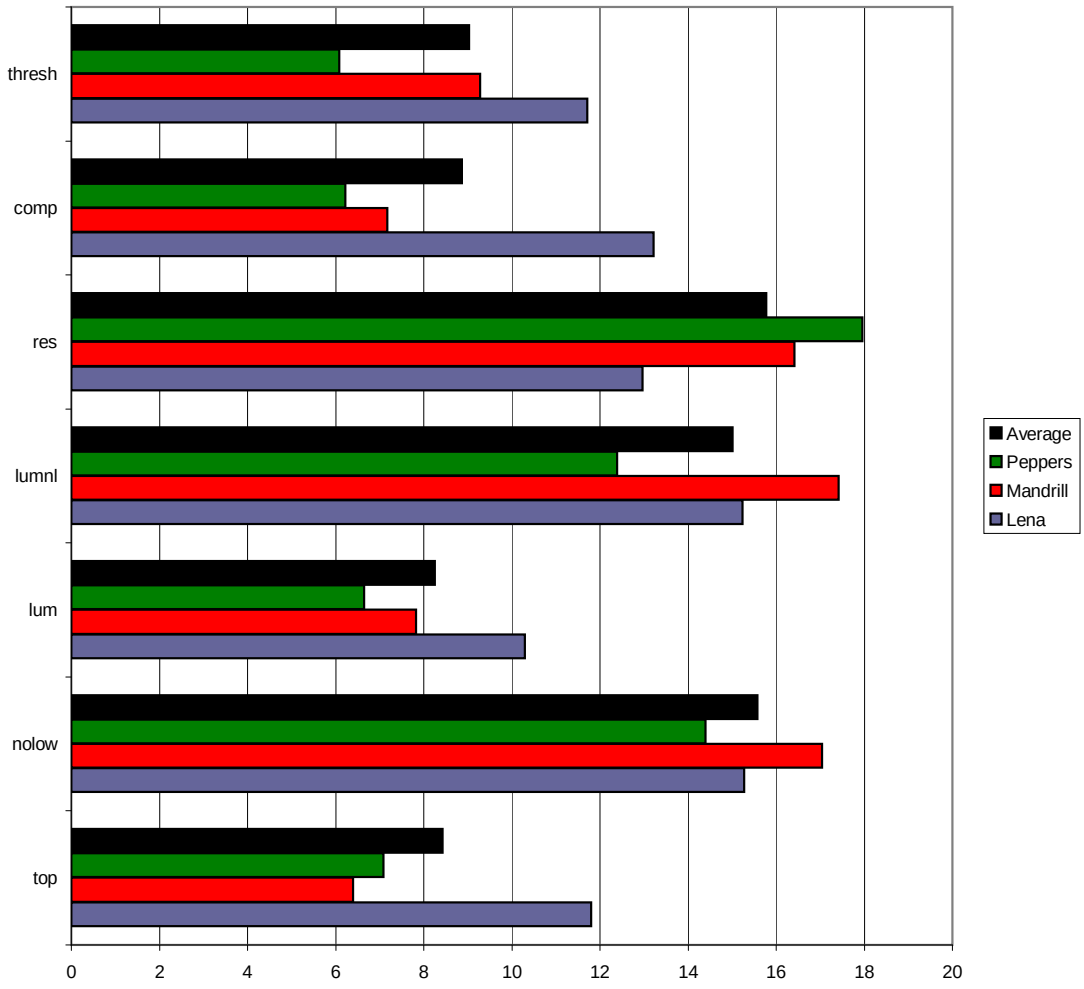


Figure 4.5: Quality detectability results for the coefficient selection schemes.

4.2.2.2.3 Graceful Improvement

Resolution graceful improvement scores (figure 4.6) are generally high. The least resolution-constrained schemes, `top`, `comp` and `lum` achieve the highest overall values.

Despite large differences in resolution detectability between the schemes that exclude the lowest resolution layer and their unconstrained counterparts, the watermark for the `nolow` and `lumnl` schemes is reasonably well distributed among the remaining layers, resulting in resolution graceful improvement values that are not dramatically lower than those of `top` and `lum`. The only schemes that are problematic overall are `thresh` and `res`, which both have $\mathcal{G}^{\mathcal{R}}$ values below 0.8 for two of the three images.

The `res` scheme is highly constrained with regards to resolution and, because the number of coefficients to be selected from each resolution layer is image independent, cannot adapt to images where the ideal number of watermark elements in each resolution layer deviates significantly from these constraints. The `res` scheme does perform well on the mandrill image, which contains a great deal of high-resolution detail, because this image matches well with its emphasis on higher resolution coefficients. It performs poorly on the Lena and peppers images, which consist largely of low-resolution features.

Many of the problems with the `thresh` scheme are caused by the interaction between the chosen threshold and the fixed watermark dimensionality. As was noted in section 4.2.1 (page 102), embedding is stopped once the `thresh` scheme reaches 1000 coefficients. This occurred with both the mandrill and peppers images, indeed embedding in the mandrill image was stopped before any coefficients in the highest resolution layer were reached. For the Lena image this problem did not occur, as the number of coefficients above the threshold was very close to the desired number of coefficients, and the resulting $\mathcal{G}^{\mathcal{R}}$ value is high, suggesting that if the 1000 coefficient constraint were removed, this scheme would have good resolution graceful improvement.

The `comp` scheme has a surprisingly low $\mathcal{G}^{\mathcal{R}}$ value for the Lena image. This occurs because the receipt of the final resolution layer produces a *drop* in similarity for the `comp` scheme, resulting in a negative value for $\epsilon^{\mathcal{R}_5}$ and thus a large deviation from ideal for this layer. Because the `comp` scheme selects equal numbers of coefficients from each component but the majority of the colour information in the Lena image is contained in relatively *few* coefficients, many low-magnitude, chrominance coefficients are selected. This, combined with the low embedding strength, results in a watermark that, in the highest-resolution chrominance components, is removed by quantization, causing the extracted watermark at that resolution layer to consist largely of noise.

The quality graceful improvement values (figure 4.7) are high for all schemes, and the differences between schemes are relatively small. This suggests that, for this spread-spectrum watermarking algorithm, any coefficient selection scheme that focuses on high-magnitude coefficients will have reasonably good quality graceful improvement.

The only clear difference between schemes is the relatively poor performance of the high-embedding-strength schemes on the mandrill image. This occurs because the combination of the selection of high-magnitude coefficients and the use of a high embedding strength overly bias the watermark towards low quality layers. This bias is mitigated in the Lena and peppers images because the low quality layers contain many lowest-resolution coefficients (which, for the high embedding strength schemes, contain little or no watermark data). However, in the mandrill image, in which the low quality layers contain many high-resolution coefficients, too much of the watermark is included in the low quality layers, resulting in a large difference from the ideal at these layers.

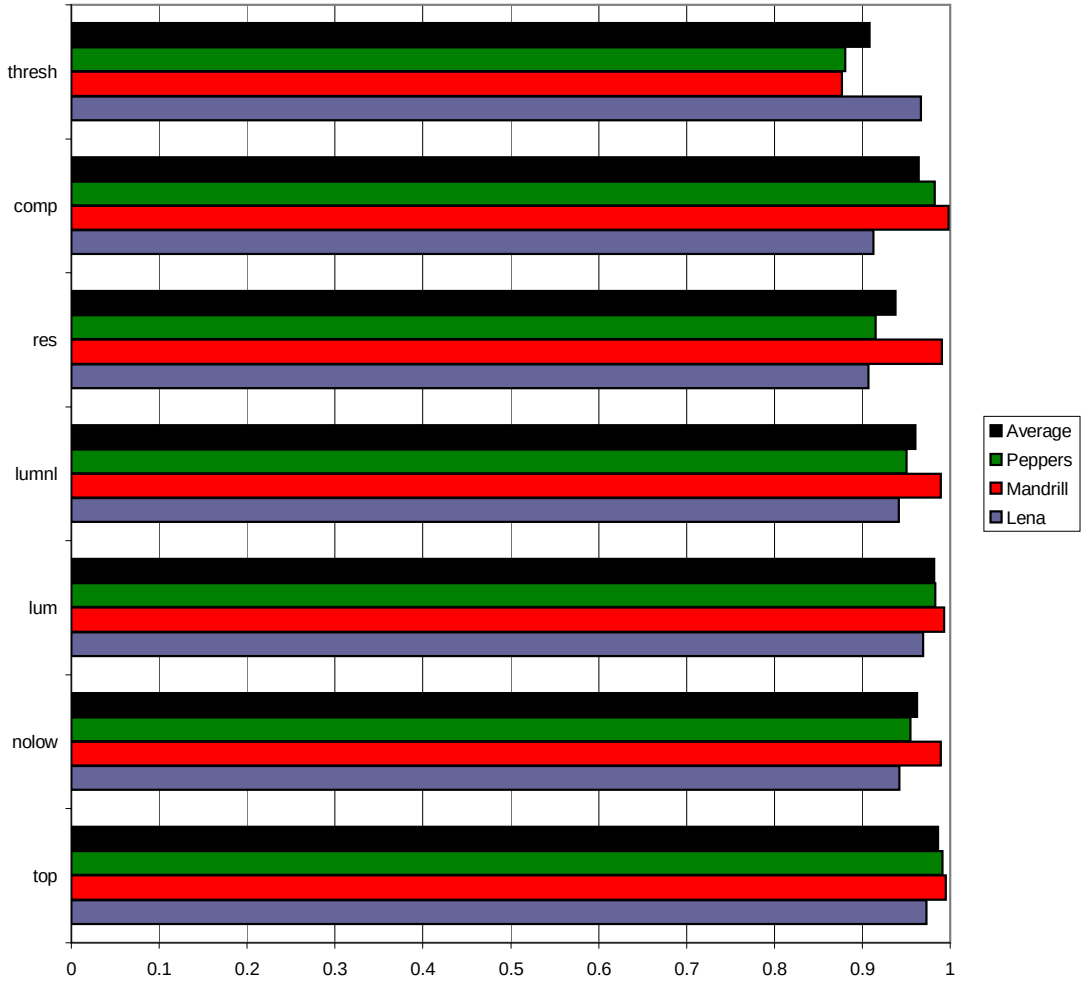


Figure 4.6: Resolution graceful improvement results for the coefficient selection schemes.

4.2.3 The Resolution/Quality Tradeoff

All the coefficient selection schemes considered here perform reasonably well on graceful improvement, thus the main property determining the scalability of the schemes is their detectability.

The examination of the resolution and quality detectability of the various schemes (section 4.2.2.2.2, page 107) found that the `top`, `comp`, `thresh` and `lum` schemes had high resolution detectability but low quality detectability, while the `res`, `lum` and `nolow` schemes had high quality detectability but low resolution detectability (figure 4.8, page 113).

Although the high embedding strengths achievable using the `nolow`, `lumnl` and `res` schemes allow high quality detectability, these high embedding strengths are obtained by disallowing embedding in the lowest resolution layer, resulting in low resolution detectability.

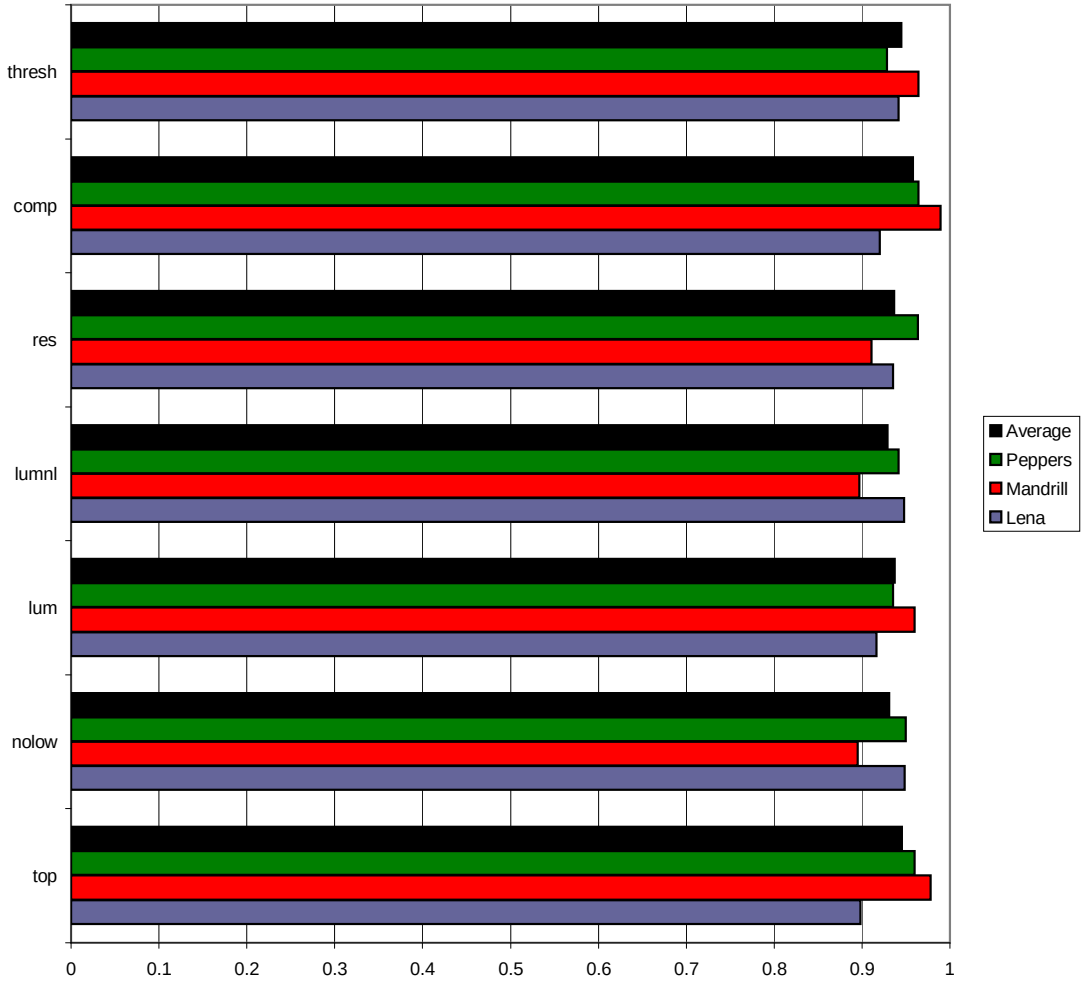


Figure 4.7: Quality graceful improvement results for the coefficient selection schemes.

Conversely, the inclusion of the lowest resolution layer in the selection process, which allows the high resolution detectability of the `top`, `comp`, `thresh` and `lum` schemes, requires a low embedding strength to ensure imperceptibility. This low embedding strength limits the watermark to less significant bits and thus reduces the quality detectability.

This results in a tradeoff between resolution and quality detectability, making it unlikely that a fully scalable watermarking algorithm can be achieved merely by altering the coefficient selection method. The following section considers a method for achieving both resolution and quality scalability in a single watermark.

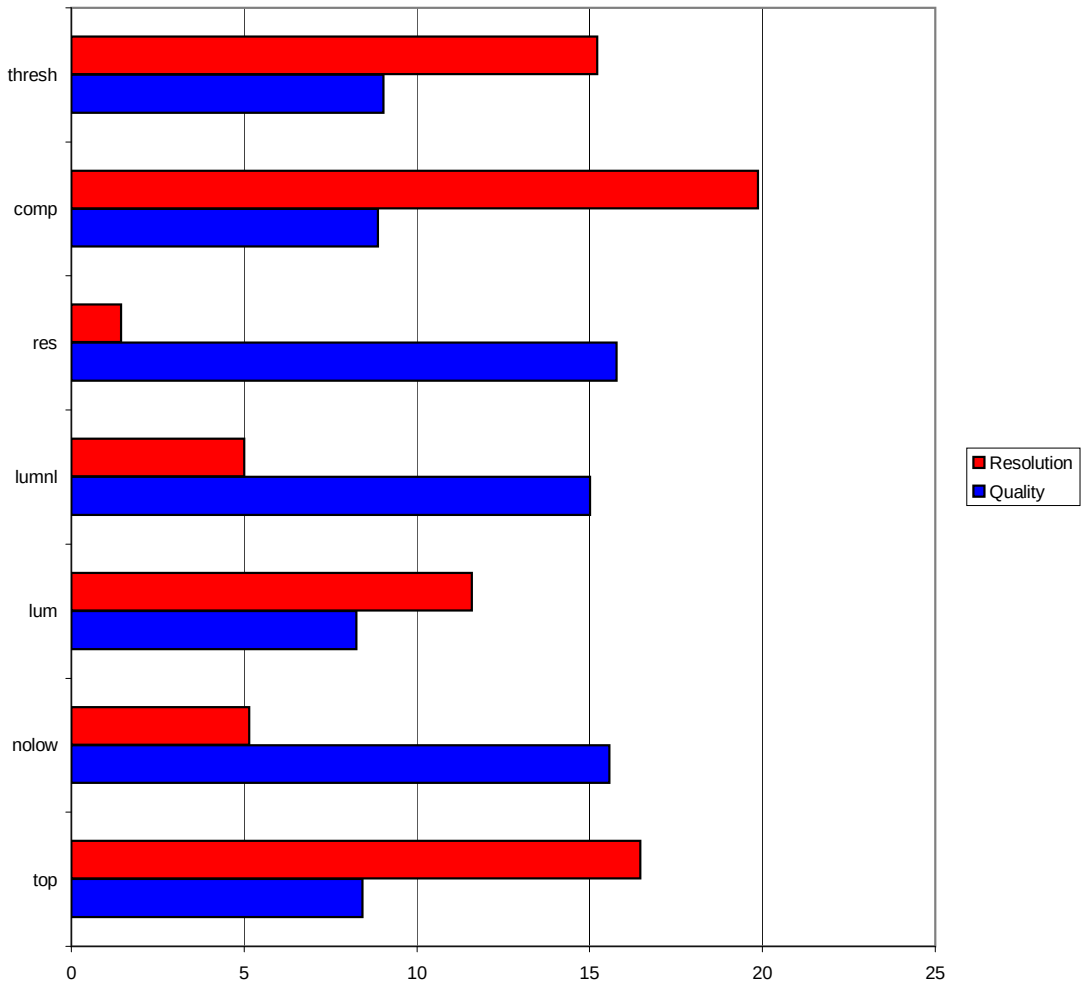


Figure 4.8: The resolution/quality tradeoff: coefficient selection schemes with high resolution detectability have low quality detectability and vice-versa.

4.3 Human Visual System Adaptation to Alleviate the Resolution-Quality Tradeoff

The tradeoff between resolution and quality scalability, observed in section 4.2.3, occurs because quality detectability requires a reasonably high embedding strength (to ensure that the watermark reaches the significant bits that form the lowest quality layer) but the only way to achieve a high embedding strength through coefficient selection is to limit the selection of low resolution coefficients, which compromises resolution detectability.

Clearly it is not possible to obtain resolution detectability when using a coefficient selection scheme that excludes low-resolution coefficients, thus any attempt to provide both resolution and quality scalability should begin with a coefficient selection scheme that provides good resolution scalability and then find alternative ways to improve the quality detectability.

The most obvious way to achieve quality detectability in a resolution scalable watermark is to increase the embedding strength, α. Of course, this cannot be done for a constant embedding strength watermarking algorithm without violating the watermark imperceptibility constraint. However, replacing the fixed strength α with a variable strength α_i allows the embedding strength to be increased in coefficients that will not substantially contribute to the perceived distortion. If the embedding strength can be sufficiently increased in this manner, for sufficient number of coefficients, then the resulting watermark will be both resolution and quality scalable.

4.3.1 HVS Adaptation

The coefficients that are most able to support an increase in embedding strength without violating the imperceptibility constraint are those to which the human visual system (HVS) is least sensitive. Thus to best achieve quality scalability, the embedding strength should be adapted according to the properties of the HVS.

As was noted in the discussion on objective quality measurement in section 3.2.3.1 (page 69), the HVS is complex and difficult to model. There are many factors that affect our perception of an image and any watermark that it contains. A selection of these include: the brightness of the image and the level of illumination at which it is viewed [136], the spatial frequencies of the watermark pattern [101], the locations of the watermarked regions relative to the focus of attention [146] and whether the regions are familiar or unfamiliar, highly structured or random [182, 203].

Two of these factors, the spatial frequencies of the watermark coefficients and the structural properties of the region, will be used to adapt the embedding strength of a simple, resolution scalable, spread spectrum watermarking algorithm (the `thresh` algorithm of section 4.2) to better match the human visual system.

4.3.1.1 Contrast Sensitivity

Contrast sensitivity is the ability to distinguish a low-contrast pattern from an area of uniform colour. The smallest detectable difference in contrast changes depending on the spatial frequency of the pattern being viewed.

This effect can be seen in a contrast sensitivity chart (figure 4.9), first produced by Campbell and Robson [132]. The chart depicts a sinusoidal pattern of luminance, with logarithmically increasing frequency along the horizontal axis and logarithmically decreasing contrast along the vertical axis. The maximum and minimum luminance values for each oscillation are identical along any horizontal line, thus the apparent height of each

vertical bar is governed by the contrast at which the variation is no longer visible at that frequency and so represents the viewer's own contrast sensitivity at that frequency.

The modification of a wavelet coefficient during the watermark embedding process causes a small change in contrast at the particular location and frequency associated with that wavelet coefficient. If the contrast sensitivity at that frequency is low, then the modification is less likely to cause a visible artifact than if the contrast sensitivity at that frequency is high. Thus the embedding strength α_i may be increased according to the estimated contrast sensitivity at the subband containing the coefficient x_i.

The phenomenon of contrast sensitivity is quite well studied, so it is relatively straightforward to determine the correct adjustment of α_i, through the use of an appropriate *Contrast Sensitivity Function* (CSF), which describes the relative contrast sensitivity of the human visual system for sinusoidal patterns of various frequencies, where the frequency f is measured in cycles per degree of visual angle. While there are many versions of the CSF available, the majority have been developed using luminance based experiments only. The contrast sensitivity functions and method of application described below are obtained from the thesis of Nadenau [131], who developed CSFs specifically for use with colour image content, for a number of different colour spaces. The CSFs used here are those for the $\mathrm{YC_bC_r}$ colourspace,[8] as this is the colourspace used at the time of embedding[9]

$$\mathrm{CSF_Y(f)} = \quad 0.997 f^2 e^{-0.970 f^{0.758}} + 0.221 e^{-0.800 f^{1.999}} \tag{4.46a}$$

$$\mathrm{CSF_{Cb}(f)} = \quad e^{-0.2041 f^{0.900}} \tag{4.46b}$$

$$\mathrm{CSF_{Cr}(f)} = \quad e^{-0.1521 f^{0.893}}. \tag{4.46c}$$

Consider the range of displayable frequencies in one dimension (horizontal or vertical). If all pixels are the same value, then the frequency is zero cycles per degree, which is the minimum possible frequency. The maximum frequency that can be displayed is a single cycle every two pixels. Thus, given the monitor resolution in pixels per inch and the viewing distance in inches, the maximum displayable frequency in cycles per degree of visual angle is

$$f_{\max} = \frac{\tan(1^\circ) \times \text{viewing distance} \times \text{monitor resolution}}{2}. \tag{4.47}$$

The first application of a wavelet transform divides this frequency range into two, so the highpass frequency range is from $f_{\max}$ to $f_{\max}$ and the lowpass frequency range is from 0 to $\frac{f_{\max}}{2}$. Each subsequent decomposition further divides the lowpass frequency range into two.

[8] The function $\mathrm{CSF_Y(f)}$ presented here is a corrected version of the one found in [131] in which the values 0.997 and 0.221 have been switched.

[9] See figure 4.1, page 88 and the description of the component transform in section 2.2.3.3, page 42.

Figure 4.9: A Campbell-Robson contrast sensitivity chart. The apparent height of each bar represents the viewer's contrast sensitivity at that frequency.

An estimate $\mathrm{CSF}(c,s)$ of the average sensitivity to modifications of coefficients in subband s of component c may be obtained by multiplying the CSF values associated with the component c at the midpoints of the horizontal and vertical[10] frequency ranges covered by subband $s = (r,o)$ with orientation o in resolution r of an image with R resolution layers:

$$\mathrm{CSF}(c,s) = \begin{cases} \mathrm{CSF}_c(\frac{1}{2^{R+1-r}} f_{\max})\mathrm{CSF}(\frac{1}{2^{R+1-r}} f_{\max}) & r = 0 \\ \mathrm{CSF}(\frac{1}{2^{R+1-r}} f_{\max})\mathrm{CSF}_c(\frac{3}{2^{R+1-r}} f_{\max}) & r \neq 0, o \neq 2 \\ \mathrm{CSF}_C(\frac{3}{2^{R+1-r}} f_{\max})\mathrm{CSF}(\frac{3}{2^{R+1-r}} f_{\max}) & o = 2. \end{cases} \tag{4.48}$$

The multiplicative inverse of the contrast sensitivity at subband s describes the relative amount of modification which can be applied to a coefficient in subband s to maintain a fixed level of watermark perceptibility. Thus to account for contrast sensitivity, the watermark embedding strength becomes:

$$\alpha_i = \alpha \frac{1}{\mathrm{CSF}(c_i, s_i)} \tag{4.49}$$

Note that although the calculation of $f_{\max}$ requires a specific viewing distance to be specified, the CSFs for the chrominance components are monotonically decreasing as spatial frequency increases. Thus the perceptibility of the watermark in these components will decrease if the image is viewed at greater distances than that used for the calculation of f_{max}. This suggests the use of a minimum expected viewing distance for the watermarked image, maintaining watermark imperceptibility over a range of viewing distances. However, imperceptibility will only be maintained if the CSF for the luminance component is also monotonically decreasing. This is achieved by defining f_{peak} to be the frequency that maximizes $\mathrm{CSF}_{\mathrm{Y}}$ and altering $\mathrm{CSF}_{\mathrm{Y}}$ so that maximum sensitivity is assumed for all frequencies lower than f_{peak}

$$\mathrm{CSF}_{\mathrm{Y}} = 1 \qquad \forall f < f_{\mathrm{peak}} \tag{4.50}$$

4.3.1.2 Texture Based Masking

Tests by Watson et al. [203], in which observers attempt to distinguish a low-contrast Gabor patch target from backgrounds of various types, show that the more complex and random the structure of the background, the better it is able to mask the target. This suggests that the strength of the watermark may be increased for coefficients in textured regions, relative to those in the simpler smooth or edged regions.

[10] The formula presented here assumes the same maximum frequency applies for both the horizontal and vertical dimensions. Thus subband orientations LH and HL are combined into a single entry ($r \neq 0, o \neq 2$). If the monitor uses non-square pixels, then $f_{\max}$ in the lowpass and highpass terms must be replaced with its horizontal and vertical equivalents, yielding different results for each orientation.

In order to achieve this it is necessary to at least identify whether or not a given coefficient lies within a textured region. Further, it is desirable to determine the degree to which the region is textured (textured regions that are of higher contrast or in which the high-contrast areas are more densely packed may be considered to be more strongly textured) in order to provide a greater increase to embedding strength in more strongly textured regions.

What is needed is a texture scoring algorithm, which will provide a numerical value indicating the strength of the texture in the region surrounding any given coefficient. By determining a texture score $\mathfrak{t}_i = \text{Texture}(x_i)$ for each coefficient x_i we can increase the embedding strength accordingly:

$$\alpha_i = \alpha \frac{1 + \mathfrak{t}_i}{\text{CSF}(c_i, s_i)}$$

There are numerous algorithms that have been developed for the identification and classification of texture [65, 150], indeed some have even been used for texture-based masking in watermarking, yet none of these appears to be a suitable candidate for integration with a resolution scalable algorithm from section 4.2. Many of the algorithms do not operate in the DWT domain, and thus would require a substantial amount of additional processing, beyond the texture scoring itself, simply to convert the image to the required domain. Where the DWT domain is used, often all resolution layers are examined when scoring a single coefficient, although using only a single resolution layer would offer similar performance with significantly less processing time. Most importantly, many such algorithms, particularly those used for watermarking, make no distinction between edged and textured regions, resulting in an increased embedding strength for both.

4.3.1.2.1 Single Resolution, Wavelet Domain

Ideally, the texture scoring algorithm will operate in the same domain as embedding, thereby both avoiding the cost of further image transformations and allowing on-the-fly texture scoring during embedding. Thus a wavelet domain texture scoring algorithm is desirable.

A major advantage of wavelet-based algorithms [9, 102, 12, 20, 194, 47] in texture-related applications is the multi-scale nature of the wavelet transform, which allows texture features to be computed at a number of different resolutions and thus improves accuracy over single-scale techniques. Given that this is the case, it seems counter-intuitive to use a wavelet domain texture algorithm which operates on a single resolution only. There are, however, two reasons we may wish to do this. The first of these is purely the implementation advantage of reduced computational overheads. The second is the watermark quality

advantage which results from increasing the embedding strength of coefficients only in those resolutions which are more surely able to mask the modifications.

4.3.1.2.1.1 Implementation

For maximum usefulness in a scalable watermarking context, the texture estimation algorithm must be able to function as part of a watermarking scheme within a scalable compression system. Given that the texture score produced by the algorithm is used to adjust the watermark strength on the embedding side, if this adjustment is to be reversed, the texture score must be obtainable on the detection side.[11] As a result, the texture algorithm should be suitable for use by a device with potentially limited capabilities and hence the less processing required, the better. Furthermore, even where all devices have ample processing power, reducing the processing required for texture scoring will reduce the time required in both the embedding and detection stages.

In a typical six-layer decomposition, up to five resolution layers might be involved in calculation of the texture score for a single coefficient, reducing this to one resolution layer can substantially reduce the computational requirements. However, this will be inconsequential if the use of only a single resolution significantly reduces our ability to determine the degree of texture in each region, resulting in inappropriate embedding strengths.

4.3.1.2.1.2 Quality

While the many wavelet-based texture algorithms make use of the multi-scale nature of the wavelet transform, these algorithms are commonly designed for tasks such as texture classification where using a number of different resolutions allows textures which are similar at some resolutions to be distinguished using other resolutions.

In a watermarking context, however, we are only interested in texture because of its ability to mask the visual distortion introduced by coefficient modification. All that is required is a single score which can be used to determine if a coefficient can tolerate a higher embedding strength or if it cannot. By calculating this score from a block of coefficients in the same resolution layer as the selected coefficient we determine the amount of texture which occurs at frequencies similar to the noise which will be added by watermarking that coefficient. Including other resolutions in this calculation risks artificially raising the score as a result of high amounts of texture detected in resolutions which may have little ability to mask the noise being added.

This problem can be seen in figure 4.10. In figure 4.10a, a coefficient with a high texture score has had it's embedding strength increased by a factor of 20. In figure 4.10b

[11] Although this only results in additional processing requirements for this non-blind algorithm, it can be a troublesome issue in blind watermarking, as the coefficient bits required for the texture score may not be available.

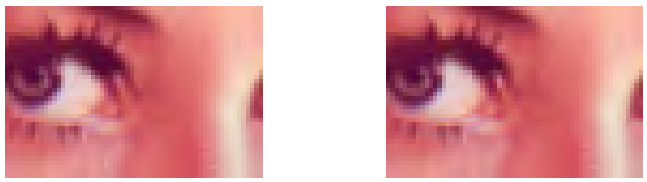
a coefficient with a low texture score, chosen from the same spatial location[12] but from a different resolution layer, has been modified in the same manner. The modification is clearly visible in figure 4.10b, where the texture score in the same resolution as the modified coefficient was low.

(a) High Texture Score, $r_i = 5$

(b) Low Texture Score, $r_i = 4$

Figure 4.10: High resolution texture masks the modification (in Lena's hair) of the high resolution ($r_i = 5$) coefficient with a high texture score, but not the lower resolution ($r_i = 4$) coefficient with a low texture score at the same spatial location.

Figure 4.11 is similar to figure 4.10 but with the modified coefficients from a different region of the image. So, whereas for the previous pair the high texture score occurred in resolution layer 5 and the low texture score in resolution layer 4, for this pair the high texture score occurs in resolution layer 4 and the low texture score in resolution layer 5. That, in this pair of images also, the change to the low texture score coefficient is visible but the change to the high texture score coefficient is not, indicates that it is not the resolution layer at which the coefficient resides which is important, but the texture score associated with it.



(a) High Texture Score, $r_i = 4$

(b) Low Texture Score, $r_i = 5$

Figure 4.11: Low resolution texture masks the modification (on Lena's upper eyelid) of the low resolution ($r_i = 4$) coefficient with a high texture score, but not the higher resolution ($r_i = 5$) coefficient with a low texture score at the same spatial location.

[12]The modified coefficients in figures 4.10a and 4.10b can be roughly located at 365 257, and those in figures 4.11a and 4.11b at 283 267, in the 512x512 Lena image.

The likely reason that high texture in the same resolution as the watermarked coefficient is more effective at hiding the distortion caused by watermarking can be found in the results of Legge and Foley [101], who performed sensitivity experiments in which both the backgrounds and target patches were of various frequencies. The results showed that the ability of the background to mask the target was highest when the frequency of the background and target were the same, but dropped off rapidly as the absolute difference between their frequencies increased.[13] This suggests that the majority of the masking effect will originate from coefficients of similar frequency to (i.e. in the same resolution layer as) the coefficient being watermarked. Thus, performing texture calculations using only the resolution layer containing the selected coefficient should produce an acceptable conservative estimate of the texture based masking effect at that coefficient at substantially reduced cost.

4.3.1.2.2 Separation from Edged and Smooth Regions

Although textured regions are fairly easy to distinguish from smooth regions due to their high contrast, both textured and edge regions are composed of high contrast areas, so it may be difficult to separate the two. As a result, texture detection algorithms used for watermarking [9, 12, 196] often detect edges as well. That is, rather than distinguishing textured regions from edged and smooth regions, they distinguish smooth regions from edged and textured regions. This problem is often not addressed, with Voloshynovskiy et al. [196] citing decreased sensitivity to 'edged and textured' regions as a whole.

While there does exist an edge based masking effect, it is highly limited [40]. First, it requires that the site of modification be very near the edge (within 2 or 3 pixels). Second, and more importantly, it requires that the modification and the edge be of the same orientation. The extremely limited orientation sensitivity of the critically sampled wavelet transform employed in JPEG2000 compression means that this second criterion is unlikely to be satisfied. Furthermore, there is the potential for increased sensitivity to modification near edges due to their importance in high-level vision tasks [63, 57, 186].

This suggests that the texture scoring algorithm should separate textured regions not only from smooth regions but also from edged regions. Doing so will allow the watermark to be embedded more strongly in textured regions without risking visibility at edges.

[13] Although the experiments of Legge and Foley were with simple sinusoidal backgrounds, the improved masking ability of similar frequencies also holds for textured backgrounds, as can be observed by comparing the masking effects of high-frequency and bandpass noise backgrounds in the paper of Watson et al. [203].

4.3.1.2.3 The Texture Scoring Algorithm

As there does not appear to be a texture scoring algorithm that satisfies the above criteria, a single-resolution wavelet domain algorithm that distinguishes textured regions from both smooth and edged regions is developed in appendix C. This algorithm provides a texture score $\mathfrak{t}_i$ indicating the degree of texture present in the neighbourhood surrounding a given coefficient x_i at the resolution r_i to which the coefficient belongs.

Preconditions:

The resolution r_i satisfies $r_i > \lfloor \frac{R}{3} \rfloor$, where R is the number of resolution layers in the decomposed image.

Setup:

Let $B_{i,0}$, $B_{i,1}$ and $B_{i,2}$ denote blocks of 17×17 coefficients in resolution r_i, centred at the spatial location of coefficient x_i, but with subband orientations of 0, 1 and 2 respectively.

Let $e(s_{i,0})$, $e(s_{i,1})$ and $e(s_{i,2})$ be the average coefficient magnitudes of the subbands containing blocks $B_{i,0}$, $B_{i,1}$ and $B_{i,2}$ respectively, and let $|B_{i,o}|$ be the number of coefficients[14] in the block $B_{i,o}$ with orientation o.

Texture
Input: $B_{i,0}, B_{i,1}, B_{i,2}, e(s_{i,0}), e(s_{i,1}), e(s_{i,2})$ **Output:** $\mathfrak{t}_i$
◦ For i = 0 to 2 $e(B_{i,o}) = \sum_{b \in B_{i,o}} \frac{\lvert b \rvert}{\lvert B_{i,o} \rvert}$ $c(B_{i,o}) = \sum_{b \in B_{i,o} \wedge \lvert b \rvert > 1.5e(s_{i,o})} \frac{1}{\lvert B_{i,o} \rvert}$ ◦ $E_i = \sqrt{\max(\ e(B_{i,0})e(B_{i,1}),\ e(B_{i,0})e(B_{i,2}),\ e(B_{i,1})e(B_{i,2})\ )}$ ◦ $C_i = \sqrt{\max(\ c(B_{i,0})c(B_{i,1}),\ c(B_{i,0})c(B_{i,2}),\ c(B_{i,1})c(B_{i,2})\ )}$ ◦ $\mathfrak{t}_i = 0.4E_i + 7.7\dot{2}C_i$

4.3.2 Comparative Evaluation of Scalability with HVS Adaptation

When HVS adaptation is applied to a resolution scalable watermarking algorithm, the quality scalability of that algorithm should be improved without adversely affecting its resolution scalability. The primary aim of this set of experiments, therefore, is to determine whether the proposed HVS adaptive algorithm demonstrates a substantial increase in quality scalability over the non adaptive algorithm, on which it is based, without showing a substantial decrease in resolution scalability.

A secondary aim is to examine the scalability of the HVS adaptive algorithm relative to the existing algorithms of Cox et al.[30] and Xia et al.[210]. The first of these has

[14]The number of coefficients $|B_{i,o}|$ in a given block may be less than the nominal size of 17×17 if the block is too close to the subband boundary.

been chosen because it is well known by the watermarking community. The second has been chosen because it, like the proposed algorithm, has been designed to allow detection from reduced resolution and compressed content and is HVS adaptive, with increased embedding strength at both edges and textured regions.

4.3.2.1 The Algorithms

The four algorithms compared in this section all use the basic spread spectrum watermarking algorithm, presented at the beginning of this chapter, differing only in coefficient selection and embedding strength.

- Cox

  The Cox algorithm is a wavelet version of the well-known algorithm presented in [30]. In the experiments in [30], the coefficients X, selected from a discrete cosine transformed image, were the highest magnitude coefficients from the greyscale component, excluding the DC coefficients. For these experiments, this coefficient selection scheme is modified to operate in the wavelet domain, using the `lumnl` scheme of section 4.2.1 (page 102), so the selected coefficients X are the 1000 highest magnitude coefficients of the luminance component only, excluding the lowest resolution layer. Embedding is performed using a fixed embedding strength α.

- Xia

  The Xia algorithm is a version of the algorithm presented in [210], which showed robustness to EZW compression. In their paper, the coefficients X, selected from a four-level wavelet transformed image, were the highest magnitude coefficients excluding the LL subband. Our implementation uses the 1000 highest magnitude coefficients excluding the LL subband but with a six-level wavelet transformed image, the `nolow` scheme of section 4.2.1 (page 102). The embedding strength is adapted according to the human visual system using the formula $\alpha_i = \alpha x_i$, which increases the strength of the watermark at "edges and textures".

  Also, in the original algorithm of Xia et al., watermark detection used the correlation statistic *cor*, rather than the similarity statistic, and watermark presence was determined by computing the correlation between the candidate watermark W and all shifted versions of the extracted watermark W^* in search of a distinct peak in the correlation. To simplify the comparison process, this implementation uses the detection procedure described at the beginning of this chapter, as do the other implementations. This should not have any significant impact on the relative performance of the Xia algorithm, as the similarity statistic is correlation based and neither resolution nor quality scaling attacks should affect the watermark synchronization.

- `nohvs`

  The `nohvs` algorithm is simply a non-adaptive resolution scalable algorithm chosen from those used in section 4.2. The coefficients X are selected using the `thresh` scheme, to be those with magnitude greater than threshold of two fifths of the maximum magnitude coefficient in their associated resolution layer. As with the previous experiment, embedding is stopped once 1000 coefficients have been selected. If less than 1000 coefficients satisfy the selection criterion, the remaining coefficients are selected from the highest magnitude coefficients in the highest resolution. Embedding is performed using a fixed embedding strength α.

- `hvs`

  The `hvs` algorithm is the same as the `nohvs` algorithm except that it employs the HVS-adapted embedding strength $\alpha_i = \alpha \frac{1+\mathfrak{t}_i}{\mathrm{CSF}(c_i,s_i)}$, described in section 4.3.1.

4.3.2.2 Experimental Framework

To reliably verify the claim that the addition of HVS adaptation to the resolution scalable `nohvs` algorithm will result in a substantial increase in quality scalability without a substantial decrease in resolution scalability, the comparison procedure outlined in section 3.2.4 (page 74) is used.

4.3.2.2.1 Design

4.3.2.2.1.1 Attacks and Performance Measures

The attacks for this comparison consist of resolution and quality scaling using the JasPer implementation of JPEG2000. Although the image is decomposed into 6 resolution layers during embedding, only the 5 highest resolution subimages are produced, to avoid disadvantaging the Cox and Xia algorithms prohibit embedding in the LL subband. The quality scaling attack uses compression ratios of 0.01, 0.02, 0.04, 0.06 and 0.9999, resulting in 5 quality scaled subimages.

The performance measures used are the scalability measures, detectability $\mathcal{D}$ and graceful improvement $\mathcal{G}$, that were defined in section 3.1.2 (page 54). These are calculated, for both resolution and quality scalability, using the mean similarity values across 10 keys. The calculation details are as follows:

The detectability value is the mean similarity value at the lowest resolution or quality subimage. However, note that in the resolution scaled case, the lowest subimage is $I_1^{\mathcal{R}}$,

because $I_0^{\mathcal{R}}$ has been excluded to avoid disadvantaging the Cox and Xia algorithms:

$$\mathcal{D}^{\mathcal{R}} = \frac{\sum_{sk=1}^{10} \gamma(I^{\mathcal{R}_1}, I, sk)}{10} \tag{4.51}$$

$$\mathcal{D}^{\mathcal{Q}} = \frac{\sum_{sk=1}^{10} \gamma(I^{\mathcal{Q}_0}, I, sk)}{10}. \tag{4.52}$$

The ideal number of watermark elements for layer l is the portion of the 1000 elements that should be embedded in a given layer according to the increase in quality, measured using PSNR, provided by that layer, relative to the total improvement in quality of the full image over a mid-grey image I^e.

$$\iota^{\mathcal{R}_1} = 1000 \frac{P^{\mathcal{R}_1} - P^e}{P^{\mathcal{R}_5} - P^e} \tag{4.53}$$

$$\iota^{\mathcal{Q}_0} = 1000 \frac{P^{\mathcal{Q}_0} - P^e}{P^{\mathcal{Q}_4} - P^e} \tag{4.54}$$

for the lowest layers, and

$$\iota^{\mathcal{R}_l} = 1000 \frac{P^{\mathcal{R}_l} - P^{\mathcal{R}_{l-1}}}{P^{\mathcal{R}_5} - P^e} \tag{4.55}$$

$$\iota^{\mathcal{Q}_l} = 1000 \frac{P^{\mathcal{Q}_l} - P^{\mathcal{Q}_{l-1}}}{P^{\mathcal{Q}_4} - P^e} \tag{4.56}$$

for all higher layers l.

The mean similarity values of sequential, scaled subimages are used to calculate an equivalent number of whole watermark elements.

$$\epsilon^{\mathcal{R}_1} = \left(\frac{\sum_{sk=1}^{10} \gamma(I^{\mathcal{R}_1}, I, sk)}{10} \right)^2 \tag{4.57}$$

$$\epsilon^{\mathcal{Q}_0} = \left(\frac{\sum_{sk=1}^{10} \gamma(I^{\mathcal{Q}_0}, I, sk)}{10} \right)^2 \tag{4.58}$$

for the lowest layers, and

$$\epsilon^{\mathcal{R}_l} = \left(\frac{\sum_{sk=1}^{10} \gamma(I^{\mathcal{R}_l}, I, sk)}{10} \right)^2 - \left(\frac{\sum_{sk=1}^{10} \gamma(I^{\mathcal{R}_{l-1}}, I, sk)}{10} \right)^2 \tag{4.59}$$

$$\epsilon^{\mathcal{Q}_l} = \left(\frac{\sum_{sk=1}^{10} \gamma(I^{\mathcal{Q}_l}, I, sk)}{10} \right)^2 - \left(\frac{\sum_{sk=1}^{10} \gamma(I^{\mathcal{Q}_{l-1}}, I, sk)}{10} \right)^2 \tag{4.60}$$

for all higher layers l.

With the resulting graceful improvement values being

$$\mathcal{G}^{\mathcal{R}} = 1 - \frac{\sum_l \frac{(\epsilon^{\mathcal{R}_l} - \iota^{\mathcal{R}_l})^2}{\iota^{\mathcal{R}_l}}}{N(\frac{N}{\iota^{\mathcal{R}_\mathrm{m}}} - 1)} \tag{4.61}$$

and

$$\mathcal{G}^{\mathcal{Q}} = 1 - \frac{\sum_l \frac{(\epsilon^{\mathcal{Q}_l} - \iota^{\mathcal{Q}_l})^2}{\iota^{\mathcal{Q}_l}}}{N(\frac{N}{\iota^{\mathcal{Q}_\mathrm{m}}} - 1)} \tag{4.62}$$

where $\mathcal{R}_{\mathfrak{m}}$ or $\mathcal{Q}_{\mathfrak{m}}$ is the resolution layer or quality layer, respectively, with the smallest, non-zero ideal value.

4.3.2.2.1.2 The Hypothesis Test and Related Assumptions

For each of the four performance measures defined above, the `hvs` algorithm will be compared to each of the other algorithms using a paired t-test (section 3.2.4.2.3.1, page 79). The paired t-test is chosen because it is a well-known and powerful test which exploits our ability to watermark identical copies of the same image to produce paired performance measurements.

To use a paired t-test three assumptions must be satisfied (section 3.2.4.3.1, page 82): pairing, independence and normality. The pairing and independence assumptions are verified as part of the design phase.

A1 Pairing assumption:

Measuring the performance of both algorithms using the same set of test images $\mathcal{I}$ allows the formation of the required matched pairs $\mathfrak{a}_i$ and $\mathfrak{b}_i$, being the corresponding performance measurements of watermarking algorithms A and B from the same subject (image) $I_i \in \mathcal{I}$.

A2 Independence assumption:

Both the image and the secret key influence the detection statistic, thus it is desirable to use multiple keys in addition to multiple images. Using a separate pair $\mathfrak{a}_{i,sk}, \mathfrak{b}_{i,sk}$ for each combination of image I_i and secret key sk would result in pairs $\{\mathfrak{a}_{i,sk}, \mathfrak{b}_{i,sk} | 1 \leq sk \leq 10\}$ all sharing a dependency on I_i, violating the independence assumption.

However, because each performance measure is derived from the average similarity value for a single image across all keys $sk \in \{1, 2, \ldots, 10\}$, in any given test, the measurements $\mathfrak{a}_i, \mathfrak{b}_i$ will be derived from the image I_i, while any other pair $\mathfrak{a}_j, \mathfrak{b}_j$ will be derived from a different image I_j, so the independence assumption is satisfied.

A3 Normality assumption:

This is a distributional assumption, so its verification requires the data generation to be complete. Thus verification is left to the analysis phase.

4.3.2.2.1.3 The Required Number of Images

The number of images n to be used in the experiment is determined (section 3.2.4.2.5, page 81) according to the type-I α and type-II β error rates, the standard deviation of the paired difference values σ_d and the size of the minimum difference δ in mean performance that the test should detect.

α

Following convention, the hypothesis tests in this experiment use a type-I error rate of $\alpha = 0.01$ (99% confidence). If the test detects a difference in the average performance of two algorithms, then there is a 99% probability that there is a real difference in the average performance.

Note that no correction to the type-1 error rate has been made to account for the use of multiple tests, as the primary focus of this experiment in testing specific claims regarding the detectability of two algorithms (`hvs` and `nohvs`). However, the p-values are reported in the results and could be compared against a different α if a more conservative test were desired.

β

Again, following convention, all tests use a type-II error rate of $\beta = 0.1$ (90% power). If the test doesn't detect a difference in the average performances of two algorithms, then there is a 10% probability that there exists a real difference in the average performances but it could not be detected.

σ_d

Unlike α and β, different values of σ_d will apply for each test, depending on which algorithms and measures are involved. Estimates of σ_d are produced by obtaining the paired differences between the `hvs` algorithm and each of the other algorithms, for each performance measure, on a sample of 40 images, and calculating the sample standard deviation. Table 4.1 shows the estimates of σ_d for the `hvs` algorithm paired with each of the other algorithms for each resolution/quality detectability/graceful improvement combination.

Table 4.1: Estimates of the standard deviations of paired difference values between the `hvs` algorithm and each other algorithm X, calculated using a 40 image sample.

X	$\sigma_d^{\mathcal{RD}}$	$\sigma_d^{\mathcal{QD}}$	$\sigma_d^{\mathcal{RG}}$	$\sigma_d^{\mathcal{QG}}$
`nohvs`	0.00288	5.48	0.0359	0.0346
Cox	3.14	4.45	0.0410	0.0326
Xia	3.05	5.52	0.0519	0.0365

δ

The choice of δ is more subjective, as it represents the minimum substantial difference between the mean performance of the algorithms. A different value of δ is chosen for each of the four performance measures.

Detectability

For detectability a substantial difference in mean performance is chosen to be one which reduces the false negative error rate of the watermarking system under the scaling $\mathcal{F}$ by at least 25%. The corresponding value of $\delta^{\mathcal{FD}}$ may be estimated by modelling the detection statistic as normally distributed, with mean $\mu^{\mathcal{FD}}$ and a standard deviation estimated by the average standard deviation of all four algorithms

$$\sigma^{\mathcal{FD}} = \sqrt{\frac{\sum_{\mathrm{X}} (\sigma_{\mathrm{X}}^{\mathcal{FD}})^2}{4}}, \tag{4.63}$$

$$\tag{4.64}$$

and setting $\delta^{\mathcal{FD}}$ to be the value such that the relative reduction in false negative error for a detection statistic with a mean $\mu^{\mathcal{FD}} \geq T$ compared to that with a mean of $\mu^{\mathcal{FD}} + \delta^{\mathcal{FD}}$, where each has a standard deviation of $\sigma^{\mathcal{FD}}$ is not below 25%.

A sample of 40 images produces detectability values with sample standard deviations as shown in table 4.2, giving averages of

$$\sigma^{\mathcal{RD}} = 2.71 \tag{4.65}$$

$$\sigma^{\mathcal{QD}} = 4.28. \tag{4.66}$$

Table 4.2: Estimated standard deviation of detectability values using a 40 image sample.

X	$\sigma_{\mathrm{X}}^{\mathcal{RD}}$	$\sigma_{\mathrm{X}}^{\mathcal{QD}}$
`hvs`	2.80	5.02
`no hvs`	2.80	4.82
Cox	2.61	3.38
Xia	2.60	3.89

The values of $\delta^{\mathcal{FD}}$, are the minimum difference in means required to produce a reduction in false negative error of 25% at the threshold level of 6, given the average standard deviations $\sigma^{\mathcal{RD}} = 2.71$ and $\sigma^{\mathcal{QD}} = 4.28$:

$$\delta^{\mathcal{RD}} = 0.863 \tag{4.67}$$

$$\delta^{\mathcal{QD}} = 1.37. \tag{4.68}$$

Graceful Improvement

Because graceful improvement measures goodness of fit; it has no direct link to the error probability, so it is not possible to use the same method to determine $\delta^{\mathcal{FG}}$ as was

used for $\delta^{\mathcal{FD}}$. Instead, $\delta^{\mathcal{FG}}$ is chosen to be a fraction of the average standard deviation for all schemes.

$$\delta^{\mathcal{FG}} = 0.72 \times \sigma^{\mathcal{FG}}, \tag{4.69}$$

A sample of 40 images produces graceful improvement values with sample standard deviations as shown in table 4.3, giving averages of

$$\sigma^{\mathcal{RG}} = 0.0356 \tag{4.70}$$

$$\sigma^{\mathcal{QG}} = 0.0342 \tag{4.71}$$

So the values of $\delta^{\mathcal{FG}}$ are

$$\delta^{\mathcal{RG}} = 0.0256 \tag{4.72}$$

$$\delta^{\mathcal{QG}} = 0.0246. \tag{4.73}$$

Table 4.3: Estimated standard deviation of graceful improvement values using a 40 image sample.

X	$\sigma_{\mathrm{X}}^{\mathcal{RG}}$	$\sigma_{\mathrm{X}}^{\mathcal{QG}}$
`hvs`	0.0416	0.0306
`no hvs`	0.0393	0.0370
Cox	0.0197	0.0278
Xia	0.0417	0.0419

The values of δ for all four measures are shown in table 4.4.

Table 4.4: Values of δ, the minimum difference in mean performance that is deemed 'substantial', for each performance measure.

$\delta^{\mathcal{RD}}$	$\delta^{\mathcal{QD}}$	$\delta^{\mathcal{RG}}$	$\delta^{\mathcal{QG}}$
0.863	1.37	0.0256	0.0246

Number of images

According to the power formula for a paired t-test, to achieve the desired error rates α and β given the desired minimum difference δ and standard deviation of differences σ_d, the number of images used in the experiment must be:

$$n = \frac{\sigma_d^2}{\delta^2}(t_{\alpha(2),\nu} + t_{\beta(1),\nu}) \tag{4.74}$$

$$\tag{4.75}$$

This formula is applied, with the values of α,β,σ_d and δ determined above, and degrees of freedom $\nu = n - 1$. The minimum values of n that satisfy the formula are presented in table 4.5:

Table 4.5: Required number of images for each algorithm and performance measure; calculated using $\alpha = 0.01$ $\beta = 0.1$, σ_d from table 4.1, and δ from table 4.4.

X	$n^{\mathcal{RD}}$	$n^{\mathcal{QD}}$	$n^{\mathcal{RG}}$	$n^{\mathcal{QG}}$
`no hvs`	1	64	33	33
Cox	53	43	42	30
Xia	50	65	65	37

To ensure each comparison has enough images, n is set to be the maximum of these values, so 65 images are used for the experiment.

4.3.2.2.2 Data Generation

4.3.2.2.2.1 Embedding

Each watermarking algorithm is applied to a copy of each of 65 original images. Watermarking occurs directly preceding the quantization stage of JPEG2000 compression with 6 resolution layers, precincts of size 128 × 128, and quality layers with rates 0.01, 0.02, 0.04, 0.06 and 0.9999.

So that each watermark has the same level of perceptibility, the embedding strength is adjusted according to the perceptual difference between the original and watermarked image. The perceptual difference is computed using the S-CIELAB metric, which was developed by Zhang and Wandell [213] as spatial extension to the CIELAB colour difference metric to allow its use in digital images, updated to use CIEDE 2000 ΔE, which better approximates the human visual system [167]. For each watermark, the global embedding strength α is adjusted so that the 99th percentile[15] of the S-CIELAB CIEDE 2000 error[16] between the original image and the full watermarked image is $4\Delta E$.

The process is repeated for each of 10 secret keys. This number is reduced relative to the 100 keys used in the previous experiment (section 4.2.2), as there appears to be relatively little variability in the detection statistic for different embedding keys.

[15]The 99th percentile (rather than the average) ΔE value is used because it is thought[129] to provide a closer match to subjective perceptibility.

[16]The settings used for the calculation of the S-CIELAB CIEDE 2000 error were those of a Dell 1702FP (Analog) monitor, 96dpi, viewed at 46cm.

4.3.2.2.2.2 Attacking

Each watermarked image I' undergoes JPEG2000 scaling, producing 5 resolution scaled images $I^{\mathcal{R}_1} \ldots I^{\mathcal{R}_5}$ and 5 quality scaled images $I^{\mathcal{Q}_0} \ldots I^{\mathcal{Q}_4}$ with compression ratios 0.01, 0.02, 0.04, 0.06 and 0.9999.

4.3.2.2.2.3 Detection

Watermark detection, using the correct detection key, is applied to each watermarked subimage and the resulting similarity value is recorded. These values are used to calculate, for each algorithm, the detectability and graceful improvement values under resolution and quality scaling for each original image, according to the procedure described earlier.

4.3.2.2.3 Analysis

For each type of scalability, a paired t-test is used to compare the average detectability and graceful improvement, across all 65 images, of the proposed `hvs` algorithm with each of the other algorithms.

4.3.2.2.4 Assumption Verification

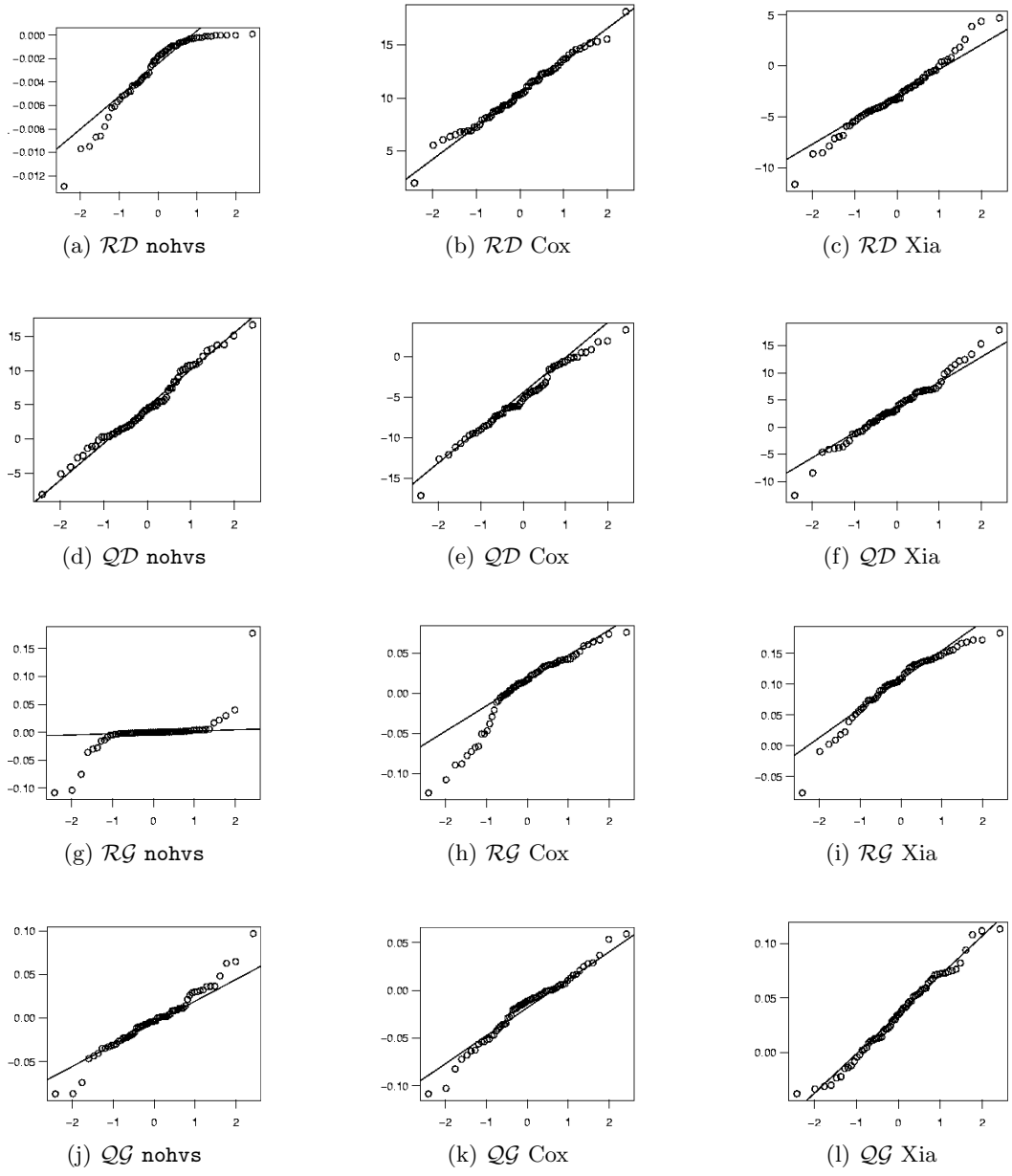
To obtain meaningful results from a paired t-test, it is important that the assumptions required for that test be satisfied. The first two of these assumptions have already been verified during the design phase, so all that remains is to check the third:

A3 Normality assumption: $d_i \sim N(\mu_d, \sigma_d)$

The normality of the paired differences in performance must be checked for each performance measure and algorithm pair that is to be tested. This is done using normal quantile plot[17] (figure 4.12). A plot that deviates more than slightly from a straight 45° line, indicates non-normality.

For the resolution detectability measure, the paired differences between `hvs` and `nohvs` do not appear to have a normal distribution (figure 4.12a), nor do any of the resolution graceful improvement paired differences (figure 4.12g to figure 4.12i). In these cases a paired t-test will be unreliable, because the normality assumption is not satisfied, so a nonparametric alternative to the paired t-test, the sign test, will be used instead. In all other cases, paired differences are reasonably normal and the planned paired t-test will be used.

[17] A normal quantile plot matches the sorted sample values against the sorted values of a theoretical normal distribution such that if the values are normally distributed the points will form a straight line through the origin at a 45° angle. Deviations from normality may be identified by the comparing the plotted points to the line.



(a) $\mathcal{RD}$ `nohvs` (b) $\mathcal{RD}$ Cox (c) $\mathcal{RD}$ Xia

(d) $\mathcal{QD}$ `nohvs` (e) $\mathcal{QD}$ Cox (f) $\mathcal{QD}$ Xia

(g) $\mathcal{RG}$ `nohvs` (h) $\mathcal{RG}$ Cox (i) $\mathcal{RG}$ Xia

(j) $\mathcal{QG}$ `nohvs` (k) $\mathcal{QG}$ Cox (l) $\mathcal{QG}$ Xia

Figure 4.12: Normal quantile plots for the paired differences between the `hvs` algorithm and each other algorithm, for each performance measure.

4.3.2.3 Results

4.3.2.3.1 Detectability

Table 4.6 shows the paired t-test results for the detectability measure. Table 4.7 shows the sign test result for the difference in resolution detectability between the `hvs` and `nohvs` algorithms. A positive difference indicates that the `hvs` algorithm shows improved detectability over the given algorithm, with larger values indicating a greater level of improvement. Figure 4.13 shows the mean detectability values for each algorithm over all 65 images used.

Table 4.6: Paired t-test results – Improvement in detectability gained using the `hvs` algorithm relative to algorithm X.

X	$\mathcal{F}$	p-value	mean difference
`nohvs`	resolution	NA	-0.00288
Cox	resolution	2.2×10^{-16}	10.4853
Xia	resolution	3.6×10^{-10}	-2.8327
`nohvs`	quality	4.4×10^{-10}	4.7591
Cox	quality	5.9×10^{-15}	-5.0401
Xia	quality	1.5×10^{-6}	3.6001

Table 4.7: Sign test results – Improvement in detectability gained using the `hvs` algorithm relative to the `nohvs` algorithm.

X	$\mathcal{F}$	p-value	median difference
`nohvs`	resolution	4.87×10^{-13}	-0.0018

Depending on the desired false positive and false negative rates for a given application, these results may or may not be acceptable. The false negative results at a threshold of 6, including the false negative rate over 65 original images each watermarked with 10 keys and the number of original images for which all watermarks were detected at the threshold value of 6, are given in table 4.8.

If only a single type of scalability is required, an algorithm which concentrates on that type of scalability will be preferred.[18] However, if both types of scalability are required,

[18] Using a threshold value of 6, the `hvs` algorithm detects all watermarks in 64 of the 65 tested images after resolution scaling to $\frac{1}{256}$th of the original image area (one watermark could not be detected in image 73708089), in contrast to 65 of 65 for the Xia algorithm. At the same threshold, the `hvs` algorithm detects all watermarks in 64 of the 65 tested images after quality scaling to $\frac{1}{100}$th of the original file size (eight watermarks could not be detected in image img36), in contrast to 65 of 65 for the Cox algorithm.

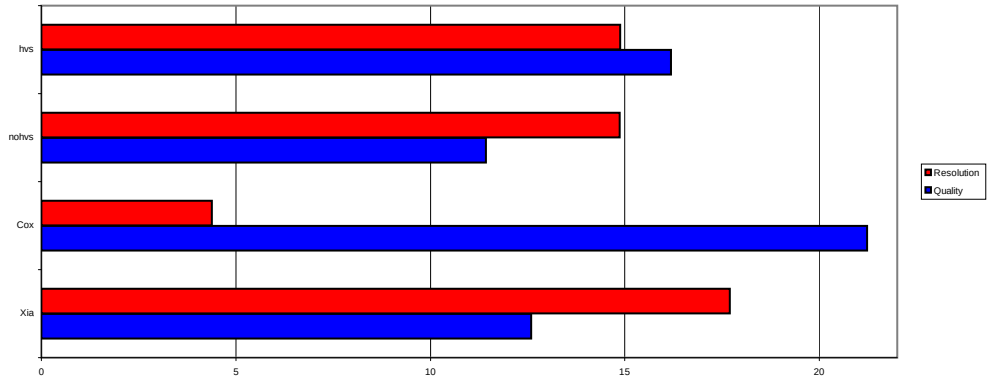


Figure 4.13: Average detectability values for each algorithm over 65 images.

the `hvs` algorithm is clearly superior. After resolution scaling to $\frac{1}{256}$th of the original image area, the Cox algorithm detects all watermarks in only 9 of the 65 images. After quality scaling to $\frac{1}{100}$th of the original file size, the `nohvs` algorithm detects all watermarks in only 45 images and the Xia algorithm detects all watermarks in only 49 images of the 65 possible.

Table 4.8: False negative rate calculated from 65 original images, each watermarked with 10 keys, at a detection threshold of 6.

X	$\mathcal{F}$	false negative rate	# originals with %100 detection
`hvs`	resolution	0.00154	64
`hvs`	quality	0.0123	64
`nohvs`	resolution	0.00154	64
`nohvs`	quality	0.148	45
Cox	resolution	0.715	9
Cox	quality	0	65
Xia	resolution	0	65
Xia	quality	0.0446	49

By assuming that the similarity statistic follows a normal distribution it is possible to estimate the false negative rate for each algorithm at a lowest resolution or quality subimage, $I^{\mathcal{R}_1}$ or $I^{\mathcal{Q}_0}$. This is done by using the mean and standard deviation of the detectability values for each algorithm to fit a normal distribution and then calculating the proportion of this distribution which lies below the detection threshold. Table 4.9 shows the estimated false negative rates for each algorithm at the lowest resolution and the lowest quality layer for a threshold value of 6. The Cox algorithm has by far the fewest false negatives for the quality decomposition and the Xia algorithm has by far the fewest

false negatives for the resolution decomposition. The `hvs` algorithm is the only one for which false negative error rates for both resolution and quality scaled content are below 0.05.

Table 4.9: Estimated rate of false negative errors for each algorithm.

X	$\mathcal{F}$	estimated false negative rate
`hvs`	resolution	0.00033
`hvs`	quality	0.020007
`nohvs`	resolution	0.00033
`nohvs`	quality	0.13175
Cox	resolution	0.74190
Cox	quality	1.3426×10^{-6}
Xia	resolution	2.951×10^{-6}
Xia	quality	0.05811

That the tradeoff between quality and resolution detectability has not been entirely eliminated is apparent from the relative performances of the Cox, Xia and `hvs` algorithms. There is a substantial increase in resolution detectability relative to the Cox algorithm, but a substantial decrease in quality detectability relative to the same algorithm. Similarly, there is a substantial increase in quality detectability relative to the Xia algorithm, but a substantial decrease in resolution detectability.

However, the lessening of this tradeoff due to the introduction of human visual system based techniques is clear. The `hvs` algorithm has substantially improved quality detectability over the `nohvs` algorithm, with a mean difference as high as 4.76, well above the value $\delta^{QD} = 1.37$ that was deemed substantial. Although the reduction in resolution detectability between the `hvs` and `nohvs` algorithms is significant, it is not substantial, the mean difference is just -0.0029, well below the value $\delta^{RD} = 0.863$ that was deemed to be a substantial difference.

4.3.2.3.2 Graceful Improvement

Both the sign test results for resolution graceful improvement (table 4.10) and the paired t-test results for quality graceful improvement (table 4.11) show the Xia algorithm to have substantially worse performance than the `hvs` algorithm (indeed it has lower graceful improvement than all other algorithms, as can be seen in figure 4.14). This may be the result of an overemphasis of the watermark in large and low-resolution coefficients and

an under-emphasis in small and high-resolution coefficients, caused by excessive weighting of the embedding strength by the coefficient magnitude.

Table 4.10: Sign test results - Graceful improvement gained using the `hvs` algorithm relative to algorithm X.

X	$\mathcal{F}$	p-value	median difference
`nohvs`	resolution	1	3.94×10^{-5}
Cox	resolution	0.002626	0.0169
Xia	resolution	1.163×10^{-16}	0.1088

Table 4.11: Paired t-test results – Graceful improvement gained using the `hvs` algorithm relative to algorithm X.

X	$\mathcal{F}$	p-value	mean difference
`nohvs`	resolution	NA	-0.001769
Cox	resolution	NA	0.007108
Xia	resolution	NA	0.1041
`nohvs`	quality	0.5118	-0.002666
Cox	quality	9.351×10^{-5}	-0.01725
Xia	quality	2.693×10^{-10}	0.03407

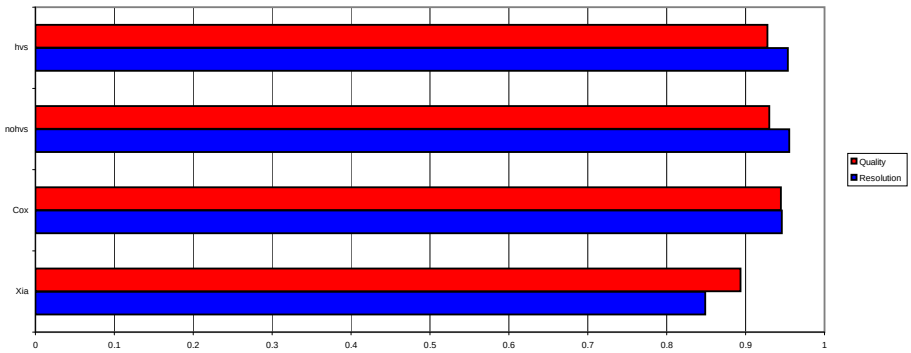


Figure 4.14: Average graceful improvement values for each algorithm over 65 images

Average graceful improvement values for each algorithm over 65 images.

The differences in graceful improvement between the `hvs` and Cox algorithms are qualitatively the same as those found in the detectability comparison. The `hvs` algorithm outperforms the Cox algorithm for a resolution decomposition, while the Cox algorithm

outperforms the `hvs` algorithm for a quality decomposition. However, these differences do not appear to be substantial.

As expected, the use of HVS adaptation did not affect the graceful improvement performance using a resolution decomposition. However, nor did it affect the graceful improvement performance using a quality decomposition. This means that the improvements in quality detectability gained through the use of HVS adaptation have not translated into improvements in quality graceful improvement. This is does not appear to be too great a problem since, as is not the case for detectability, the `nohvs` algorithm already shows good graceful improvement for both resolution and quality decompositions.

4.4 Conclusion

The primary challenge in designing scalable spread spectrum watermarking techniques lies in satisfying the detectability property, as graceful improvement results for most algorithms are reasonably high. Using a false positive model that accounts for resolution scaling, such as the one adapted from Miller and Bloom's hypersphere model [127] in section 4.1, allows a small reduction in the detection threshold, and thus a small increase in detectability; however, fully accounting for quality scaling appears prohibitively complex.

More substantial improvements may be obtained by altering method by which coefficients are selected for embedding. Resolution detectability can be dramatically improved by ensuring that a sufficient number of low resolution coefficients are selected, while quality detectability can be dramatically improved by selecting coefficients that permit a high embedding strength. However, low resolution coefficients do not permit a high embedding strength, so it is not possible to achieve both resolution and quality scalability through coefficient selection alone (section 4.2).

Yet it is possible to achieve both resolution and quality scalability in a single algorithm. This can be done by taking a resolution scalable algorithm and increasing the embedding strength in coefficients where the sensitivity of the human visual system is lower. The `hvs` algorithm developed in section 4.3 is one such algorithm. The embedding strength is increased according to the contrast sensitivity functions developed by Nadenau [131] and the texture scoring function developed in appendix C. The use of this HVS adaptive embedding strength resulted in a substantial improvement in quality detectability (with an average improvement of 4.76 over 65 images) without a substantial reduction in resolution detectability (with an average reduction of 0.0029 over 65 images), relative to the non-adaptive algorithm, while graceful improvement remained high.

Better performance for a single type of scalability can be achieved with an algorithm providing only that type of scalability, in this example the Xia algorithm for resolution

scalability and the Cox algorithm for quality scalability. However, when both resolution and quality scalability are required, the `hvs` algorithm shows substantially superior performance, with quality scalability to $\frac{1}{100}$th the original file size and resolution scalability to $\frac{1}{256}$th area of the original.[19]

Furthermore, although differences in experimental conditions make a direct comparsion questionable, this level of scalability is greater than the best reported results of other resolution and quality scalable watermarking algorithms, (section 3.1.3, page 60) of $\frac{1}{80}$th the original file size [37],[123] and resolution scalability to $\frac{1}{64}$th area [114] of the original.

[19]These comparisons have focused on scalability only, so robustness to attacks has not been examined. This said, some general speculation on the robustness of the `hvs` algorithm is possible. The algorithm of Cox et al. [30] is robust to non-geometric attacks (such as additive noise, JPEG compression, dithering and print-and-scan). Yet it provides little robustness to geometric attacks, unless manual steps are taken to revert the geometric component of the change (e.g. cropping, but restoring cropped sections using the original image). The structure of the `hvs` algorithm is close to that of Cox et al., so it is reasonable to expect that the `hvs` algorithm would, similarly, offer robustness against non-geometric attacks but little robustness against geometric attacks. Of course, performance against specific attacks within these general classes will vary, so if robustness to specific attacks are required for a given application, these attacks should be tested and the algorithm with the best overall performance chosen.

Chapter 5

A Blind Scalable Watermark for JPEG2000: Basic Algorithm

The algorithms examined in the previous chapter were all non-blind, as spread spectrum techniques best lend themselves to non-blind watermarking, where any interference from the host image can be explicitly removed. In this chapter, a blind scalable watermarking algorithm is developed for use with JPEG2000 compression using a quantization watermarking (section 2.1.5.3, page 26) technique.

In a blind watermarking algorithm, the watermark should ideally be unaffected by host image interference [22]. Thus if there is no other source of noise, i.e. if the detection algorithm is given an unprocessed watermarked image I' and the correct embedding parameters Λ, it should be possible to perfectly extract the watermark. That is, the candidate watermark should be identical to the embedded watermark and, if the watermarked image is received complete and unaltered, the candidate and extracted watermarks should match exactly.

In this chapter, this principle is extended to include the case when JPEG2000 resolution and quality scaling have been applied to the image. In traditional approaches the candidate watermark remains identical to the embedded watermark, and the extracted watermark is treated as a noisy version of the embedded watermark, so any approximate match that exceeds some threshold must be allowed. However, it is possible to adjust the candidate watermark according to the observed level of scaling, retaining the exact match between candidate and extracted watermarks.

As was the case in the preceding chapter, no watermark message is used. Instead, the watermark consists of a sequence of non-negative integer elements, drawn from some distribution. By replacing each watermark element with its bit representation, the watermark can also be considered as a sequence of bits. Depending on the type and severity of scaling applied, some wavelet coefficients may have been lost completely or be missing varying numbers of least significant bits (section 2.2.3.7, page 48), this will typically also result in missing watermark bits.

Although the loss of watermark bits during scaling is unavoidable,[1] we can reasonably expect the extracted watermark to be the same as the embedded watermark except for those missing watermark bits. In this case, the candidate watermark should be identical to the embedded watermark with the exception of some deleted bits, corresponding to those watermark bits which should be missing given the amount of scaling.

By generating a candidate watermark that consists of all (and only) non-missing watermark bits, and correctly extracting all (and only) non-missing watermark bits from the scaled image, then, provided no other changes have been made to the image, an exact match between candidate and extracted watermarks may be obtained.

5.1 Design

The primary problem when designing such an algorithm is to maintain the match between the candidate and extracted watermarks without relying on the original image I. The procedures employed for coefficient selection, embedding, and watermark generation in the embedding algorithm must be mirrored by coefficient selection, extraction and candidate watermark generation in the detection algorithm so that the correct candidate and extracted sequences are produced using only the received image I^* and the embedding parameters Λ. For a blind, *scalable* algorithm, this must be achieved despite a majority of image coefficients being lost or altered due to scaling.

5.1.1 Coefficient Selection and Embedding

As with the algorithms of the preceding chapter, watermark embedding will be applied in the wavelet domain, during the JPEG2000 compression process. However, in this chapter, embedding will occur immediately following the quantization step (figure 5.1). This eliminates the quantization step as a source of watermark interference.[2] Note that moving the embedding procedure to follow the quantization step will not cause any quality scaled subimage to exceed its target compression ratio, because the embedding procedure still occurs before the rate distortion optimization step of Tier-2 encoding.

[1] To avoid losing any bits from the watermark after either resolution or quality scaling, the watermarking process would have to be restricted to only those bits guaranteed to be received in the both the lowest resolution and lowest quality subimages. This would result in all higher resolution and quality layers being unprotected by the watermark, as was discussed in section 3.1.1 (page 52). These unprotected layers potentially comprise the vast majority of the image, depending on how highly scaled an image may be before it is deemed of unacceptable quality.

[2] Recall that for the spread spectrum algorithms, embedding was applied before the quantization step. While this did not have a dramatic effect, it did result in a watermark consisting largely of noise in the highest resolution layer for the `comp` algorithm (figure 4.2.2.2.3). Any such noise is easily avoidable by delaying embedding until after the quantization step.

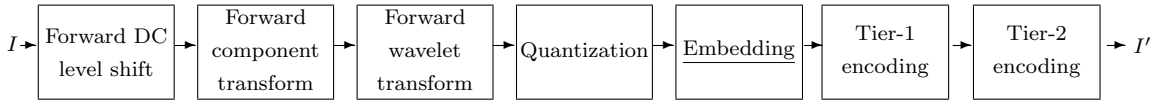


Figure 5.1: Watermark embedding is performed during JPEG2000 compression, immediately following the quantization step.

The aim of coefficient selection is to select a suitable set of quantized coefficients $V = \mathsf{Select}(I, \Lambda)$ for watermark embedding, given watermark parameters Λ, and to ensure that all (and only) coefficients which may still contain watermark bits will be selected during detection. The coefficient selection rule is as follows

$$|v| \geq t = 2^n, \tag{5.1}$$

for the threshold parameter $t = 2^n$ where $n \in \Lambda, n \in \mathbb{N}$.

This rule selects high-magnitude coefficients, which both aids in concealing the watermark and ensures that the watermark is embedded in the important regions of the image. What constitutes a sufficiently high magnitude may be controlled by the choice of the input parameter n. It only uses the value of the coefficient being examined for selection and does not require values for coefficients from higher resolution levels, which may be unavailable in the resolution scaled image, or other coefficients, which may be lost or altered in the quality scaled image. Because the threshold is restricted to be a power of two, coefficients selected in the watermarked image that are not completely lost due to quality scaling,[3] will exceed the threshold and be selected in the quality scaled image.

For a given image I^*, and watermark parameters Λ^* the set of selected coefficients

$$V^* = \mathsf{Select}(I^*, \Lambda^*) \tag{5.2a}$$

$$= \{v \in I^* : |v| \geq t^* = 2^{n^*}\}, \tag{5.2b}$$

contains all coefficients in I^* with magnitudes greater than or equal to the threshold t^*. Note that in this and other descriptions, the superscript * is used to denote the procedure as used with *any* given image and associated parameter set. Thus V^* represents the set of selected coefficients V from the original image, and also V' from the (unscaled) watermarked image, $V^{\mathcal{R}}$ from a resolution scaled image, $V^{\mathcal{Q}}$ from a quality scaled image and even the set of selected coefficients V^Z from an image entirely unrelated to the original.

[3] Even if a coefficient itself has only been partially lost due to quality scaling, the watermark contained in that coefficient may be completely missing, and this cannot be resolved without first determining both the number of watermark bits and the number of missing bits. Rather than determine this during coefficient selection for every coefficient, partially lost coefficients in which watermark bits were embedded but may all have been lost are still included in the set of selected coefficients $V^{\mathcal{Q}}$, with the understanding that these coefficients will be identified at a later stage (section 5.1.5, page 152).

A simple quantize-and-replace embedding scheme is used, with quantization step size $2^j \leq |v|$ for some $j \in \mathbb{N}$. For each selected coefficient $v \in V$, the coefficient is quantized to produce

$$Q_{2^j}(v) = \left\lfloor \frac{|v|}{2^j} \right\rfloor 2^j \geq 0. \tag{5.3a}$$

The coefficient bits removed by quantization are replaced by adding j watermark bits u

$$u \in \mathbb{Z} : 2^j > u \geq 0 \tag{5.3b}$$

to produce the watermarked coefficient

$$v' = sign(v)(Q_{2^j}(v) + u). \tag{5.3c}$$

Note that if the coefficient is not selected, $v \notin V$, then the coefficient is not watermarked and the watermark element u is considered not to exist (denoted $\nexists u$). In such cases, the 'watermarked' coefficient $v' \in I'$ has the value $v' = v$. Similarly, for selected coefficients in which no watermark bits are embedded, $v' = v$ and $\nexists u$. Both situations can alternatively be considered to be embedding with $j = 0$ and $u = 0$.

This embedding formula will ensure that if the original coefficient magnitude $|v|$ satisfies $2^{k-1} \leq |v| < 2^k, k \in \mathbb{N}$, then so does the watermarked coefficient magnitude. I.e., if we let $\bar{v}$ be the smallest non-negative integer power of two that is greater than $|v|$,

$$\bar{v}_i = \bar{v}'_i, \tag{5.3d}$$

thus the coefficient selection rule will perform identically on the original and watermarked images. Similarly, in a resolution or quality scaled image, all coefficients that were selected during embedding and were not 'completely lost' during scaling will be selected during detection:

Let $I' = \mathsf{Embed}(I, \Lambda)$ denote the watermarked image and $I^{\mathcal{Q}} = \mathcal{Q}(I')$ and $I^{\mathcal{R}} = \mathcal{R}(I')$ represent arbitrarily quality scaled and resolution scaled versions respectively. Given the correct embedding parameters $\Lambda^* = \Lambda$, the coefficient selection rule and embedding formula described above allow the following requirements to be satisfied:

- for the unscaled watermarked image, where v' is the coefficient corresponding to $v \in I$,

$$v' \in V' \iff v \in V; \tag{5.4a}$$

- for the resolution scaled image, where $v^{\mathcal{R}}$ is the coefficient corresponding to $v' \in I'$,

$$v^{\mathcal{R}} \in V^{\mathcal{R}} \iff v' \in V' \wedge v^{\mathcal{R}} \in I^{\mathcal{R}}; \tag{5.4b}$$

- for the quality scaled image, where $v^{\mathcal{Q}}$ is the quality scaled coefficient corresponding to $v' \in I'$,

$$v^{\mathcal{Q}} \in V^{\mathcal{Q}} \iff v' \in V' \wedge v^{\mathcal{Q}} \neq 0; \tag{5.4c}$$

provided that, for any watermark element u embedded in a coefficient v,

$$\exists j \in \mathbb{N} \text{ s.t. } 0 \leq u < 2^j \leq |v|. \tag{5.5}$$

A proof for this can be found in section D.3.1 (page 383).

5.1.2 Effects of scaling on a quantized coefficient

Because the watermark is embedded in quantized coefficients in the encoder, it is extracted from quantized coefficients in the decoder (figure 5.2). Before describing the watermark extraction process, the effects of JPEG2000 resolution and quality scaling (section 2.2.3, page 40) on a quantized coefficient v' in a watermarked image I' are reviewed.

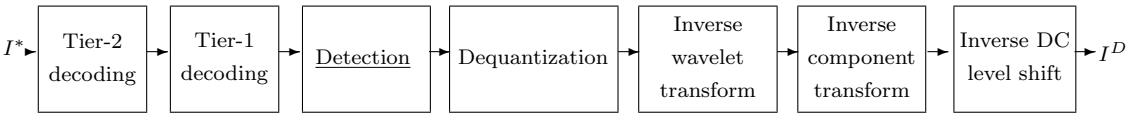


Figure 5.2: Watermark detection is performed during JPEG2000 decompression, immediately preceding the dequantization step.

In resolution scaling, some number K of the R resolution layers will not be received by the decoder. These are the K highest resolution layers. If the watermarked coefficient v', in resolution layer r', belongs to one of these resolution layers, then the corresponding coefficient $v^{\mathcal{R}}$ will not exist in the resolution scaled image; otherwise, it will remain unaffected.

$$r' \geq R - K \iff v^{\mathcal{R}} \notin I^{\mathcal{R}} \tag{5.6a}$$

$$v^{\mathcal{R}} = v' \iff v^{\mathcal{R}} \in I^{\mathcal{R}} \tag{5.6b}$$

In quality scaling, if $v' \in \mathbb{Z}$ is the unscaled coefficient, and the m least significant bits have not been received, the corresponding coefficient at the decoder exists in the quality scaled image

$$v^{\mathcal{Q}} \in I^{\mathcal{Q}} \tag{5.7a}$$

and has the value

$$v^{\mathcal{Q}} = sign(v') \times \begin{cases} \left\lfloor \frac{|v'|}{2^m} \right\rfloor 2^m + \lfloor 2^m \mathrm{r} \rfloor & \left\lfloor \frac{|v'|}{2^m} \right\rfloor \neq 0 \\ 0 & \left\lfloor \frac{|v'|}{2^m} \right\rfloor = 0. \end{cases} \tag{2.20a}$$

where

$$\mathrm{r} : 0 \leq \mathrm{r} < 1 \tag{2.20b}$$

is the JPEG2000 coefficient reconstruction parameter in the decoder.[4]

Note that when $m = 0$, the decoded coefficient is identical to the transmitted coefficient v'. In an unscaled image I' or a resolution (only) scaled image $I^{\mathcal{R}}$, all received coefficients are identical to the corresponding transmitted coefficients and m is taken to be zero for these images.[5]

Recall that a coefficient for which no bits have been received is termed a *completely lost* coefficient. For resolution scaling, these are all coefficients belonging to resolution layers which are not included in the image $v^{\mathcal{R}} \notin I^{\mathcal{R}}$. For quality scaling, these are the coefficients where the number of missing least significant magnitude bits is greater than or equal to the number of significant magnitude bits $v^{\mathcal{Q}} = 0$. If some but not all bits have been received, the coefficient is termed *partially lost*. No watermark bits can be extracted from a completely lost coefficient; while a partially lost coefficient may still contain watermark bits, depending on the value of m.

5.1.3 Watermark Extraction

Given a (potentially scaled) watermarked image I', $I^{\mathcal{R}}$ or $I^{\mathcal{Q}}$, the corresponding set of selected coefficients $V', V^{\mathcal{R}}$ or $V^{\mathcal{Q}}$ will contain all the embedded coefficients that have not been completely lost due to scaling. These selected coefficients will in turn contain every watermark bit that is still present in the image (although, as noted above, not every selected coefficient will necessarily contain watermark bits).

The aim of watermark extraction is to retrieve as much of the watermark as remains in each selected coefficient, given only the (potentially scaled) watermarked image and the correct embedding parameters Λ.

Let $U^d = \{u^d : v^* \in V^*\}$ represent the extracted watermark given selected coefficients V^*, from an image I^*, and watermarking parameters Λ^*. The watermark extraction

[4] This is for the JasPer decoder, as discussed in section 2.2.3.5.1 (page 46); for other decoders some adjustments to the watermarking algorithm may be required.

[5] In fact this is not strictly the case: there exist some images, most notably those containing empty codeblocks in non-empty subbands, for which all coefficients can be perfectly reconstructed without having received all passes for all codeblocks. For such images, the unscaled and resolution (only) scaled versions, i.e. with no missing bits, may only exist in a theoretical sense; any final passes which do not increase the image quality will not be included in the codestream, resulting in a 'quality scaled' image that has identical coefficient values but a non-zero number of missing bits.

formula for the coefficient $v^* \in V^*$ is defined to be

$$u^d = \mathsf{Extract}(v^*, I^*, \Lambda^*) = \begin{cases} |v^*| - Q_{2^{j^*}}(v^*) - \left\lfloor 2^{m^*} \mathrm{r} \right\rfloor & \text{if } m^* < j^* \\ \nexists & \text{if } m^* \geq j^*, \end{cases} \tag{5.8}$$

where m^* and j^* represent the calculated values for the number of missing bits m and the number of embedded watermark bits j, r is the coefficient reconstruction parameter[6] and the symbol $\nexists$ is used to indicate that no watermark is extracted for that element.

For either the unscaled image I' or the resolution scaled image $I^{\mathcal{R}}$, any selected coefficient will be identical to the transmitted coefficient, with $m^* = 0$, so the extraction formula will simplify to be a direct inversion of the embedding formula. For the quality scaled image $I^{\mathcal{Q}}$, a varying number m of least significant bits will have been lost (overwritten with zero) from each selected coefficient. If $m \geq j$, all watermark bits are lost so no watermark is extracted[7]. If $m < j$, the embedding formula is inverted and the m overwritten bits are removed to produce a truncated version of the watermark $u^d \approx u$, where the $j - m$ most significant watermark bits are preserved.

Assuming that both the number of embedded bits and the number of missing bits are calculated correctly ($j^* = j$, $m^* = m$) the watermark extraction procedure described above allows the following properties to be satisfied:

- for the unscaled watermarked image, where $v' \in V'$ is the coefficient corresponding to $v \in V$ and $u^d = \mathsf{Extract}(v', I', \Lambda)$

$$\begin{aligned} &\text{if } j > 0 \text{ then } u^d = u \\ &\text{if } j = 0 \text{ then } \nexists u^d; \end{aligned} \tag{5.9a}$$

- for the resolution scaled image, where $v^{\mathcal{R}} \in V^{\mathcal{R}}$ is the coefficient corresponding to $v' \in I'$ and $u^d = \mathsf{Extract}(v^{\mathcal{R}}, V^{\mathcal{R}}, \Lambda)$

$$\begin{aligned} &\text{if } j > 0 \text{ then } u^d = u \\ &\text{if } j = 0 \text{ then } \nexists u^d; \end{aligned} \tag{5.9b}$$

[6] The value of the coefficient reconstruction parameter r is required for this extraction formula. However, unlike m and j, r is typically a known fixed value for a given JPEG2000 decoder, and does not need to be calculated based on the given image. Thus no * is used with the parameter r.

[7] There is an explicit check for $m^* \geq j^*$ in the extraction formula (5.8) to ensure that nothing is extracted. Without such a check, the extraction formula would evaluate to $-\left\lfloor \frac{\lfloor 2^m r \rfloor}{2^j} \right\rfloor 2^j$, which is not guaranteed to produce an invalid result (i.e. a result outside the range $0 \leq u^d < 2^j$ that constitutes a valid watermark).

- for the quality scaled image, where $v^{\mathcal{Q}} \in V^{\mathcal{Q}}$, the quality scaled coefficient corresponding to $v' \in V'$, has m missing least significant bits and $u^d = \mathsf{Extract}(v^{\mathcal{Q}}, V^{\mathcal{Q}}, \Lambda)$

$$\text{if } m < j \text{ then } u^d = \lfloor \frac{u}{2^m} \rfloor 2^m$$
$$\text{if } m \geq j \text{ then } \nexists u^d. \tag{5.9c}$$

The proof can be found in section D.3.2 (page 388).

Of course, it is first necessary to correctly calculate m^* and j^*. Furthermore, to maintain the blindness property, their values must be determined from the coefficient, image and embedding parameters (v^*, I^*, Λ^*) provided to the extraction algorithm (sections 5.1.6 and 5.1.7).

5.1.4 Watermark Generation

Both embedding and detection procedures require the generation of a watermark. On the encoder side a watermark $U = \{u : v \in V\}$ is to be embedded in the image I, while on the decoder side a candidate watermark U^c must be generated, for comparison with the extracted watermark U^d.

The watermark U should be generated from the original image I and the watermarking parameters Λ such that the watermarked image I' is not perceptibly different from the original image I and the coefficient selection procedure is unaffected.

The candidate watermark U^c represents the watermark we expect to extract if the given image I^* was indeed watermarked using the given parameters Λ^*. This allows us to check the extracted watermark U^d against the candidate U^c to see if they match. Ideally, as was suggested in the introduction to this chapter, if $I^* = I^{\mathcal{F}} = \mathcal{F}(Embed(I, \Lambda))$, where the scaling $\mathcal{F}$ consists only of JPEG2000 resolution and/or quality scaling, and the correct watermarking parameters $\Lambda^* = \Lambda$ are given, the extracted and candidate watermarks should match exactly.

For the unscaled image, we expect to extract the entire watermark, so the candidate watermark should be identical to the embedded watermark. For a resolution or quality scaled image, where we expect some watermark bits to be lost, the candidate watermark should be identical to the embedded watermark with the exception of some deleted bits, that correspond to those watermark bits which were lost during scaling. There are two ways we might achieve this, we can either

- create a candidate watermark in which the lost bits are never generated

or

- generate the candidate watermark as for an unscaled image and explicitly remove the lost watermark bits.

The first method will be used to solve the problems which occur when coefficients are completely lost and the second method for coefficients that are partially lost.

If a resolution or quality scaled coefficient is lost completely, it becomes impossible to tell whether the coefficient magnitude would have exceeded the selection threshold during embedding, and hence whether it would have contained watermark bits. As a result, we are unable to explicitly remove any watermark bits corresponding to such coefficients from the candidate watermark, because we have no way to identify the number of (if indeed there were any) watermark bits embedded in these coefficients. Instead, the watermark bits corresponding to a particular coefficient are generated only if that coefficient is selected in the received image (completely lost coefficients are never selected). This is done by indexing the selected coefficients and generating a single watermark element corresponding to each index, such that for an image $I^{\mathcal{F}} = \mathcal{F}(Embed(I, \Lambda))$, where $\mathcal{F}$ consists only of JPEG2000 resolution and/or quality scaling, the following properties hold:

1. corresponding coefficients in $V^{\mathcal{F}}$ and V will have the same index

$$i^{\mathcal{F}} = \mathsf{Index}(v^{\mathcal{F}}, I^{\mathcal{F}}) = \mathsf{Index}(v_i, I) = i \quad \forall v^{\mathcal{F}} \in V^{\mathcal{F}}; \tag{5.13}$$

2. given the scaled image $I^{\mathcal{F}}$ and the correct watermarking parameters $\Lambda^{\mathcal{F}} = \Lambda$, selected coefficients with the same index will produce the same watermark element

$$u_i^{\mathcal{F}} = \mathsf{G}(v_i^{\mathcal{F}}, i, \Lambda, I^{\mathcal{F}}) = \mathsf{G}(v_i, i, \Lambda, I) = u_i. \tag{5.19}$$

In less extreme cases of quality scaling, the coefficient $v_i^{\mathcal{F}}$ is not completely lost but some m_i watermark bits are missing and cannot be extracted. For these partially lost coefficients, the corresponding element in the generated sequence contributes more bits than we can hope to extract, so we must truncate each such element, to remove those bits, or remove the element if all bits have been lost:

3. given the scaled image $I^{\mathcal{F}}$ and the correct watermarking parameters $\Lambda^{\mathcal{F}} = \Lambda$, if $v_i^{\mathcal{F}}$ is missing m_i bits and a j_i-bit watermark element u_i was embedded then the candidate watermark will be a truncated version of the embedded watermark such that

$$u_i^c = \begin{cases} \lfloor \frac{u_i}{2^{m_i}} \rfloor 2^{m_i} & m_i < j_i \\ \nexists u_i^c & m_i \geq j_i. \end{cases} \tag{5.21}$$

The indexing and element generation procedures (sections 5.1.4.1 and 5.1.4.2), will be used during both embedding and detection, while the candidate truncation procedure (section 5.1.5) is used during detection only.

5.1.4.1 Indexing

The indexing procedure should assign a unique index $i \in \mathbb{Z}$ to each selected coefficient v in the original image I and should produce an identical index $i^{\mathcal{F}} = i$ for the corresponding coefficient $v^{\mathcal{F}}$ in the scaled image $I^{\mathcal{F}}$.

Let the image I consist of C components, each with dimensions[8] $X \times Y$. Each component has been decomposed into R resolutions, using a forward wavelet transform, as is described in section 2.2.3.4 (page 42). In each component $c \in \{0, 1, ..., C-1\}$, the coefficients from resolutions 0 to r are contained in an area with dimensions

$$\begin{aligned} X[r] &= \left\lceil \frac{X}{2^{R-r-1}} \right\rceil \\ Y[r] &= \left\lceil \frac{Y}{2^{R-r-1}} \right\rceil . \end{aligned} \tag{2.14}$$

For $r = 0$ the area with dimensions $X[r], Y[r]$ contains only the LL subband of resolution layer 0. So subband $s = (0, 0)$ has dimensions

$$\begin{aligned} X[0,0] &= X[0] \\ Y[0,0] &= Y[0]. \end{aligned} \tag{5.10}$$

For $r > 0$ the area with dimensions $X[r], Y[r]$ contains all coefficients in resolution layers 0 to $r-1$ (which are used to reconstruct the LL subband of resolution r) and the three subbands $(r, 0), (r, 1)$ and $(r, 2)$ that form resolution layer r. So subband s, of resolution $r > 0$ and orientation o, has dimensions $X[r, o], Y[r, o]$, where

$$\begin{aligned} X[r,0] &= X[r] - X[r-1] \\ Y[r,0] &= Y[r-1] \\ \\ X[r,1] &= X[r-1] \\ Y[r,1] &= Y[r] - Y[r-1] \\ \\ X[r,2] &= X[r] - X[r-1] \\ Y[r,2] &= Y[r] - Y[r-1]. \end{aligned} \tag{5.11}$$

[8] For simplicity, it is assumed that all C components are of equal dimensions. If components are sampled at different rates then the dimensions $X[c], Y[c]$ of individual components will replace the common dimensions X, Y, and CXY will be replaced by $\sum_{d=0}^{C} X[d]Y[d]$. Resolution and subband dimensions will also be component dependent and similar replacements must be made using $X[c, r], Y[c, r]$ for resolutions and $X[c, r, s], Y[c, r, s]$ for subbands.

Let $v^* \in V^*$ be a selected coefficient in an image I^* with coordinates x^* y^* in subband $s^* = (r^*, o^*) \in \{0, 1, 2, \ldots, R^* - 1\} \times \{0, 1, 2\}$ in component $c^* \in 0, 1, 2, \ldots, C^* - 1$, having dimensions X^*, Y^*. The coefficient v^* is assigned the index

$$\begin{aligned} i^* &= \mathsf{Index}(v^*, I^*) \\ &= c^* X^*[r^*] Y^*[r^*] + (C^* - c^*) X^*[r^* - 1] Y^*[r^* - 1] + \\ &\qquad \sum_{a=0}^{a=o^*-1} X^*[r^*, a] Y^*[r^*, a] + y^* X^*[r^*, o^*] + x^* \end{aligned} \tag{5.12}$$

This is equivalent to sequentially indexing each coefficient in the image I^* according to a raster scan of each subband in each component of resolution 0, followed by each subband in each component of resolution 1 and so on, until all coefficients in I^* have been assigned an index, except that the index is calculated directly from the coordinates of v^*, and thus does not require scanning of unselected coefficients. It ensures that each selected coefficient is assigned a unique index $i^* \in \mathbb{N}$, $0 \leq i^* < C^* X^* Y^*$.

The index is completely dependent on coefficient position, image dimensionality and number of resolutions and components. Thus for the unscaled image I' and the quality scaled image $I^{\mathcal{Q}}$, where all these things remain unchanged, the coefficient indices will also remain unchanged. Furthermore, because all coefficients with resolution less than r are assigned lower indices than any coefficient in resolution r, no reordering occurs when the higher resolution coefficients are removed to produce a resolution scaled image $I^{\mathcal{R}}$, so all remaining coefficients will have correct indices.

The indexing formula assigns an index i^* to each coefficient $v^* \in V^*$ so that, given the correct watermarking parameters $\Lambda^* = \Lambda$, the following properties are satisfied:

- for the unscaled watermarked image $I' = \mathsf{Embed}(I, \Lambda)$, where $v'_{i'} \in I'$ is the coefficient corresponding to $v_i \in I$

$$i' = i; \tag{5.13a}$$

- for the resolution scaled image, where $v^{\mathcal{R}}_{i^{\mathcal{R}}} \in I^{\mathcal{R}}$ is the coefficient corresponding to $v'_{i'} \in I'$

$$i^{\mathcal{R}} = i'; \tag{5.13b}$$

- for the quality scaled image, where $v^{\mathcal{Q}}_{i^{\mathcal{Q}}} \in I^{\mathcal{Q}}$, is the coefficient corresponding to $v'_{i'} \in I'$

$$i^{\mathcal{Q}} = i'. \tag{5.13c}$$

The proof can be found in section D.3.3 (page 392).

5.1.4.2 Watermark Element Generation

The watermark generation function G produces the watermark element $u_i = \mathsf{G}(v_i, i, \Lambda, I)$, during embedding, and the candidate watermark element $u^*_{i^*} = \mathsf{G}(v^*_{i^*}, i^*, \Lambda^*, I^*)$, during detection.

A pseudorandom watermark is used, with a generation function similar in form to that of the previous chapter

$$\begin{aligned} u^*_{i^*} &= \mathsf{G}(v^*_{i^*}, i^*, \Lambda^*, I^*) \\ &= \lfloor \alpha^* \alpha^*_{i^*} w^*_{i^*} \rfloor, \end{aligned} \tag{5.14}$$

where $w^*_{i^*} \in \mathbb{R}$ is a pseudorandom element, and $\alpha^*_i \in \mathbb{R}$ is a local embedding strength and $\alpha^* \in \mathbb{R}$ is a global strength parameter.

The pseudorandom element $w^*_{i^*}$ is generated for each selected coefficient $v^*_{i^*}$. It is a function of both the position dependent index i^* and a secret key parameter $sk \in \Lambda$

$$w^*_{i^*} = \mathsf{g}(sk^*, i^*), \tag{5.15}$$

in the range[9] $0 \leq w^*_{i^*} < 2^h$ for some $h \in \mathbb{Z}$. If the correct parameters are given $\Lambda^{\mathcal{F}} = \Lambda$ then the correct element $w^{\mathcal{F}}_{i^{\mathcal{F}}} = w_i$ will be produced from a scaled image $I^{\mathcal{F}}$.

To allow for different imperceptibility requirements, the overall strength of the watermark is adjustable using a single global strength parameter $\alpha \in \Lambda$, in the range $0 \leq \alpha < 1$. Even though the floor function (eqn. 5.14) eventually produces an integer number of embedded bits for each coefficient, real-valued numbers are used for both the strength parameters and for the pseudorandom element; this allows a finer strength adjustment than would the use of integer-valued numbers.

The global strength parameter α will be most effective if all watermark elements are similarly perceptible. Because the ability of the HVS to detect a change in a signal is proportional to the strength of the signal (Weber's Law), watermark imperceptibility may be improved by using a local embedding strength proportional to the coefficient magnitude $\alpha_i \propto |v_i|$. However, because both quality scaling and watermark embedding alter the coefficient magnitude and the magnitude of the original coefficient is not known at the detector, we would be unable to reconstruct the correct value of α_i, resulting in an incorrect candidate watermark. To overcome this, $\bar{x}$ is defined to be the smallest non-negative integer power of two that is greater than $|x|$

$$\bar{x} = 2^k \iff \lfloor 2^{k-1} \rfloor \leq |x| < 2^k, \qquad k \in Z, k \geq 0 \tag{5.16}$$

[9] Because the candidate and extracted watermark elements should match exactly, correlation based detection methods are not needed when only scaling has occurred. Thus there is no requirement that the pseudorandom sequence be zero mean. If a zero mean sequence is desired, w_i must be signed, in which case, the most significant bit of u_i can be reserved to encode the sign, leaving $j_i - 1$ bits for the magnitude.

and $\bar{v}^*_{i^*}$ is used as an approximation to $|v^*_{i^*}|$ that is obtainable not only from the original image I but also from any scaled version $I^{\mathcal{F}}$ of the watermarked image. The constant of proportionality is set to $2^{-(h+1)}$ to ensure that equation 5.5:

$$\exists j \in \mathbb{N} \text{ s.t. } 0 \leq u < 2^j \leq |v|. \tag{5.5}$$

is satisfied, giving

$$\alpha^*_{i^*} = 2^{-(h+1)}\bar{v}^*_{i^*}. \tag{5.17}$$

For simplicity no other HVS based adjustments are made to the watermark. While further adjustments might be added, it must be possible to reconstruct the correct adjustment parameters using either I or any scaled version $I^{\mathcal{F}}$ of the watermarked image and the watermark element u_i must still not disturb the most significant bit of the coefficient v_i.

Thus the generated watermark element is

$$u^*_{i^*} = \mathsf{G}(v^*_{i^*}, i^*, \Lambda^*, I^*) = \lfloor \alpha^* 2^{-(h+1)} \bar{v}^*_{i^*} w^*_{i^*} \rfloor \in \mathbb{Z}, \tag{5.18}$$

where $w^*_{i^*} = \mathsf{g}(sk^*, i^*) \in \mathbb{R}$, in the range $0 \leq w^*_i < 2^h$ for some $h \in \mathbb{N}$, is pseudorandomly generated using the key $sk^* \in \Lambda^*$ and the index i^* and where $\alpha^* \in \Lambda^*$ is a global strength parameter in the range $0 \leq \alpha^* < 1$.

This ensures that

- for the original image, where $v_i \in V$ is a selected coefficient, the generated watermark element u_i satisfies the requirement (section 5.1.1, page 140) that its embedding will not disturb the most significant coefficient bit

$$\exists j_i \in \mathbb{N} \text{ s.t.} 0 \leq u_i < 2^{j_i} \leq |v_i| \tag{5.5}$$

and that neither watermarking nor scaling impact the generated watermark value:

- for the unscaled watermarked image, where $v'_i \in V'$ is the coefficient corresponding to $v_i \in V$

$$\mathsf{G}(v'_i, i, \Lambda, I') = \mathsf{G}(v_i, i, \Lambda, I); \tag{5.19a}$$

- for the resolution scaled image, where $v^{\mathcal{R}}_i \in V^{\mathcal{R}}$ is the coefficient corresponding to $v'_i \in V'$ and $v \in V$

$$\mathsf{G}(v^{\mathcal{R}}_i, i, \Lambda, I^{\mathcal{R}}) = \mathsf{G}(v_i, i, \Lambda, I); \text{ and} \tag{5.19b}$$

- for the quality scaled image, where $v^{\mathcal{Q}}_i \in V^{\mathcal{Q}}$, is the coefficient corresponding to $v'_i \in V'$ and $v \in V$

$$\mathsf{G}(v^{\mathcal{Q}}_i, i, \Lambda, I^{\mathcal{Q}}) = \mathsf{G}(v_i, i, \Lambda, I). \tag{5.19c}$$

The proof can be found in section D.3.4 (page 395).

The quantization step size used in the embedding formula (eqn. 5.3a) can now be determined, as the smallest power of two that allows the entire watermark element to be embedded:

$$2^{j_i} = \bar{u}_i. \tag{5.20}$$

5.1.5 Candidate Truncation

The coefficient selection, indexing and watermark element generation procedures ensure that no watermark bits corresponding to coefficients that have been completely lost due to scaling, and all watermark bits corresponding to the remaining coefficients, appear in the candidate watermark sequence. However, if some of the remaining coefficients have been partially lost, not all watermark bits corresponding to remaining coefficients *should* appear in the candidate sequence.

The aim of candidate truncation is to remove, from the candidate sequence, any watermark bits which are expected to have been lost due to scaling from the selected coefficients, so that, given a scaled watermarked image $I^{\mathcal{F}}$ and the correct watermarking parameters $\Lambda^{\mathcal{F}} = \Lambda$, the candidate watermark $U^c = \{u^c_{i^{\mathcal{F}}} : v^{\mathcal{F}}_{i^{\mathcal{F}}} \in V^{\mathcal{F}}\}$ and extracted watermark $U^d = \{u^d_{i^{\mathcal{F}}} : v^{\mathcal{F}}_{i^{\mathcal{F}}} \in V^{\mathcal{F}}\}$ match exactly.

Recall from section 2.2.3.4 (page 42) that for an unscaled or resolution scaled image, no coefficients are partially lost so the candidate sequence does not need to be altered. If quality scaling has occurred, the m_i least significant bits have been lost from the coefficient, so the m_i least significant watermark bits should be removed from the candidate element to form the truncated candidate element. Finally, if a coefficient is selected but all watermark bits have been lost[10] ($m_i \geq j_i$), then no truncated candidate watermark element should be calculated for that coefficient[11], denoted $\nexists u^c_i$.

The truncated candidate watermark element for the selected coefficient $v^*_{i^*} \in V^*$, from an image I^* given watermarking parameters Λ^*, is calculated by

$$u^c_{i^*} = \mathsf{Candidate}(v^*_{i^*}, i^*, \Lambda^*, I^*) = \begin{cases} \left\lfloor \frac{u^*_{i^*}}{2^{m^*_{i^*}}} \right\rfloor 2^{m^*_{i^*}} & \text{if } m^*_{i^*} < j^*_{i^*} \\ \nexists & \text{if } m^*_{i^*} \geq j^*_{i^*}, \end{cases} \tag{5.21}$$

where $m^*_{i^*}$ and $j^*_{i^*}$ are the number of missing bits and number of embedded watermark bits, respectively, as calculated (sections 5.1.6 and 5.1.7) at the decoder.

Given a scaled but otherwise untampered image $I^{\mathcal{F}}$ and the correct watermarking parameters $\Lambda^{\mathcal{F}} = \Lambda$ then, assuming that both the number of embedded bits and the

[10] The loss of all watermark bits does not necessarily result in exclusion of the coefficient from selection, as was discussed in section 5.1.1

[11] It is necessary explicitly check for $m^*_{i^*} \geq j^*_{i^*}$ to ensure that no truncated candidate element is calculated.

number of missing bits are correct $j^{\mathcal{F}}_{i^{\mathcal{F}}} = j_i$, $m^{\mathcal{F}}_{i^{\mathcal{F}}} = m_i$, the candidate truncation procedure described above allows the following properties to be satisfied:

- for the unscaled watermarked image, where $v'_i \in V'$ is the coefficient corresponding to $v_i \in V$

$$\begin{aligned} &\text{if } j_i > 0 \text{ then } u^c_{i'} = u_i \\ &\text{if } j_i = 0 \text{ then } \nexists u^c_{i'}; \end{aligned} \tag{5.22a}$$

- for the resolution scaled image, where $v^{\mathcal{R}}_i \in V^{\mathcal{R}}$ is the coefficient corresponding to $v'_i \in V'$

$$\begin{aligned} &\text{if } j_i > 0 \text{ then } u^c_{i^{\mathcal{R}}} = u_i \\ &\text{if } j_i = 0 \text{ then } \nexists u^c_{i^{\mathcal{R}}}; \end{aligned} \tag{5.22b}$$

- for the quality scaled image, where $v^{\mathcal{Q}}_i \in V^{\mathcal{Q}}$, is the coefficient corresponding to $v'_i \in V'$,

$$\begin{aligned} &\text{if } m_i < j_i \text{ then } u^c_{i^{\mathcal{Q}}} = \lfloor \frac{u_i}{2^{m_i}} \rfloor 2^{m_i} \\ &\text{if } m_i \geq j_i \text{ then } \nexists u^c_i. \end{aligned} \tag{5.22c}$$

The proof can be found in section D.3.5 (page 400).

5.1.6 Calculating the Number of Missing Bits

For both watermark extraction and candidate truncation to perform correctly, they require the correct number m_i of least significant bits that were lost from the watermarked original coefficient v'_i to produce the received coefficient $v^*_{i^*}$. However, the coefficient v'_i belongs to the watermarked image, which is not available at the decoder, so it is not possible to calculate m_i directly.

What can be calculated is $m^*_{i^*}$, the number of least significant magnitude bits missing from the representation of the coefficient $v^*_{i^*}$. This is done using information from the QCC or QCD marker segments and from the packet headers (section 2.2.3.5, page 45). The QCC or QCD marker segment contains the quantization exponent $E_{s^*_{i^*}}$ for the subband[12] $s^*_{i^*}$ and the number of guard bits $G_{c^*_{i^*}}$ for the component $c^*_{i^*}$. These allow us to determine the maximum number of bit planes

$$M_{s^*_{i^*}} = E_{s^*_{i^*}} + G_{c^*_{i^*}} - 1 \tag{5.23a}$$

[12] Although only one QCC or QCD marker segment applies per component, it contains (implicitly or explicitly) a quantization exponent for every subband in that component. Thus the exponent E can vary with each subband so the subscript $s^*_{i^*}$ is used, rather than $c^*_{i^*}$.

which can be expected from the JPEG2000 codestream for subband s_i of resolution r_i in component c_i. So the received coefficient $v^*_{i^*}$ is composed of at most $M_{s_{i^*}}$ magnitude bits.

The number $Z_{b^*_{i^*}}$ of most significant bit planes in the codeblock $b^*_{i^*}$ that are taken to be all zero is obtained from the first packet header for the codeblock $b^*_{i^*}$. The next X_{i^*} magnitude bits are obtained from $P_{b^*_{i^*}}$ passes in the packet bodies and the remaining $m^*_{i^*}$ magnitude bits are missing from the codestream. The number of missing bits is thus

$$m^*_{i^*} = M_{s^*_{i^*}} - (Z_{b^*_{i^*}} + X_{i^*}). \tag{5.23b}$$

Therefore, in order to calculate $m^*_{i^*}$, we must determine the number of received bits X_{i^*} from the $P_{b^*_{i^*}}$ received passes.

Each bit plane in the codeblock containing $v^*_{i^*}$ has been coded using three coding passes: significance, refinement and cleanup (see the discussion of Tier-1 coding in section 2.2.3.6, page 47). The first received pass is a cleanup pass, which contains one magnitude bit of $v^*_{i^*}$. Exactly one of every three subsequent passes contains an additional bit (precisely which pass contains this bit will depend on context information). So $1 + \left\lfloor \frac{P_{b^*_{i^*}}-1}{3} \right\rfloor$ magnitude bits are contained in the first $1 + \left\lfloor \frac{P_{b^*_{i^*}}-1}{3} \right\rfloor$ 3 passes. This leaves either no remaining passes (when $P_{b^*_{i^*}} \equiv 1 \mod 3$), a significance pass only (when $P_{b^*_{i^*}} \equiv 2 \mod 3$), or a significance and a refinement pass (when $P_{b^*_{i^*}} \equiv 0 \mod 3$).

The set of remaining passes may or may not contribute a single additional magnitude bit: A significance pass will only contain a bit if the coefficient becomes significant (non-zero) in that pass. So if the final pass is a significance pass ($P_{b^*_{i^*}} \equiv 2 \mod 3$), then if and only if the coefficient became non-zero in that pass then it contains an additional bit. A refinement pass will contain a bit if the coefficient is already significant and did not become significant in the immediately preceding (significance) pass. So if the final pass is a refinement pass ($P_{b^*_{i^*}} \equiv 0 \mod 3$), then if and only the coefficient is non-zero we received an additional bit (in either the refinement or the immediately preceding significance pass).

$$X_{i^*} = \begin{cases} 1 + \lfloor \frac{P_{b^*_{i^*}}-1}{3} \rfloor & \text{if } P_{b^*_{i^*}} \equiv 1 \pmod 3 \\ 1 + \lfloor \frac{P_{b^*_{i^*}}-1}{3} \rfloor + 1 & \text{if } P_{b^*_{i^*}} \equiv 2 \pmod 3 \text{ and } \bar{v}^*_{i^*} = 2^{(M_{s^*_{i^*}} - Z_{b^*_{i^*}} - (1 + \lfloor \frac{P_{b^*_{i^*}}-1}{3} \rfloor))} \\ 1 + \lfloor \frac{P_{b^*_{i^*}}-1}{3} \rfloor & \text{if } P_{b^*_{i^*}} \equiv 2 \pmod 3 \text{ and } \bar{v}^*_{i^*} \neq 2^{(M_{s^*_{i^*}} - Z_{b^*_{i^*}} - (1 + \lfloor \frac{P_{b^*_{i^*}}-1}{3} \rfloor))} \\ 1 + \lfloor \frac{P_{b^*_{i^*}}-1}{3} \rfloor & \text{if } P_{b^*_{i^*}} \equiv 0 \pmod 3 \text{ and } v^*_{i^*} = 0 \\ 1 + \lfloor \frac{P_{b^*_{i^*}}-1}{3} \rfloor + 1 & \text{if } P_{b^*_{i^*}} \equiv 0 \pmod 3 \text{ and } v^*_{i^*} \neq 0. \end{cases} \tag{5.23c}$$

Because the number of missing bits is only calculated for selected coefficients $|v^*_{i^*}| \geq t^*$, which are necessarily nonzero (eqn. 5.2b), this can be simplified to

$$X_{i^*} = \begin{cases} 1 + \lfloor \frac{P_{b^*_{i^*}} - 1}{3} \rfloor & \text{if } P_{b^*_{i^*}} \equiv 1 \pmod 3 \\ 1 + \lfloor \frac{P_{b^*_{i^*}} - 1}{3} \rfloor + 1 & \text{if } P_{b^*_{i^*}} \equiv 2 \pmod 3 \text{ and } \bar{v}^*_{i^*} = 2^{(M_{s^*_{i^*}} - Z_{b^*_{i^*}} - (1 + \lfloor \frac{P_{b^*_{i^*}} - 1}{3} \rfloor))} \\ 1 + \lfloor \frac{P_{b^*_{i^*}} - 1}{3} \rfloor & \text{if } P_{b^*_{i^*}} \equiv 2 \pmod 3 \text{ and } \bar{v}^*_{i^*} \neq 2^{(M_{s^*_{i^*}} - Z_{b^*_{i^*}} - (1 + \lfloor \frac{P_{b^*_{i^*}} - 1}{3} \rfloor))} \\ 1 + \lfloor \frac{P_{b^*_{i^*}} - 1}{3} \rfloor + 1 & \text{if } P_{b^*_{i^*}} \equiv 0 \pmod 3 \end{cases}. \tag{5.23d}$$

and then to

$$X_{i^*} = 1 + \left\lfloor \frac{P_{b_{i^*}}}{3} \right\rfloor + \begin{cases} 1 & \text{if } P_{b_{i^*}} \equiv 2 \pmod 3 \text{ and } \bar{v}^*_{i^*} = 2^{(M_{s_{i^*}} - Z_{b_{i^*}} - 1 - \lfloor \frac{P_{b_{i^*}}}{3} \rfloor)} \\ 0 & \text{otherwise.} \end{cases} \tag{5.23e}$$

So the number of missing magnitude bits is

$$\begin{aligned} m^*_{i^*} =& M_{s^*_{i^*}} - Z_{b^*_{i^*}} - 1 - \left\lfloor \frac{P_{b^*_{i^*}}}{3} \right\rfloor - \\ & \begin{cases} 1 & \text{if } P_{b^*_{i^*}} \equiv 2 \text{ (mod 3) and } \bar{v}^*_{i^*} = 2^{(M_{s^*_{i^*}} - Z_{b^*_{i^*}} - 1 - \lfloor \frac{P_{b^*_{i^*}}}{3} \rfloor)} \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{5.24}$$

Given a scaled, but otherwise untampered, watermarked image $I^{\mathcal{F}}$, the above marker segment and packet header information accurately represents the scaling of v'_i to produce $v^{\mathcal{F}}_{i^{\mathcal{F}}}$. Thus the number of least significant magnitude bits $m^{\mathcal{F}}_{i^{\mathcal{F}}}$, missing from the representation of the coefficient $v^{\mathcal{F}}_{i^{\mathcal{F}}}$, will be identical to the number of bits that were lost from the watermarked coefficient v'_i to produce $v^{\mathcal{F}}_{i^{\mathcal{F}}}$, that is

$$m^{\mathcal{F}}_{i^{\mathcal{F}}} = m_i. \tag{5.25}$$

Note that if $I^{\mathcal{F}}$ is an unscaled or only resolution scaled image that is otherwise untampered, and thus the full number of passes $P_{b^{\mathcal{F}}_{i^{\mathcal{F}}}} = 1 + 3(M_{s^{\mathcal{F}}_{i^{\mathcal{F}}}} - Z_{b^{\mathcal{F}}_{i^{\mathcal{F}}}} - 1)$ is received, then $m^{\mathcal{F}}_{i^{\mathcal{F}}} = 0$.

5.1.7 Calculating the Number of Watermark Bits

For both watermark extraction and candidate truncation to perform correctly, they require the correct number j_i of watermark bits embedded in the coefficient v_i. However, the coefficient v_i belongs to the original image, and is not available at the decoder, so it is not possible to calculate j_i directly.

What can be calculated, is the number of watermark bits j^*_i that would have been embedded in to produce received coefficient $v^*_i \in V^*$, if the embedding parameters were

Λ^*. The candidate watermark element $u^*_{i^*} = \mathsf{G}(v^*_{i^*}, i^*, \Lambda^*, I^*)$ (section 5.1.4.2) represents what would have been embedded to produce the coefficient v^*_i. Thus the number of watermark bits that would have been embedded is simply the number of bits in $u^*_{i^*}$:

$$j^*_{i^*} = \log_2(\bar{u}^*_{i^*}) \tag{5.26}$$

Given a scaled, but otherwise untampered, watermarked image $I^{\mathcal{F}}$, and the correct watermarking parameters $\Lambda^{\mathcal{F}} = \Lambda$, either the candidate watermark element is equal to the embedded watermark element

$$\begin{aligned} u^{\mathcal{F}}_{i^{\mathcal{F}}} &= \mathsf{G}(v^{\mathcal{F}}_{i^{\mathcal{F}}}, i^{\mathcal{F}}, \Lambda, I^{\mathcal{F}}) \\ &= \mathsf{G}(v^{\mathcal{F}}_{i^{\mathcal{F}}}, i^{\mathcal{F}}, \Lambda^{\mathcal{F}}, I^{\mathcal{F}}) && \Lambda^{\mathcal{F}} = \Lambda \\ &= \mathsf{G}(v^{\mathcal{F}}_{i}, i, \Lambda, I^{\mathcal{F}}) && \text{eqns. (5.13)} \\ &= \mathsf{G}(v^{\mathcal{F}}_{i}, i, \Lambda, I) && \text{eqns. (5.19)} \\ &= u_i \end{aligned} \tag{5.27a}$$

and therefore they must both contain the same number of bits

$$\begin{aligned} j^{\mathcal{F}}_{i^{\mathcal{F}}} &= \log_2(\bar{u}^{\mathcal{F}}_{i^{\mathcal{F}}}) \\ &= \log_2(\bar{u}_i) \\ &= \log_2(2^{j_i}) && \text{eqn. (5.20)} \\ &= j_i. \end{aligned} \tag{5.27b}$$

or the candidate watermark element does not exist, in which case no corresponding watermark element is extracted, so both the extracted and the candidate watermark elements have 0 bits

$$\nexists u^{\mathcal{F}}_{i^{\mathcal{F}}}, \nexists u_i \iff j^{\mathcal{F}}_{i^{\mathcal{F}}} = j_i = 0. \tag{5.27c}$$

Note that, unlike the calculation of $m^*_{i^*}$, which does not require the watermarking parameters, without the correct watermarking parameters Λ, the candidate watermark will be incorrect and $j^*_{i^*}$ may be different to j_i, resulting in watermark extraction errors. (Candidate truncation errors may also result from an incorrect value of $j^*_{i^*}$ but that is of little importance as, to reach a point where $j^*_{i^*} \neq j_i$, the (non-truncated) candidate $u^*_{i^*}$ must already be incorrect.)

5.1.8 The Blind Scalable Watermarking Algorithm

Although the watermark detection algorithm can be applied to any image I^*, of primary interest is its use with $I^{\mathcal{F}}$ a JPEG2000 scaled version of a watermarked image $I' = \mathsf{Embed}(I, \Lambda)$, which itself is the original image I watermarked using parameters Λ.

The embedding algorithm:

Embed

Input: $I, \Lambda = \{\alpha, t, sk\}$

Output: I'

- *Coefficient Selection* $V = \{v \in I : |v| \geq t = 2^n\} \qquad n \in \mathbb{N}$
- *Indexing* $i = cX[r]Y[r] + (C-c)X[r-1]Y[r-1] + \sum_{a=0}^{a=o-1} X[r,a]Y[r,a] + yX[r,o] + x$
- *Watermark Generation* $U = \{u_i : v_i = I(c,r,o,x,y) \in V\}$
- *Element Generation*
$$u_i = \mathsf{G}(v_i, i, \Lambda, I)$$
$$= \lfloor \alpha\, 2^{-(h+1)}\, \bar{v}_i\, \mathsf{g}(i, sk) \rfloor \qquad 0 \leq \alpha < 1, \quad 0 \leq \mathsf{g}(i, sk) < 2^h, \quad h \in \mathbb{Z}$$
$$j_i = \log_2(\bar{u}_i)$$
$$\text{where } \bar{x} = 2^k \iff \lfloor 2^{k-1} \rfloor \leq |x| < 2^k, \quad k \in Z, k \geq 0$$
- *Coefficient Quantization* $Q_{2^{j_i}}(v_i) = \left\lfloor \frac{|v_i|}{2^{j_i}} \right\rfloor 2^{j_i}$
- *Watermark Embedding* $v_i' = \text{sign}(v_i)(Q_{2^{j_i}}(v_i) + u_i)$
- $$I'(c,r,o,x,y) = \begin{cases} v_i' & v_i \in V \\ v_i & v_i \notin V \end{cases}$$

and the detection algorithm:

Detect

Input: $I^*, \Lambda^* = \{\alpha^*, t^*, sk^*\}$

Output: True / False

- *Coefficient Selection* $V^* = \{v \in I^* : |v| \geq t^* = 2^{n^*}\} \qquad n^* \in \mathbb{N}$
- *Indexing*
$$i^* = c^*X^*[r^*]Y^*[r^*] + (C^* - c^*)X^*[r^*-1]Y^*[r^*-1] + \sum_{a=0}^{a=o^*-1} X^*[r^*,a]Y^*[r^*,a] + y^*X^*[r^*,o^*] + x^*$$
- *Element Generation*
$$u_{i^*}^* = \mathsf{G}(v_{i^*}^*, i^*, \Lambda, I^*)$$
$$= \lfloor \alpha^*\, 2^{-(h^*+1)}\, \bar{v}_{i^*}^*\, \mathsf{g}(i^*, sk^*) \rfloor$$
$$0 \leq \alpha^* < 1, \quad 0 \leq \mathsf{g}(i^*, sk^*) < 2^{h^*}, \quad h^* \in \mathbb{Z}$$
- *Missing Bits Calculation*
$$m_{i^*}^* = M_{s_{i^*}^*} - Z_{i^*} - 1 - \left\lfloor \frac{P_{b_{i^*}^*}}{3} \right\rfloor - \begin{cases} 1 & \text{if } P_{b_{i^*}^*} \equiv 2 \pmod 3 \text{ and } \bar{v}_{i^*}^* = 2^{(M_{s_{i^*}^*} - Z_{i^*} - 1 - \lfloor \frac{P_{b_{i^*}^*}}{3} \rfloor)} \\ 0 & \text{otherwise.} \end{cases}$$
- *Watermark Bits Calculation* $j_{i^*}^* = \log_2(\bar{u}_{i^*}^*)$
- *Watermark Extraction* $U^d = \{u_{i^*}^d : v_{i^*}^* \in V^*\}$
$$u_{i^*}^d = \begin{cases} |v_{i^*}^*| - Q_{2_{i^*}^{j^*}}(v_{i^*}^*) - \lfloor 2^{m_{i^*}^*} \mathrm{r} \rfloor & \text{if } m_{i^*}^* < j_{i^*}^* \\ \nexists & \text{if } m_{i^*}^* \geq j_{i^*}^*, \end{cases}$$
- *Candidate Truncation* $U^c = \{u_{i^*}^c : v_{i^*}^* \in V^*\}$
$$u_{i^*}^c = \begin{cases} \left\lfloor \frac{u_{i^*}^*}{2^{m_{i^*}^*}} \right\rfloor 2^{m_{i^*}^*} & \text{if } m_{i^*}^* < j_{i^*}^* \\ \nexists & \text{if } m_{i^*}^* \geq j_{i^*}^*, \end{cases}$$
- $$\textit{Output} = \begin{cases} \text{True} & \text{if } U^c = U^d \\ \text{False} & \text{if } U^c \neq U^d \end{cases}$$

have been designed such that if the image $I^{\mathcal{F}} = \mathcal{F}(\mathsf{Embed}(I, \Lambda))$, is a scaled watermarked image where $\mathcal{F}$ consists only of JPEG2000 resolution and/or quality scaling, and the correct watermarking parameters $\Lambda^{\mathcal{F}} = \Lambda = \{\alpha, t, sk\}$ are provided, then there will be an exact match between the candidate and extracted watermarks U^c and U^d and the result of $\mathsf{Detect}(I^*, \Lambda))$ will be True (the proof can be found in section section D.3.6, page 403). A worked example of the operation of the algorithm as it applies to a single coefficient is provided in appendix D.1.

5.2 Evaluation of the Basic Algorithm

An important feature of this algorithm is the exact match between candidate and extracted watermarks under resolution and quality scaling. Given a resolution or quality scaled image, the detection algorithm determines the expected number of bits missing from each coefficient, and truncates elements of the candidate watermark U^* accordingly, to produce a new version U^c of the candidate watermark that is appropriate to the scaled image. The candidate and extracted vectors, U^c and U^d, may have reduced dimensionality or reduced magnitude, but will match each other exactly.

If, on the other hand, some processing other than resolution or quality scaling is applied to the watermarked image, the exact match between the candidate and extracted marks can no longer be ensured, and there are likely to be mismatches.

This suggests that the algorithm may be suitable in a semi-fragile image authentication scenario such as the following:

> A company makes its image collection available online via a subscription service. The images are scalably compressed using JPEG2000 so that subscribers may access these images at the resolution and quality that best suits their display and bandwidth preferences, but only a single version of the image need be stored. To allow the images to be authenticated, each subscriber is provided with a watermark detector, which may be used to check the integrity of an image when it is received.

Note that in a scenario such as this one, the only types of non-malicious processing to be expected are resolution and quality scaling, which are used to tailor the image to suit the target device. As a result, there is no reason for the authentication system to be tolerant of operations such as rotation or blurring provided it can distinguish these from scaling. Thus if the candidate and extracted watermarks differ by even one bit, the watermark is assumed to be inauthentic.

However, an exact match between candidate and extracted watermarks does not ensure that an authentic watermark was embedded, since it is possible to obtain an exact match

purely by chance (section 5.2.2.1.1, page 186). Thus, in addition to the exact match property, the watermark must still be detectable, in the sense that there should be reasonable confidence that an authentic watermark was indeed embedded. Similarly, there should be graceful improvement, so that confidence in the authenticity of the image increases as more layers are received.

Finally, to be useful in an image authentication scenario, the watermark must be fragile to deliberate tampering. This includes not only tampering in the spatial domain, but also tampering in the wavelet domain using more sophisticated methods, such as mark transfer [109] or Holliman-Memon [72] attacks.

In this section the watermarking algorithm of section 5.1 is evaluated, to determine to what extent it satisfies these requirements. This evaluation is divided into three parts. Section 5.2.1 considers the correctness and fragility of the algorithm; it demonstrates the exact match property under resolution and quality scaling and examines the sensitivity of the watermark to changes in the detection key and to a series of image manipulations. Section 5.2.2 examines the scalability of the algorithm, in terms of appropriate detectability and graceful improvement measures. Section 5.2.3 focuses on various methods of deliberate tamping tampering, each of which is applied to a single image to uncover the weaknesses in the algorithm.

5.2.1 Correctness and Fragility

To evaluate the correctness of the algorithm in resolution or quality scaled content and it's fragility to various attacks, the bit error rate (BER) between the candidate and extracted watermarks is measured, for a number of different images and embedding keys, from reduced resolution and reduced quality subimages.

The algorithm should behave correctly for the full image, or whenever only resolution or quality scaling have been applied. This means that detection using the correct parameters should result in exact match between candidate and extracted watermarks. It also means that detection using an incorrect secret key should result in a substantial mismatch between candidate and extracted watermarks, even when that key is close to the one used for embedding.

The watermark should be fragile to changes to the image other than resolution and quality scaling. In this section we examine fragility to image processing in which no attempt is made to defeat the watermarking algorithm, more deliberate tampering is considered in section 5.2.3.

Exact Match Under Resolution and Quality Scaling

There should be an exact match between candidate watermark U^c and extracted watermark U^d under resolution and quality scaling. This will ensure that the algorithm generates no false alarms, in which an authentic image is incorrectly classified as inauthentic.

An exact match can be identified by a BER of zero. If all untampered scaled subimages have a BER of zero, we can be confident that the algorithm is correctly implemented. Furthermore, if a BER of zero is obtained when there are lost watermark coefficients, the synchronization between the embedding and detection algorithms under scaling can be verified.

Key Sensitivity

Small differences between the embedding and detection keys should produce large differences between the extracted and candidate watermarks. This can be tested by measuring the BER between the candidate and extracted watermarks for a range of detection keys close to but different from the embedding key.

Ideally, candidate and extracted watermarks corresponding to different keys should be completely unrelated. If this is the case, and each consists of an equal number of 0s and 1s, BERs of 50% should be obtained.

Fragility to Processing

The watermark should be fragile to processing other than resolution and quality scaling. This can be tested by processing each watermarked image and measuring the BER between the candidate and extracted watermarks.

The BER is expected to vary according to the type and severity of processing; however, fragility is indicated by *any* non-zero BER, We first examine the bit error rate after decoding the watermarked images to RGB format and then re-encoding to JPEG2000 format without any other changes. This establishes a baseline from which to consider other attacks where an image is decompressed, modified and recompressed.

These other attacks are applied using the Checkmark benchmarking suite [139]. We divide these into nongeometric and geometric attacks and the copy attack. Nongeometric and geometric attacks are characterized by different patterns of watermark error rates in quality scaled images; the copy attack is the only attack in the Checkmark suite that can be considered a deliberate attempt to defeat a fragile watermarking algorithm.

Not all attacks in the Checkmark suite are used. Attacks which specifically attempt to destroy a watermark (for example dpr1 and templateremove) are excluded as these attacks are meaningless for an authentication algorithm. In most cases we apply only the least distortive attack in each class (for example gaussian1, a 3x3 filter, is used rather than the more distortive gaussian2, a 5x5 filter) as fragility to the less distortive version of an attack indicates fragility to the more distortive versions.

5.2.1.1 Experimental Framework

The above properties are all tested using similar experiments. The following section describes how the experiments are performed in the majority of tests. If the testing for a particular property involves any deviations from the method described here, the differences are noted in the section on that property.

Embedding

The watermark is embedded as part of the JPEG2000 compression process, using the algorithm described in section 5.1.8. JPEG2000 compression is performed using the JasPer algorithm [1] with 6 resolution layers and 8 quality layers with compression rates 0.0025, 0.005, 0.01, 0.02, 0.03, 0.04, 0.05 and 0.9999.

So that each watermark has the same level of perceptibility, the embedding strength is adjusted according to the perceptual difference between the original and watermarked image. The perceptual difference is computed using the S-CIELAB metric, which was developed by Zhang and Wandell [213] as spatial extension to the CIELAB colour difference metric to allow its use in digital images, updated to use CIEDE 2000 ΔE, which better approximates the HVS [167]. For each watermark, the global embedding strength α is adjusted so that the 99th percentile[13] of the S-CIELAB CIEDE 2000 error[14] between the original image and the full watermarked image is $4\Delta E$.

The process is repeated for 20 original images and 10 secret keys to produce 200 watermarked images.

Attacking

Some form of attack, such as cropping, is applied to the watermarked images. Unless otherwise stated, this involves decompression of each image into RGB format, application of the attack and subsequent recompression using JPEG2000 and the original compression parameters.

[13]The 99th percentile (rather than the average) ΔE value is used because it is thought[129] to provide a closer match to subjective perceptibility.

[14]The settings used for the calculation of the S-CIELAB CIEDE 2000 error were those of a Dell 1702FP (Analogue) monitor, 96dpi, viewed at 46cm.

The vast majority of these attacks are applied using a modified version of the Checkmark [139] benchmarking program. The Checkmark program traditionally stores all attacked images in JPEG compressed format; because this detection algorithm expects a JPEG2000 compressed image, this has been changed to JPEG2000 format, meaning that each attacked image is recompressed using the same compression parameters as the embedding phase.

Detection

For each watermarked image, watermark detection is performed for all 6 resolution scaled subimages and all 8 quality scaled subimages. In most cases, the detection key is the same as the embedding key.

The watermark is treated as a sequence of bits, and any bit for which the candidate watermark u^c and the extracted watermark u^d do not match is counted as a bit error. To calculate the total number of bit errors corresponding bits in the sign-magnitude representations of corresponding candidate and extracted watermark elements are compared, and all mismatching bits are counted. For example, a candidate watermark element $u_i^c = -28$ and an extracted watermark element $u_i^d = 13$ would contribute 3 bit errors to the total.

u_i^c	-28	$\mathbf{-1}1110\mathbf{0}$
u_i^d	13	$\mathbf{+0}1101$

If all of an element's magnitude bits have been lost due to scaling but the sign bit is still present then we still check for (and count) mismatching sign bits.

The bit error rate (BER) is the total number of bit errors, divided by the total number of (non-missing) extracted bits. The average BER and the total number of extracted bits, across all original images and all embedding keys, are reported for each scaled subimage. More detailed information, such as the BERs for individual images, may also be of interest and is provided in section D.4 (page 408).

5.2.1.2 Exact Match under Resolution/Quality Scaling

To test the claim that the candidate and extracted watermark sequences match exactly regardless of resolution and quality scaling, the experiment described in TCExperiment is performed, but the attack phase is skipped, so no attacks are applied to the watermarked images. If the exact match property holds we should see a BER of zero, not only for the full image but also for any resolution or quality scaled subimage.

Results

The bit error rate for all watermark detections in both resolution and quality decompositions is zero. That is, for all detections, for resolution scaled subimages (table 5.1) and quality scaled subimages (table 5.2), there was an exact match between the candidate and extracted watermark sequences. This suggests that the implementation is correct, and the exact match property holds.

The number of extracted bits increases with each additional layer, so confidence in the authenticity of the image increases with the value of the image. Note that the results shown are compiled from detections across 200 subimages. For a single subimage, the number of extracted bits will be far smaller than those in tables 5.1 and 5.2. If too few bits are extracted, a zero BER should not be considered authentic. As a result, the watermark is not considered authentic, for some tested images, once quality scaling exceeds a compression rate of 0.01, as less than 30 bits can be extracted (section D.4.1.1, page 408). An authentic watermark *is* detectable for all images at all tested levels of resolution scaling (down to $\frac{1}{1024}$th the original area).

Table 5.1: Total bit errors and extracted bits for resolution scaled subimages, data for each row is obtained from the 200 subimages with the shown number of resolution layers.

resolution layers	bit errors	extracted bits	BER
1	0	572494	0
2	0	900753	0
3	0	168899	0
4	0	2424497	0
5	0	3145409	0
6	0	3465796	0

Table 5.2: Total bit errors and extracted bits for quality scaled subimages, data for each row is obtained from the 200 subimages with the shown number of quality layers.

quality layers	bit errors	extracted bits	BER
1	0	29531	0
2	0	62076	0
3	0	129196	0
4	0	285263	0
5	0	486682	0
6	0	720704	0
7	0	950015	0
8	0	3465796	0

5.2.1.3 Key Sensitivity

To test that small differences between the embedding and detection keys result in large differences between the candidate and extracted watermarks, an experiment is performed using the method described in section 5.2.1.1 (page 161) with the processing phase skipped so there are no attacks. However, rather than use the same embedding and detection keys, we use a single embedding key $sk_e = 101$ and the detection process is performed using 100 sequential detection keys $sk_d \in \{1, 2, ..., 100\}$ on each quality or resolution scaled subimage.

If the algorithm is sensitive to changes in the detection key, error rates for all tested (incorrect) detection keys should be high, and there should be no trend towards lower error rates as the detection key nears the embedding key.

Results

Figures 5.3 and 5.4 show, for the first 5 images,[15] the BERs for all scaled subimages, plotted against the detection key used.

The colour of each point represents the number of layers contained in that subimage. There is a clear pattern of lower bit error rates for detection keys $k_d = 6n + 3$ and $k_d = 6n+5$, $n \in \mathbb{N}$; the error rate for the $k_d = 6n+5$ sequence approaches 20% as the detection key nears the embedding key. So the key sensitivity property is not satisfied. These lower than expected error rates appear to be the result of the poor choice of pseudorandom number generation function used in this implementation. For each selected coefficient, a seed is generated by adding a function of the image and coefficient location parameters to the secret key. The watermark element is the result of a single call to rand after first calling srand with the calculated seed. This means that each watermark element is the first element of a pseudorandom sequence. In many implementations of rand, including this one, the first number generated from a newly re-seeded random number generator does not have the desired randomness across different initial seeds. As a result, the seeds generated using the detection key do not produce watermark elements sufficiently random relative to those generated using the embedding key, causing the observed low bit error rates. Replacing calls to rand with a pseudorandom number generator that does not share this weakness and/or generating pseudorandom numbers for each index in order, to eliminate re-seeding, should result in consistent BERs for all incorrect keys. The proposed changes are made for the algorithm in the following chapter, the results of the key sensitivity test may be found in section 6.4.2.2 (page 247).

[15] The full, 20-image version of this graph is too cluttered to be clear.

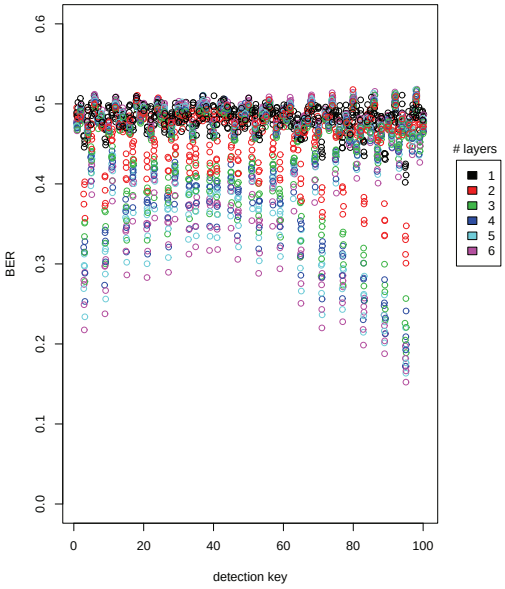


Figure 5.3: Bit error rates for resolution scaled subimages with incorrect detection key. As the detection key nears the embedding key, the BER falls. This makes it easier to guess the embedding key.

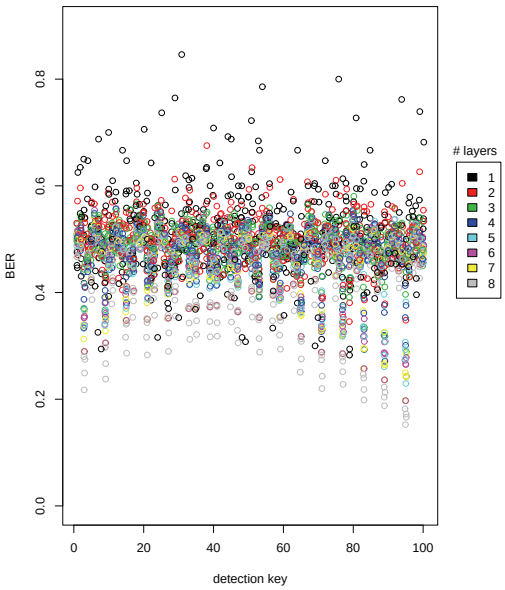


Figure 5.4: Bit error rates for quality scaled subimages with incorrect detection key – As the detection key nears the embedding key, the BER falls. This makes it easier to guess the embedding key.

5.2.1.4 Recompression

Any processing other than resolution or quality scaling may result in mismatched watermark bits. Measuring the mismatch between candidate and extracted watermarks after decompression and recompression of the image provides baseline error rates with which other attacks (involving decompression, an attack and recompression) can be compared. The attack phase in this experiment consists only of decompression and recompression using JPEG2000 and the same compression parameters as were used during embedding (section 5.2.1.1, page 161).

Results

The decompression and recompression process causes watermark detection errors at all levels of scaling. The error rate for the resolution scaled subimages is between 37 and 39 percent with no clear trend in overall rates across all images (table 5.3).

Table 5.3: Total bit error rates after recompression for resolution scaled subimages.

resolution layers	bit errors	extracted bits	BER
1	219836	571931	0.3844
2	356467	900212	0.396
3	661895	1687794	0.3922
4	934796	2422404	0.3859
5	1191375	3142501	0.3791
6	1288126	3462423	0.372

Table 5.4: Total bit error rates after recompression for quality scaled subimages.

quality layers	bit errors	extracted bits	BER
1	2242	29062	0.0771
2	5804	62601	0.0927
3	15714	132542	0.1186
4	53674	325463	0.1649
5	110268	530336	0.2079
6	208655	842128	0.2478
7	247151	1001952	0.2467
8	1288126	3462423	0.3720

There is, however, a clear decrease in the overall bit error rate as the number of quality layers decreases (table 5.4). This occurs because recompression primarily affects the least significant bit planes of the quantized coefficients, which are generally included in the higher quality layers. Similar effects can be observed in individual images (section D.4.1.3, page 411).

5.2.1.5 Nongeometric Attacks

The fragility of the algorithm is tested against a subset of the nongeometric attacks in the Checkmark [139] benchmarking suite. Each of these attacks causes changes across the entire image but none alter the image dimensions or orientation. The attacks used are gaussian1, hardthresh1, jpegcompression1_j100, jpegcompression1_j40, medfilt1, midpoint1, trimmedmean1, sampledownup1, sharpening1 and waveletcompression1_W10. With the exception of jpegcompression1_j40, each attack is the least distortive of its type. The Checkmark suite traditionally stores all attacked images in JPEG compressed format, this step has been eliminated in favour of recompression into JPEG2000 format, using the same compression parameters that were used during the embedding phase.

The watermark should be fragile to all attacks, although the degree of fragility will depend on the attack strength and characteristics. Note that, because the attack strengths are fixed for each type of attack, i.e. are not adjusted to produce a consistent output image quality, no attempt is made to compare the error rates between different types of attacks.

Gaussian

The Gaussian attack smooths the image using a 3x3 gaussian filter, producing a softened image.

The error rates are only slightly higher than the baseline decompress/recompress on subimages with few resolution or quality layers, but this difference increases as more layers are added. This is apparent in the average error rates across multiple originals (tables 5.5 and 5.6) and also occurs within each watermarked image (section D.4.1.4, page 413).

The increase in bit error rate for higher resolution layers occurs because the gaussian filter is low pass and shaped to eliminate progressively more data at higher frequencies. The increase in bit error rate for higher quality layers occurs because the changes to the image are minor, so the majority of the differences occur in the less significant bits of the coefficients, which correspond to the higher quality layers.

Table 5.5: Total bit error rates after Gaussian filtering for resolution scaled subimages.

resolution layers	bit errors	extracted bits	BER
1	220775	571933	0.386
2	366608	900045	0.4073
3	731148	1685013	0.4339
4	1091478	2384222	0.4578
5	1387103	2948222	0.4705
6	1456729	3083128	0.4725

Table 5.6: Total bit error rates after Gaussian filtering for quality scaled subimages.

quality layers	bit errors	extracted bits	BER
1	2638	31042	0.085
2	7555	66630	0.1134
3	24386	148079	0.1647
4	112133	401117	0.2796
5	206420	622706	0.3315
6	331380	911235	0.3637
7	411125	1070104	0.3842
8	1456729	3083128	0.4725

Hard Thresholding

The hard thresholding attack applies adaptive wiener filtering, with 3x3 neighbourhood estimates of the noise mean and variance. Parts of the residual between the watermarked image and its filtered version are added back to the filtered image to produce the output image. Residuals which exceed some threshold are added in full; for residuals which do not exceed the threshold nothing is added.

Table 5.7: Total bit error rates after hardthresh filtering for resolution scaled subimages.

resolution layers	bit errors	extracted bits	BER
1	229389	571815	0.4012
2	382241	898650	0.4254
3	762594	1666185	0.4577
4	1084037	2295730	0.4722
5	1322567	2762019	0.4788
6	1428460	2969978	0.481

Table 5.8: Total bit error rates after hardthresh filtering for quality scaled subimages.

quality layers	bit errors	extracted bits	BER
1	3183	32606	0.0976
2	9348	68664	0.1361
3	36212	161213	0.2246
4	142809	415816	0.3434
5	272253	684574	0.3977
6	375940	895519	0.4198
7	507334	1157361	0.4384
8	1428460	2969978	0.481

The pattern of error rates is similar to that of gaussian filtering, showing increased error rates with additional resolution or quality layers. This occurs for both the average results (tables 5.7 and 5.8) and the results for individual images (section D.4.1.5, page 414). This is because the wiener filter is also low pass, so there is little alteration of the low resolution layers, and the thresholding removes only the less important residuals, so the majority of changes occur in the higher quality layers.

JPEG compression

Separate JPEG compression attacks are applied to two repetitions of the experiment. In the first case the least distortive quality setting of 100 is used. In the second case a lower quality setting of 40 is used.

Table 5.9: Total bit error rates after JPEG100 for resolution scaled subimages.

resolution layers	bit errors	extracted bits	BER
1	216589	571925	0.3787
2	351612	899339	0.391
3	652204	1672174	0.39
4	894697	2383490	0.3754
5	1113763	3094394	0.3599
6	1200833	3413166	0.3518

Table 5.10: Total bit error rates after JPEG100 for quality scaled subimages.

quality layers	bit errors	extracted bits	BER
1	2447	29374	0.0833
2	6492	63028	0.103
3	18986	135518	0.1401
4	71689	347750	0.2062
5	134005	558065	0.2401
6	226106	851288	0.2656
7	274894	1026180	0.2679
8	1200833	3413166	0.3518

JPEG compression with quality 100 produces similar error rates to JPEG2000 re-compression without any manipulation (compare table 5.9 with table 5.3, page 166). No increase in error rate is evident as more resolution layers are added.

Table 5.11: Total bit error rates after JPEG40 for resolution scaled subimages.

resolution layers	bit errors	extracted bits	BER
1	252851	572419	0.4417
2	409948	900045	0.4555
3	770579	1654105	0.4659
4	1102383	2338298	0.4714
5	1449668	3024374	0.4793
6	1581898	3279415	0.4824

Table 5.12: Total bit error rates after JPEG40 for quality scaled subimages.

quality layers	bit errors	extracted bits	BER
1	3750	29744	0.1261
2	11407	63797	0.1788
3	41022	150626	0.2723
4	118213	341650	0.346
5	202906	526869	0.3851
6	343160	825904	0.4155
7	423751	988302	0.4288
8	1581898	3279415	0.4824

JPEG compression with quality 40, on the other hand, produces error rates patterns similar to gaussian and hard-thresholding. Error rates for JPEG40 compression do increase with the addition of both resolution (table 5.11) and quality (table 5.12) layers. This is because at lower JPEG quality settings, quantization increases and fewer high resolution DCT coefficients are preserved.

Median

The median attack applies median filtering to the image using a 2x2 window. Non-integer medians are rounded down.

Error rates follow the same general pattern as the other nongeometric attacks, with a clear increase in error rates as the number of layers increases. This is apparent both from the average results across all images (tables 5.13 and 5.14) and the individual image results (section D.4.1.7, page 417).

Median filtering tends to remove impulse noise and thus primarily affects higher resolution layers. The majority of the changes are minor and thus mostly affect the less significant bits, causing more errors in higher quality layers.

Table 5.13: Total bit error rates after median2×2 filtering for resolution scaled subimages.

resolution layers	bit errors	extracted bits	BER
1	248365	572406	0.4339
2	412848	900987	0.4582
3	804701	1681466	0.4786
4	1155483	2374141	0.4867
5	1434585	2923044	0.4908
6	1516410	3080646	0.4922

Table 5.14: Total bit error rates after median2×2 filtering for quality scaled subimages.

quality layers	bit errors	extracted bits	BER
1	5262	31184	0.1687
2	16018	66124	0.2422
3	53862	157629	0.3417
4	180487	424910	0.4248
5	305897	679832	0.45
6	435913	939343	0.4641
7	538749	1144533	0.4707
8	1516410	3080646	0.4922

Midpoint

The midpoint attack replaces each pixel by the midpoint between the maximum and minimum values in a 3x3 window.

Like the majority of filters considered here, midpoint filtering is used for noise removal and thus primarily affects the high resolution layers. Similarly, images with high contrast textures and sharp edges are the most affected. The majority of changes caused are minor and thus mostly affect the higher quality layers. Thus, unsurprisingly, both the average (tables 5.15 and 5.16) and individual results (section D.4.1.7, page 417) show increasing BERs as layers are added.

Table 5.15: Total bit error rates after midpoint3×3 filtering for resolution scaled subimages.

resolution layers	bit errors	extracted bits	BER
1	244616	571248	0.4282
2	402733	895879	0.4495
3	782346	1644607	0.4757
4	1064530	2197950	0.4843
5	1183351	2427795	0.4874
6	1201298	2461203	0.4881

Table 5.16: Total bit error rates after midpoint3×3 filtering for quality scaled subimages.

quality layers	bit errors	extracted bits	BER
1	5306	36744	0.1444
2	22106	88904	0.2487
3	74278	213207	0.3484
4	197123	474665	0.4153
5	346494	774612	0.4473
6	446344	974002	0.4583
7	604827	1286439	0.4702
8	1201298	2461203	0.4881

Trimmed Mean

The trimmed mean attack removes the two highest and two lowest pixels in a 3x3 window and computes the mean of the remaining samples.

The overall error rates for the low resolution layers are similar to the baseline error rates from decompression and recompression (table 5.3, page 166). The pattern of increasing error rates with increasing resolution and quality layers is consistent with other nongeometric attacks both for average (tables 5.17 and 5.18) and individual image (section D.4.1.9, page 419) results.

The trimmed mean is a noise removal technique, eliminating high frequency data both through averaging and the removal of outliers, so the majority of errors occur at higher resolution layers. The majority of the changes are minor and thus mostly affect the less significant bits, causing more errors in higher quality layers.

Table 5.17: Total bit error rates after trimmed mean for resolution scaled subimages.

resolution layers	bit errors	extracted bits	BER
1	210888	572331	0.3685
2	355837	900386	0.3952
3	727882	1674649	0.4346
4	1055782	2310293	0.457
5	1247287	2683998	0.4647
6	1260948	2709722	0.4653

Table 5.18: Total bit error rates after trimmed mean for quality scaled subimages.

quality layers	bit errors	extracted bits	BER
1	3006	34170	0.088
2	9429	74986	0.1257
3	39306	186829	0.2104
4	147956	463542	0.3192
5	304925	799424	0.3814
6	436941	1075709	0.4062
7	603965	1410893	0.4281
8	1260948	2709722	0.4653

Downsampling

The watermarked image is downsampled to 75% of its original size using bicubic interpolation and anti-aliasing. The downsampled image is then upsampled to its original size.

Again, because the downsampling preserves more of the lower frequency data and the changes to pixels are generally confined to less significant bits, the resulting error patterns are similar to those of the other nongeometric attacks, showing an increase in error rate with the addition of resolution or quality layers and for images high contrast edges and texture (tables 5.19 and 5.20). The same behaviour occurs for individual images (section D.4.1.10, page 420).

Table 5.19: Total bit error rates after sampledownup75 for resolution scaled subimages.

resolution layers	bit errors	extracted bits	BER
1	237195	572375	0.4144
2	398318	901468	0.4419
3	787543	1688953	0.4663
4	1145798	2399304	0.4776
5	1483860	3064673	0.4842
6	1711501	3508063	0.4879

Table 5.20: Total bit error rates after sampledownup75 for quality scaled subimages.

quality layers	bit errors	extracted bits	BER
1	3846	29957	0.1284
2	11395	60711	0.1877
3	34053	129102	0.2638
4	94750	272389	0.3478
5	190442	475050	0.4009
6	265356	630779	0.4207
7	413514	933465	0.443
8	1711501	3508063	0.4879

Sharpening

The sharpening attack subtracts a 3x3 Laplacian filtered image to enhance the image contrast.

Although the sharpening attack differs from the other attacks given here in that it enhances high resolution detail rather than removing it, it still produces minor changes concentrated towards the higher resolution layers, and thus produces a similar pattern of BERs to the other nongeometric attacks (tables 5.21 and 5.22 and section D.4.1.11, page 421).

The increase in error rate with additional layers is also less clear than for other attacks because this attack is quite strong; the overall error rate quickly approaches 50%, which is the error rate expected when the candidate and extracted marks are unrelated.

Table 5.21: Total bit error rates after Laplacian3×3 filtering for resolution scaled subimages.

resolution layers	bit errors	extracted bits	BER
1	250718	567952	0.4414
2	415294	901492	0.4607
3	857845	1764102	0.4863
4	1419147	2870951	0.4943
5	2540166	5083707	0.4997
6	4147286	8247084	0.5029

Table 5.22: Total bit error rates after Laplacian3×3 filtering for quality scaled subimages.

quality layers	bit errors	extracted bits	BER
1	3042	17875	0.1702
2	8169	37291	0.2191
3	21520	73254	0.2938
4	61897	163608	0.3783
5	121925	288556	0.4225
6	196423	439075	0.4474
7	278556	603646	0.4615
8	4147286	8247084	0.5029

JPEG2000 Compression

The JPEG2000 compression attack applies JPEG2000 with a compression rate of 0.0125. Note that this attack involves full recompression of the image, and is quite different from quality scaling which simply discards some quality layers. Furthermore, the compression rate is stronger than the 0.9999 used in the recompression test (section 5.2.1.4, page 166), so we may expect higher BERs. This attack affects all resolution layers, although higher resolution layers again show higher errors (table 5.23) as compression tends to preserve more low resolution data.

There is a noticeable gap in the overall results for subimages consisting of three and four quality layers (table 5.24). This is because the first three layers correspond to compression rates 0.0025, 0.005 and 0.01, which are below 0.0125, while the remaining layers correspond to rates 0.02 and above. Thus compression with rate 0.0125 causes little change to the first three quality layers but removes the majority of the fourth and subsequent layers.

Table 5.23: Total bit error rates after JPEG2000 at rate 0.0125 for resolution scaled subimages.

resolution layers	bit errors	extracted bits	BER
1	265078	573026	0.4626
2	429556	905124	0.4746
3	805118	1658685	0.4854
4	1143533	2330156	0.4908
5	1454411	2943830	0.4941
6	1539900	3113690	0.4946

Table 5.24: Total bit error rates after JPEG2000 at rate 0.0125 for quality scaled subimages.

quality layers	bit errors	extracted bits	BER
1	4070	29434	0.1383
2	11919	60989	0.1954
3	38256	132506	0.2887
4	298652	656521	0.4549
5	664557	1376269	0.4829
6	925614	1892991	0.489
7	1143194	2323050	0.4921
8	1539900	3113690	0.4946

5.2.1.6 Geometric Attacks

The fragility of the algorithm is also tested using geometric attacks from the Checkmark [139] benchmarking suite. While in nongeometric attacks the values of pixels at corresponding locations in the watermarked and attacked image are modified, in geometric attacks the image undergoes spatial alteration, so coefficients with the same index may no longer represent the same part of the image.

The experiment described in section 5.2.1.1 (page 161) is performed repeatedly, using a different attack with each repetition. The attacks used are cropping4, linear5, projective7, rotation1, rotation45 and scale1. The Checkmark suite traditionally stores all attacked images in JPEG compressed format, this step has been eliminated in favour of recompression into JPEG2000 format, using the same compression parameters which were used during the embedding phase.

The watermark is expected to be particularly fragile to geometric attacks, because the indexing procedure, used in candidate watermark generation, depends not only on coefficient position but also on the image dimensions. Thus, even if the positions of most coefficients are unchanged, changes to the image dimensions will alter the value of the candidate watermark for the vast majority of coefficients, resulting in high bit error rates for all subimages.

Cropping

The cropping attack removes 10% of the image area, rows and columns are deleted from the bottom right and the aspect ratio is maintained as closely as possible.

The average error rates are between 45% and 50% for all resolution and quality layers (tables 5.25 and 5.26). This is because the image dimensions are used as part of the watermark generation procedure, so any reduction in the image dimensions causes many incorrect candidate watermark elements to be generated, resulting in error rates around 50% for all mismatched elements.

The individual images show similar error rates, with the exception of images 3 and 5 (see section D.4.1.13, page 423) in which some synchronisation is retained at the lowest resolution layer.

Table 5.25: Total bit error rates after 10% cropping for resolution scaled subimages.

resolution layers	bit errors	extracted bits	BER
1	252361	520554	0.4848
2	400858	820982	0.4883
3	739480	1522166	0.4858
4	1062339	2161509	0.4915
5	1390225	2792137	0.4979
6	1527023	3056635	0.4996

Table 5.26: Total bit error rates after 10% cropping for quality scaled subimages.

quality layers	bit errors	extracted bits	BER
1	11316	25039	0.4519
2	24059	53442	0.4502
3	55376	120708	0.4588
4	131634	281691	0.4673
5	221597	469467	0.472
6	354914	738270	0.4807
7	430667	894376	0.4815
8	1527023	3056635	0.4996

Linear Transformation

The linear transformation attack transforms the image coordinates using the matrix $\begin{pmatrix} 1.15 & -0.02 \\ -0.03 & 0.90 \end{pmatrix}$ and applies bicubic interpolation to determine the pixel values. The resulting non-rectangular image is surrounded by black to form a rectangular image.

The overall bit error rates are around 50% for all resolution and quality layers (tables 5.27 and 5.28). The error rates for individual images are similar (section D.4.1.14) although there is more variation, particularly when the number of extracted bits is small.

Linear transformation changes not only the size of the image but also the x, y coordinates of significant coefficients, both of which affect the watermark element generated for each significant coefficient. New significant coefficients which did not exist in the original may also be formed, either through the stretching of the image or from the addition of the black surround. This combination of effects means that the vast majority of extracted watermark elements will be compared against incorrect candidate watermark elements, resulting in the observed 50% error rates for all subimages.

Table 5.27: Total bit error rates after linear transformation for resolution scaled subimages.

resolution layers	bit errors	extracted bits	BER
1	311346	619940	0.5022
2	518044	1027435	0.5042
3	983938	1946513	0.5055
4	1461475	2887644	0.5061
5	1887231	3722433	0.507
6	2083302	4106580	0.5073

Table 5.28: Total bit error rates after linear transformation for quality scaled subimages.

quality layers	bit errors	extracted bits	BER
1	14542	28446	0.5112
2	33421	65591	0.5095
3	80576	157966	0.5101
4	202363	397499	0.5091
5	354956	699729	0.5073
6	578587	1140992	0.5071
7	701993	1384875	0.5069
8	2083302	4106580	0.5073

Projective Transformation

The projective transformation attack wraps the image onto a sphere, then a perspective projection is applied to obtain a 2d image. A 3-pixel border is cropped from the image to remove the majority of the black edges and the image is resized using nearest neighbour interpolation.

Overall bit error rates increase as more layers are added, reaching 49-50% at most resolution layers and at higher quality layers (tables 5.29 and 5.30).

Because the image dimensions are preserved in this attack, the correct watermark element will be generated for any significant coefficients whose position within the image has not changed. However, the projective transformation results in changes to both the position and value of many coefficients, particularly those at the edges of the image where the distortion is most severe (section D.4.1.15, page 426).

Table 5.29: Total bit error rates after projective transformation for resolution scaled subimages.

resolution layers	bit errors	extracted bits	BER
1	276589	570645	0.4847
2	442739	899200	0.4924
3	841311	1685462	0.4992
4	1201393	2395172	0.5016
5	1505696	2991424	0.5033
6	1613238	3200439	0.5041

Table 5.30: Total bit error rates after projective transformation for quality scaled subimages.

quality layers	bit errors	extracted bits	BER
1	12522	34166	0.3665
2	26712	65334	0.4089
3	62431	138416	0.451
4	175005	361502	0.4841
5	285061	580339	0.4912
6	446335	898858	0.4966
7	506121	1017002	0.4977
8	1613238	3200439	0.5041

Rotation

In the rotation attack, the image is rotated clockwise and then cropped to restore the image to a rectangular shape. Two repetitions of the experiment are performed, one using a one degree rotation and one using a forty five degree rotation.

The rotation alters the location and value of many coefficients, particularly those closer to the edges of the image. The cropping changes the image dimensions, causing a different set of watermark elements to be generated.

Table 5.31: Total bit error rates after 1° rotation for resolution scaled subimages.

resolution layers	bit errors	extracted bits	BER
1	275649	554448	0.4972
2	433212	866217	0.5001
3	806951	1602258	0.5036
4	1150943	2282723	0.5042
5	1477689	2924137	0.5053
6	1580339	3121108	0.5063

Table 5.32: Total bit error rates after 1° rotation for quality scaled subimages.

quality layers	bit errors	extracted bits	BER
1	14742	32236	0.4573
2	29318	61761	0.4747
3	66255	134774	0.4916
4	165974	331599	0.5005
5	261595	520245	0.5028
6	419905	831769	0.5048
7	479529	949544	0.505
8	1580339	3121108	0.5063

The combination of these effects means that even a slight rotation of one degree causes a 49-50% overall bit error rate in all images but those with one or two quality layers (tables 5.31 and 5.32), while a forty five degree rotation results in 50% overall bit error rates for all images (tables 5.33 and 5.34) including those with few quality layers. The results for individual images (section D.4.1.16, page 427) are consistent with the average results.

Table 5.33: Total bit error rates after 45° rotation for resolution scaled subimages.

resolution layers	bit errors	extracted bits	BER
1	99988	199763	0.5005
2	167579	333711	0.5022
3	272905	540529	0.5049
4	465906	918977	0.507
5	597216	1175724	0.508
6	632202	1242308	0.5089

Table 5.34: Total bit error rates after 45° rotation for quality scaled subimages.

quality layers	bit errors	extracted bits	BER
1	3381	6599	0.5124
2	8350	16652	0.5014
3	18399	36471	0.5045
4	50005	98631	0.507
5	75153	148349	0.5066
6	112534	221701	0.5076
7	149295	294171	0.5075
8	632202	1242308	0.5089

Scale

The scale attack re-scales the image, reducing each dimension to 50% of the original. An 11×11 low-pass filter is applied before rescaling to reduce aliasing effects. The overall error rates are around 50% regardless of the number of resolution or quality layers (tables 5.35 and 5.36), and BERs for individual images are similar (section D.4.1.17, page 427).

Note that this is significantly different to the 0 BER with JPEG2000 resolution scaling (section 5.2.1.2, page 162). This is because the 50% scaled image has 6 resolution layers rather than the 5 we would expect from the JPEG2000 scaled image. Coefficients in the rth resolution layer are a reasonable match to those in the $(r-1)$th resolution layer of the unmodified watermarked image; however, the change in image dimensions caused by the scale attack results in incorrect candidate elements being generated for these coefficients, and the observed 50% error rate. The coefficients in the lowest resolution layer do not correspond to coefficients in the watermarked image, again resulting in a 50% error rate.

Table 5.35: Total bit error rates after 50% (non-JPEG2000) rescaling for resolution scaled subimages.

resolution layers	bit errors	extracted bits	BER
1	76897	152939	0.5028
2	127777	253835	0.5034
3	209195	417115	0.5015
4	385071	762860	0.5048
5	510051	1007916	0.506
6	585138	1152697	0.5076

Table 5.36: Total bit error rates after 50% (non-JPEG2000) rescaling for quality scaled subimages.

quality layers	bit errors	extracted bits	BER
1	2063	3924	0.5257
2	6367	12574	0.5064
3	13130	25975	0.5055
4	32900	65384	0.5032
5	53418	106646	0.5009
6	76860	153635	0.5003
7	103619	207127	0.5003
8	585138	1152697	0.5076

5.2.1.7 The Copy Attack

The copy attack attempts to transfer the watermark from an authentic watermarked image to a different target image by: estimating the watermark from the watermarked image, estimating a perceptual mask from the target image, re-weighting the watermark as a function of the mask and adding the re-weighted watermark onto the target image. In all cases the target image is the same greyscale image from the Checkmark database, resized to fit the original image size.

The average bit error rates are approximately 50% regardless of the number of resolution and quality layers (tables 5.37 and 5.38). Individual image results are consistent with this, although there is greater variation in BERs in subimages where fewer watermark bits are extracted (section D.4.1.18, page 427).

This attack is the only protocol attack in the Checkmark suite that is specifically for fragile watermarking algorithms but it was designed for effectiveness against linear additive watermarking algorithms and so is not effective against this algorithm, which uses quantization and replacement. A mark transfer attack designed specifically to defeat this algorithm is presented in section 5.2.3.7 (page 201).

Unlike previous attacks, which represent typical image processing operations, the copy attack does not limit the distortion between the watermarked image and the output image. Thus although the copy attack is not a geometric attack (so there is no loss of synchronization) there are consistently high error rates at all levels of quality scaling, which does not occur for the other nongeometric attacks.

Table 5.37: Total bit error rates after a copy attack for resolution scaled subimages.

resolution layers	bit errors	extracted bits	BER
1	113722	227522	0.4998
2	268099	534689	0.5014
3	729611	1444849	0.505
4	1406484	2782143	0.5055
5	2451786	4837689	0.5068
6	3430610	6743351	0.5087

Table 5.38: Total bit error rates after a copy attack for quality scaled subimages.

quality layers	bit errors	extracted bits	BER
1	413	810	0.5099
2	837	1604	0.5218
3	4328	8394	0.5156
4	5041	9675	0.521
5	16093	31038	0.5185
6	16657	32090	0.5191
7	16785	32375	0.5185
8	3430610	6743351	0.5087

5.2.2 Scalability

The scalability of this algorithm is evaluated, in terms of detectability and graceful improvement, in section 5.2.2.2. This evaluation uses measures of detectability and graceful improvement that are based on the number of correctly extracted watermark bits, which are defined in section 5.2.2.1.

5.2.2.1 Scalability Measures for Exact Matching

The scalability measures in section 3.1.2 (page 54) were defined using the similarity detection statistic; however, the algorithm presented in this chapter does not use a similarity statistic for watermark detection. Instead, the watermark is considered to be detected intact if there is an exact match between the candidate and extracted watermarks.

What follows is the formulation of detectability and graceful improvement measures derived using the number of bits extracted without error. Although similar measures could be used for robust watermark detection, by either using error correction codes or allowing a bit error rate up to some threshold, these measures are discussed in the context of the semi-fragile watermarking scenario described in section 5.2 (page 158).

5.2.2.1.1 Detectability

For a semi-fragile watermarking algorithm, the objective of the detection algorithm is not to determine the presence of a given watermark, but to determine whether or not the received image is authentic.

If the output of the watermark detection algorithm is False, then a watermark mismatch has occurred and the image is not authentic.[16] However, if the output of the detection algorithm is True, then either the image is authentic, or the image is not authentic and the watermark has failed to detect the alteration.

Given the number of extracted bits, and an estimate of the watermark bit error rate for a tampered image, we can estimate the probability of obtaining a detection output of True for a tampered image. For example, if only 3 bits have been extracted, and the expected BER for a tampered image is 50%, then we would have a 12.5% chance of obtaining the output True using the tampered image.

Thus if too few bits can be extracted, then even with a detection output of True, we cannot be reasonably certain that the image has not been tampered with. That is, there is a threshold of extracted watermark bits above which an output of True should indicate an authentic image and below which the image should be considered potentially inauthentic, even if the detection output is True. The value of this threshold will be application dependent, based on the estimated bit error rate for a tampered watermark and cost of false positive decisions for various attacks.

The more watermark bits that can be extracted without error, the more certain we are that the candidate watermark remains unaltered in the image. If the number of correctly extracted watermark bits exceeds some threshold, we would conclude that the watermark has been adequately detected and that the image is authentic. Thus the number of correctly extracted watermark bits at the lowest resolution or lowest quality subimage provides a measure of watermark detectability

$$\mathcal{D} = \text{extractedbits}(I^{\mathcal{F}_0}).$$

Because threshold that the detectability measure must exceed before the watermark is deemed detectable is application dependent, the choice of threshold in this thesis is of necessity somewhat arbitrary. Ideally, such a threshold would be determined based on the expected probability of error in any extracted bit for various attacks, the expected tampered region sizes and the minimum acceptable false negative error rate for the application.

A threshold of 30 bits will be used for these tests, which corresponds to a false negative rate of roughly 1×10^{-9} assuming a 50% chance of an error in each extracted bit, [17] putting it in line with the detectability threshold used in the preceding chapter; however, raw

[16] From the properties described in section 5.1 (page 140) and the corresponding proofs in section D.3 (page 383), a mismatch between the candidate and extracted watermarks can not occur if the image is authentic, i.e. is a JPEG2000 scaled image which has not otherwise been processed.

[17] As stated in section D.4.1.1 (page 408), if there is a 50% chance of error in any extracted bit then the probability of extracting 30 bits without error is $P(X <= 0), X \sim B(30, 0.5) = 9.31 \times 10^{-10}$.

detectability results are also shown, so performance at other thresholds may be estimated if desired.

5.2.2.1.2 Graceful Improvement

The graceful improvement measure $\mathcal{G}$ is constructed by comparing the ideal amount of watermark present in a given layer l with the amount extracted from the same layer.

The ideal number of watermark bits ι^l to be embedded in layer l, is simply the number of embedded bits N times the improvement in perceptual quality iq_l provided by that layer.

$$\iota^l = \mathrm{iq}^l N \tag{5.28}$$

Because this algorithm does not have a fixed number of embedded watermark bits, the value of N, for each original image, is set to the average number of bits embedded in that image across all tested keys. The improvement in quality provided by layer l, as a proportion of the improvement provided by the full image I^{L-1} is

$$\mathrm{iq}_l = \frac{P^l - P^{l-1}}{P^{L-1} - P^e} \tag{5.29}$$

where P^l denotes the perceptual quality of subimage l relative to the original image, measured using PSNR, and subimage e denotes an empty, mid-grey image.

The number of watermark bits in layer l is simply the number of bits which could be correctly extracted from subimage l but could not be correctly extracted from subimage $l-1$

$$\epsilon^l = \mathrm{extractedbits}^l - \mathrm{extractedbits}^{l-1}. \tag{5.30}$$

How greatly the actual number of watermark elements differs from the ideal number of watermark elements across all layers is measured by

$$\Delta = \sum_l \frac{(\epsilon^l - \iota^l)^2}{\iota^l}, \tag{5.31}$$

which is normalised to range between 0 and 1 so that a value of 1 indicates a perfect fit to the ideal

$$\mathcal{G} = 1 - \frac{\Delta}{N(\frac{N}{\iota^{\mathfrak{m}}} - 1)} \tag{5.32}$$

where $\mathfrak{m}$ is the layer for which $0 < \iota^{\mathfrak{m}} \leq \iota^l \; \forall l$.

5.2.2.2 Scalability Results

Detectability

Figures 5.5 and 5.6 show the detectability results under resolution and quality scaling respectively, for each of the 20 original images and 10 secret keys. For all 200 watermarked

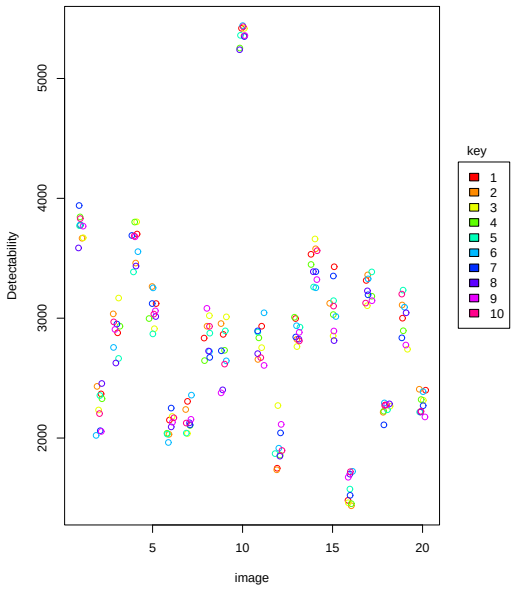


Figure 5.5: Resolution detectability: number of correctly extracted bits in the lowest resolution layer ($\frac{1}{1024}$th area).

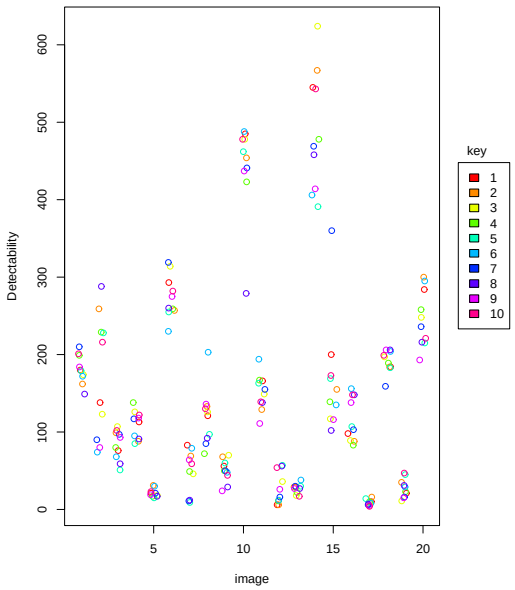


Figure 5.6: Quality detectability: number of correctly extracted bits in the lowest quality layer (rate 0.0025). Many values are below the threshold of 30 bits.

images, no fewer than 1435 bits were extracted from the lowest resolution layer ($\frac{1}{1024}$th the area of the original), well above the 30-bit threshold required for the watermark to be considered detectable. However, good detectability is not obtained at the lowest tested quality layer (compression rate 0.0025) with 20% of the watermarked images falling below

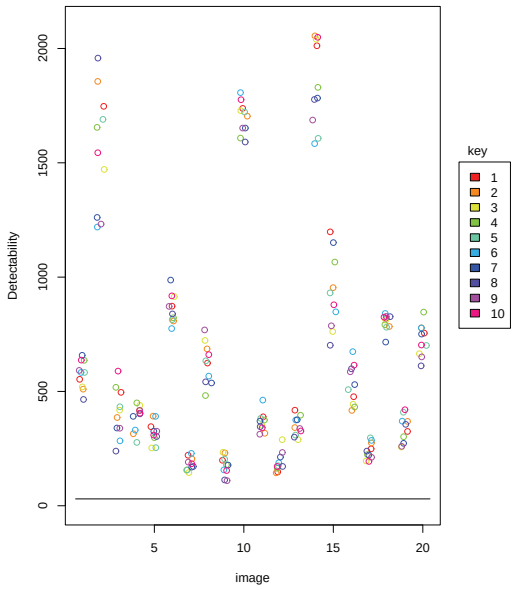


Figure 5.7: Quality detectability: number of correctly extracted bits in the third quality layer (rate 0.01). All values are above the threshold of 30 bits.

the 30 bit threshold. Indeed, the detectability threshold is not exceeded for all tested watermarked images until the third quality layer (compression rate 0.01). The detectability results obtained by defining the lowest quality layer to be at a compression rate of 0.01 are shown in figure 5.7

Based on these results, the fraction of original images for which detectability is expected to be below 30 is estimated to be 5.7×10^{-62} for resolution scaling. For quality scaling, the fraction of images with detectability below 30 is estimated to be as high as 0.1481. At a compression rate of 0.01, the estimated rate of missed detections is reduced to a more acceptable 0.000049. The details of the estimation process are discussed in section D.4.1.2 (page 409).

Graceful Improvement

For each watermarked image, a graceful improvement value is calculated using the number of correctly extracted bits at each quality layer. Recall that a graceful improvement value of 1 indicates that the proportion of bits extracted from each layer precisely matches its contribution to the perceptual quality of the image, while a value of 0 indicates a poorly allocated watermark, concentrated in the least perceptually important layer.

Figures 5.8 and 5.9 show the graceful improvement results under resolution and quality scaling respectively. For resolution scaling, graceful improvement is reasonably good, with the majority of original images resulting in values over 0.9. For quality scaling, the graceful

improvement results suffer from the same problem as the quality detectability results; too little of the watermark is embedded in the low quality layers, so the majority of values are between 0.8 and 0.9. Increasing the strength of the watermark in the lowest quality layers would thus improve both detectability and graceful improvement.

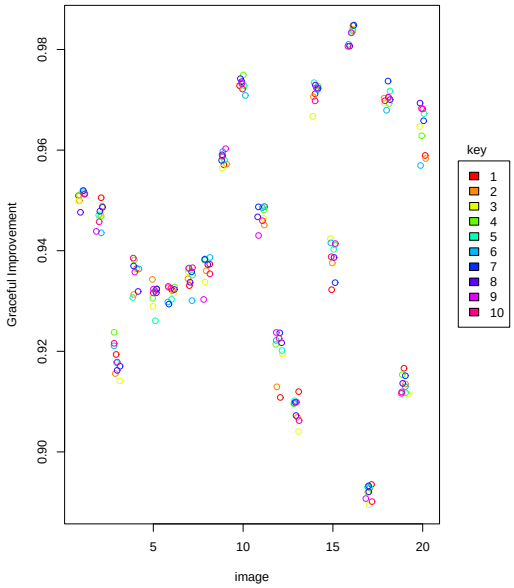


Figure 5.8: Graceful improvement for resolution scaled images.

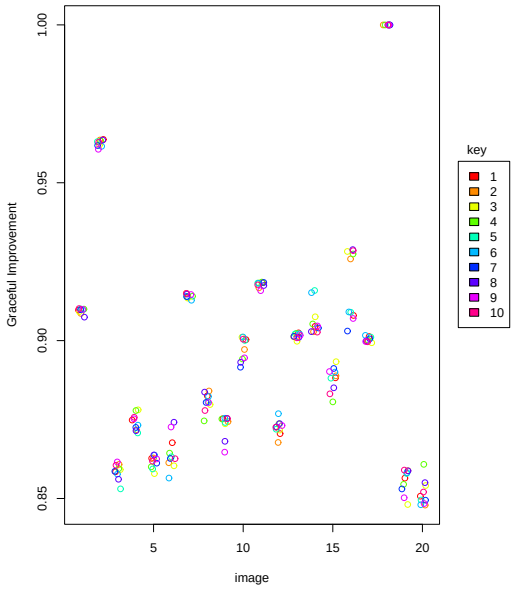


Figure 5.9: Graceful improvement for quality scaled images.

5.2.3 Tamper Detection

To evaluate the tamper detection ability of the watermarking algorithm as it stands (with no security enhancing modifications), deliberate tampering attacks are applied to a single image which has been watermarked using the algorithm described in section 5.1 (page 140). Following each attack detection is performed and a 'tamper map', showing where the watermark has been successfully extracted and where errors have occurred, is constructed.

5.2.3.1 Attack Model

Attacker's Goal

The aim of the attacker is to alter the content of the watermarked image in such a way as to fool both the human receiver and the watermark detector. A successful attack is one in which the attacker can produce a tampered image that is

- meaningfully different from the watermarked image,
- of acceptable visual quality, and
- produces no watermark detection errors.

Attacker's Capabilities

We assume that the attacker access to

- one (or more) compressed watermarked images and
- the associated compression parameters

but does *not* have access to either

- the original unwatermarked image(s),
- a watermark detector containing the secret key.

Attack Strategies

Without access to the secret key, a new valid watermark cannot be generated for the tampered image directly, so the attacker must construct a watermark for the tampered image using the watermarks from existing images. This can be achieved either by copying only the watermark, or by copying both the watermark and the associated image content. If only the watermark is copied, then it must first be located and some modifications may

be needed to match it to the tampered image. If both the watermark and image content are copied, then there must be a sufficient range of valid image content to rearrange, so that the tampered image is of acceptable quality.

Depending on the number of available images and knowledge of the watermarking algorithm, the attacker may try one of several possible options with varying degrees of success. These are listed in increasing order of sophistication.

- **Tampering in the spatial domain.** The watermarked image is decompressed and modified in the spatial domain before recompression. Since the tampering process involves recompression, if recompression can be detected, then the algorithm will also detect tampering in the spatial domain.

- **Tampering in the wavelet domain.** The tampered image is formed by decoding the watermarked image, modifying some wavelet coefficients and re-encoding. Note that an algorithm that detects tampering by collage and mark transfer attacks will also detect tampering in the wavelet domain.

- **Mark transfer.** This attack is an extension to tampering in the wavelet domain in which the attacker attempts to copy the watermark bits from the watermarked coefficients to the tampered coefficients. The attacker must know the watermarking algorithm.

- **Mark transfer with quality scaling.** The attacker applies quality scaling after the mark transfer attack, in order to remove the remaining detection errors. The amount of quality scaling is limited by that which will still produce an image of acceptable quality. However, with a sufficiently small modification and some luck, it is possible to remove all detection errors using this method.

- **Collage attack.** This attack is an extension to tampering in the wavelet domain in which the attacker replaces tampered coefficients in the watermarked image with the most similar coefficient of the same index from the other watermarked images. The attacker must know the watermarking algorithm and have access to multiple images that use the same secret key.

5.2.3.2 Experimental Framework

Embedding

Figure 5.10 shows the original image used in these experiments (Image 34, section A.37, page 334): a satellite photo of the Greek isles, with an orange satellite in the lower portion of the image, slightly to the right of centre.

Figure 5.10: The original Greek isles image.

Figure 5.11: The watermarked Greek isles image.

In these experiments the original image is watermarked using the algorithm described in section 5.1.8 (page 156) with watermarking parameters $\alpha = 0.056641$, $t = 8$ and $sk = 1$. This is done during JPEG2000 compression with the JasPer algorithm, with 6 resolution levels and 5 quality layers with compression rates 0.01, 0.02, 0.04, 0.06 and 0.9999. The watermarked image (figure 5.11) is not noticeably different from the original (figure 5.10).

Attacking

Some changes are applied to the authentic watermarked image to alter the image content in a meaningful way. The specifics of these changes depend on the attack strategy being examined and, therefore, are discussed in the appropriate subsubsection. During this stage the role of the attacker is assumed, thus the changes may make use of the compression parameters but not the secret key.

Detection

Watermark detection is performed using the same watermarking parameters as were used in the embedding stage. The output γ will either be $True$ if there are no watermark detection errors and $false$ if at least one bit of the extracted watermark differs from its corresponding candidate watermark bit.

To allow visualisation of the detection results, the detection algorithm has been modified to produce a *tamper map*. For each selected coefficient $v_i^* \in V^*$ at position (x, y) within the subband $s = (r, o)$ that contains a watermark element, we colour the $2^{R-r+1} \times 2^{R-r+1}$ block of pixels with top left hand corner $(2^{R-r+1}x, 2^{R-r+1}y)$ on the image grid. If the candidate and extracted watermark elements for the coefficient v_i^* match exactly, the green component of these pixels is set to 255. If they do not match exactly, the red component is set to 255. This produces a different, image-sized map for each subband in each component of the image, showing the correct and incorrect watermark elements as non-overlapping regions.

The maps for all subbands and components are merged to obtain a single tamper map showing combined detection results for the entire image. Regions corresponding only to correct watermark elements are shown in green, those corresponding only to incorrect watermark elements are shown in red and regions with both correct and incorrect watermark elements (from different subbands or components) will appear yellow. Black regions are those for which no watermark could be extracted.

A successful attack is one in which meaningful image alterations are present, the tampered image is of acceptable quality and the detector output is $True$ (a green, or green and black tamper map). The detector output is objective, but whether 'meaningful' image

alterations are present and what constitutes 'acceptable quality' are subjective. No attempt is made to objectively measure the presence or absence of the image alterations nor the acceptability of the image quality; instead, corresponding watermarked and tampered images are displayed and discussed.

The attacker is assumed to have access to the full watermarked image; however, watermark detection should function correctly at different levels of resolution and quality scaling. Although results for all levels of scaling are included, many of the corresponding tamper maps are not shown in this section. These tamper maps, and the scaled subimages from which they were obtained, can be found in section D.4.2 (page 432).

5.2.3.3 Embedding and Detection Without Tampering

The correct functioning of the algorithm is first demonstrated on an untampered image, by skipping the attack phase. Detection using the correct parameters on the full, untampered, watermarked image produces the correct output $\gamma = True$, with 0 errors from 38653 extracted bits, and the corresponding completely green tamper map, shown in fig figure 5.12. This indicates the successful receipt of the tampered image.

Figure 5.12: Tamper map for the watermarked Greek isles image.

All scaled, untampered subimages produce similar results, which are provided in section D.4.2.1 (page 432).

5.2.3.4 Tampering in the Spatial Domain

Since most image editors function in the spatial domain, the simplest form of attack is to decompress the watermarked image, make modifications in the spatial domain and then recompress the image using the original parameters.

Figure 5.13: Spatially tampered image.

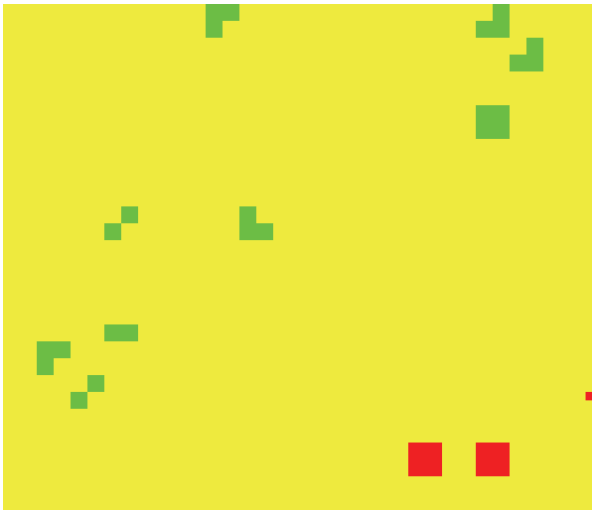


Figure 5.14: Tamper map for the spatially tampered image.

For the spatial domain attack, the watermarked image (figure 5.11) is decompressed using IrfanView [170] and pixels are altered, in the rectangular region from (330, 360) to (430, 437), to remove the satellite. The image is then recompressed, using the same compression parameters but no watermarking, to produce a spatially tampered image (figure 5.13).

The resulting image I^{τ} is meaningfully different from the watermarked image (the satellite is missing), and the careful modification in the spatial domain has not impaired the image quality. However, the attack is not successful, because watermark detection using the correct parameters produces the output $\gamma = False$ with 2806 errors from 37072 bits and the tamper map shown in figure 5.14. The spatial tampering is detected at all

levels of resolution and quality scaling (section D.4.2.2, page 440).

Although the detection errors seen in figure 5.14 are most severe in the tampered region, they extend across the majority of the image. These errors occur as a result of the decompression/recompression process, which is required for spatial tampering. This makes spatial tampering very easy to detect, although it may interfere with tamper localisation for small tampered regions. These errors can be avoided by performing the tampering in the wavelet domain.

5.2.3.5 Tampering in the Wavelet Domain

If tampering is performed in the wavelet domain, all changes are made directly to the wavelet coefficients corresponding to the tampered region of the image without disturbing any watermarked coefficients outside that region.

All coefficients required to reconstruct a rectangular region of tampered pixels (from (330, 360) to (430, 437)) are copied from the spatially tampered image (figure 5.13) into the valid watermarked image (figure 5.11), producing the wavelet tampered image shown in figure 5.15. This process only involves partial decompression and partial recompression, and thus does not cause recompression errors. The identification of which coefficients are required to reconstruct a given tampered region must be done correctly if visible artifacts are to be avoided, and is discussed in appendix D.5.

The satellite has been removed and there are no visible artifacts around the tampered region. However, the attack is not successful; watermark detection using the correct parameters produces the output $\gamma = False$. There are still 374 errors from 37110 bits, with these errors occurring in the location of the tampered coefficients (figure 5.16). The wavelet tampering is detected at all levels of scaling (section D.4.2.3, page 444)

It may be possible to further reduce detection errors by not replacing coefficients in the watermarked image if they are sufficiently similar to their tampered counterparts. Of course, if all errors were removed in this manner, the resulting image would necessarily be identical to the watermarked image and would no longer contain a meaningful change.

5.2.3.6 Holliman-Memon Counterfeiting

To eliminate the detection errors on the remaining, tampered coefficients, the attacker may be able to generate an image $I^{\tau'}$, with fake watermarked coefficients $v^{\tau'}$ that are similar in value to the tampered coefficients v^{τ} (so they will reconstruct a similar tampered region) but contain the watermark bits required to pass the detector.

The mark transfer and collage attacks which follow (sections 5.2.3.7 and 5.2.3.8), are forms of Holliman-Memon counterfeiting [72]. This method of counterfeiting exploits

Figure 5.15: Wavelet tampered image.

Figure 5.16: Tamper map for the wavelet tampered image.

block-wise independence, common to many fragile watermarking algorithms, to generate the fake coefficients and has been successfully applied to defeat several watermarking algorithms, producing tampered images that pass detection.

The algorithms considered in [72] all embed a logo value W_i into a block X_i using a key K_i. They define a block-based watermarking technique as one which partitions an image into non-overlapping blocks $\{X_1, X_2, ...X_n\}$ and inserts a watermark W_i in block X_i using the key K_i. They state that a block-based technique is *block-wise independent* if each watermarked block X_i' depends only on the original block X_i the watermark W_i and the insertion key K_i and, similarly, the detection process is applied independently to every block.

For the algorithm under evaluation (section 5.1.8, page 156), the block X_i corresponds to the coefficient v_i and the block key K_i to both the global embedding strength and local pseudorandomly generated number $\{\alpha, w_i\}$. While this algorithm does not embed a logo W, we can assume a known logo where all logo bits are $W_i = 0$; this will produce the same watermarked coefficient value, within the framework used by Holliman-Memon

$$\begin{aligned} X'_i &= \mathcal{E}_{K_i}(X_i, W_i) \\ &= \text{sign}(v_i)(Q_{2_i^j}(v_i) + (u_i \oplus W_i)) \\ &= \text{sign}(v_i)(Q_{2_i^j}(v_i) + u_i) = v'_i \end{aligned} \tag{5.33}$$

and, because, $Q_{2_i^j}(v_i)$ and u_i are both functions of $\{\alpha, w_i\} = K_i$ and $v_i = X_i$ only (the key sk is constant across a given image), the embedding algorithm is block-wise independent. Similarly, detection becomes

$$\begin{aligned} \hat{W}_i &= \mathcal{D}_{K_i}(\hat{X}'_i) \\ &= min(u_i^d \oplus u_i^c, 1) \end{aligned} \tag{5.34}$$

where the correct logo bit $\hat{W}_i = 0$ is extracted if the candidate and extracted elements u_i^c and u_i^d match exactly, and $\hat{W}_i = 1$ is extracted otherwise. Again, u_i^d and u_i^c depend only on $\{\alpha, w_i\}$ and v'_i, so the detection algorithm is block-wise independent with single-coefficient sized blocks.

The Holliman-Memon attack consists of sorting blocks into K-equivalence classes (sets of blocks which, for a given key K, have the same logo bit $\mathcal{D}_K(X_i) = \mathcal{D}_K(X_j) = W_i$ for any pair (X_i, X_j) and replacing each tampered block Y_i, with a similar fake block Y'_i that is K_i-equivalent to the corresponding watermarked block X'_i.

In order to apply such an attack, it must be possible to find coefficient values $v_i^{\tau'}$ that are similar to the tampered coefficients v_i^{τ} and are $\{\alpha, w_i\}$-equivalent to the watermarked coefficients v'_i, without knowledge of $\{\alpha, w_i\}$.

Unlike the Yeung and Mintzer [212] and the Wong [207] schemes, where the use of the same key for all blocks means that all image blocks embedded with the same logo bit belong to the same equivalence class, every block v'_i in the algorithm considered here uses a different key $\{\alpha, w_i\}$ depending on the block position i. As a result, coefficients which have been marked with the same logo bit need not belong to the same equivalence class, so it is not possible to simply generate fake coefficients by copying existing coefficients from other parts of the image.

However, it may yet be possible to generate fake coefficients of the same $\{\alpha, w_i\}$-equivalence class as v'_i. If only a single watermarked image is available, an attacker can attempt to transfer the watermark from the watermarked coefficient to the tampered

coefficient in such a way that the fake coefficient $v_i^{\tau'}$ will be in the equivalence class of v_i'. This results in the mark transfer attack of section 5.2.3.7.

If multiple images that have been watermarked with the same key are available, coefficients v_i^j and v_i^k with the same index in different images will belong to the same equivalence class. This results in the collage attack of section 5.2.3.8

5.2.3.7 Mark Transfer

Mark transfer attacks exploit the limited dependence between the original unwatermarked image and the watermark element, to transfer the watermark element to a different (tampered) unwatermarked coefficient.

From the description of the embedding algorithm (section 5.1.8, page 156) we know that the watermark element embedded in a coefficient $v_i' \in V'$ is of the form

$$u_i = \lfloor \alpha 2^{-(h+1)} \bar{v}_i w_i \rfloor. \tag{5.35}$$

The values of α, h are public, and although $w_i = \mathsf{g}(sk, i)$ depends on the secret key, which is unknown to an attacker, it does not depend on the image content.[18] The only image dependent part $\bar{v}_i$ can be easily computed by the attacker from the watermarked image, because $\bar{v}_i = \bar{v}_i'$ (eqn. 5.3d, pg. 142).

If the corresponding coefficient in the tampered image has value v_i^τ then the detector expects it to contain the watermark $\lfloor \alpha 2^{-(h+1)} \bar{v}_i^\tau w_i \rfloor$ which only differs from u_i by a factor of $\frac{\bar{v}_i^\tau}{\bar{v}_i'}$. So we can reduce the ability to detect tampering by embedding a fake watermark element

$$u_i^\tau = u_i \frac{\bar{v}_i^\tau}{\bar{v}_i'} \tag{5.36}$$

.

Now the value of u_i is unknown, but it is formed by some number of least significant bits of v_i'; because $0 \le u_i \le |v_i|$ (eqn. 5.5, pg. 143) the most significant bit of v_i' cannot be a watermark bit. For example if $v_i' = 27$ (11011) then the watermark could be $1011, 011, 11$ or 1. If we embed the largest possible fake watermark then we can be sure that all bits of u_i will be included; however, only the most significant bit of v^τ will be left untouched by the fake watermarking process, which may cause visible distortion in the tampered image.

The image in figure 5.17 shows the result of embedding the largest possible fake watermark

$$u_i^\tau = \lfloor (|v_i'| - Q_{\frac{\bar{v}_i'}{2}}(v_i')) \frac{\bar{v}_i^\tau}{\bar{v}_i'} \rfloor \tag{5.37}$$

[18] The value w_i does not depend on the image content in the sense that the value of the coefficient v_i may be changed arbitrarily without affecting the value w_i. Although w_i does depend on the dimensions of the image, the number of components and the number of resolution layers (because it is derived using the index i) even these may be changed, with the result that corresponding coefficients v_i' and v_i^τ would simply have different locations within their respective images I' and I^τ.

into every coefficient v_i^τ in the tampered region using the embedding formula

$$v_i^{\tau'} = \text{sign}(v_i^\tau)(Q_{\frac{\bar{v}'_i}{2}}(v_i^\tau) + u_i^\tau). \tag{5.38}$$

The satellite has been removed but is some visible distortion in the tampered region.

Despite a dramatic reduction in the number of mismatched watermark bits at the detector, now only 40 errors from 37110 bits, the detector output is still $\gamma = False$, with the tamper map indicating the attack location (figure 5.18). The attack is identified at all levels of scaling (section D.4.2.4, page 451).

Figure 5.17: Tampered image with transferred watermark, preserving MSB only.

Figure 5.18: Tamper map for the tampered image with transferred watermark, preserving MSB only.

The availability of multiple images watermarked with the same parameters could be exploited to increase the effectiveness of the mark transfer attack by improving the estimate of the watermark. With only one image available, the attacker is forced to assume that all bits of v'_i except the most significant bit are part of u_i. With multiple coefficients having the same index it is possible to determine a consistent estimate of $\alpha 2^{-(h)} w_i$, reducing the upper bound on j_i and allowing for an improved fake watermark u_i^τ with reduced visual distortion.

5.2.3.7.1 Mark Transfer with Quality Scaling

The errors seen in figure 5.18 are the result of tampered coefficients which are larger than their corresponding watermarked coefficient, specifically where $\bar{v}_i^\tau > \bar{v}'_i$. In these cases, there are no watermark bits remaining in the coefficient v'_i to transfer to the $\log_2 v_i^\tau - \log_2 \bar{v}'_i$ least significant bits of the faked coefficient $v_i^{\tau'}$, so every such bit has a 50% chance of producing a mismatch.

Quality scaling has the effect of removing some least significant bits, so by applying sufficiently high quality scaling to the tampered image it may be possible to remove the mismatched LSBs from a mark-transferred image. Furthermore, because the coefficient bits that cause this problem can be easily identified, it is possible (though this has not been done here) to test different levels of quality scaling to find the smallest level of scaling that will remove all such bits, without access to the watermark detector.

The level of quality scaling required may be such that the minimum acceptable quality is still too high to remove all errors. This was the case with the Greek isles image (figure D.103, page 457), where the tampered region was quite large.

However, if the tampered region is sufficiently small, this method may result in a successful attack. This is demonstrated by watermarking the Lena image (fig. 5.19) and adding a single pixel spot to her cheek using a mark transfer attack and quality scaling to remove the remaining watermark errors. Quality scaling the tampered image with transferred watermark to a compression rate of 0.02, results in a successful attack. Despite some distortion to the tampered region at this level of quality scaling, the modification is still visible (fig. 5.20) and the detector output is $\gamma = True$ with 0 errors from 4976 bits (fig. 5.22).

Other tampering methods yield results which are largely similar to those of the Greek isles image, these are shown in section D.4.3.

Figure 5.19: The original Lena image.

Figure 5.20: Tampered image with transferred watermark, preserving MSB only, compression rate 0.02.

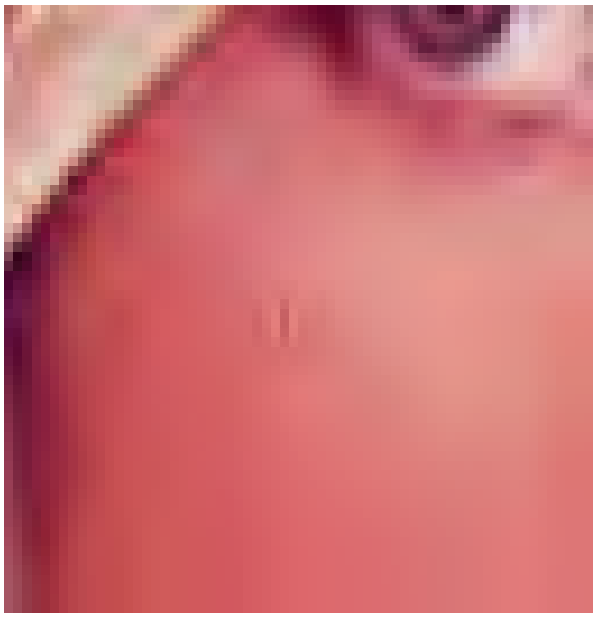


Figure 5.21: Zoomed View. A 64×64 view of the tampered region with transferred watermark, preserving MSB only, compression rate 0.02. Although scaling has blurred the modification somewhat, it is visible.

Figure 5.22: Tamper map for the tampered image with transferred watermark, preserving MSB only, compression rate 0.02.

5.2.3.8 The Collage Attack

The collage attack, proposed by Fridrich, Goljan and Memon [54], is a variation of the Holliman-Memon attack that can be used when the attacker has access to multiple images that have been watermarked with the same parameters. The fake region is created from a collage of the available watermarked images. Although the attack presented in their paper is against the Yeung and Mintzer algorithm, which is pixel based, it is easily adapted to attack this watermarking algorithm by using coefficients as the basis for the collage, rather than pixels.

Given a number of images $I^1, I^2, \ldots, I^{|\mathcal{I}|}$ all watermarked using the same parameters and secret key, the coefficients $v_i^1, v_i^2, \ldots, v_i^{|\mathcal{I}|}$, which have the same index i in their respective images will contain the same watermark (i.e. be $\{\alpha, w_i\}$-equivalent to each other). Thus to construct a fake coefficient $v_i^{\tau'}$ one can simply choose the coefficient in the set $\{v_i^1, v_i^2, \ldots, v_i^{|\mathcal{I}|}\}$ which most closely resembles the spatially-tampered coefficient v_i^τ.

$$v_i^{\tau'} = v_i^k, \quad k \in \{1, 2, \ldots, |\mathcal{I}|\} \text{ s.t. } |v_i^k - v_i^\tau| \leq |v_i^j - v_i^\tau| \; \forall j \in \{1, 2, \ldots, |\mathcal{I}|\}.$$

This technique will clearly be successful in eliminating all detection errors from the tampered image; however, it may also cause unacceptable distortion. The amount of distortion caused will depend on the number of images watermarked with the same parameters that are available to the attacker and the similarity between their coefficients and the tampered coefficients.

The image in figure 5.23 shows the result of a collage attack using the watermarked Greek isles image and five different (yet identically watermarked) images to construct the tampered region. Watermark detection produces 0 errors from 37110 extracted bits (figure 5.24). Although the tampered image passes watermark detection, with only five additional images the collage attack does not produce an image of acceptable quality; severe artifacts are clearly present in the tampered region.

The image in figure 5.25 shows a similar collage attack, with the number of additional images increased to twenty. Watermark detection produces 0 errors from 37095 extracted bits (figure 5.26). With twenty-one images, the artifacts are reduced to a slight green discolouration in the tampered region, which may well escape the notice of the observer.

Using eighty additional images undoubtedly produces a successfully tampered image (figure 5.27) that is meaningfully different from the watermarked image, is of acceptable quality and easily passes the watermark detector, with 0 errors from 37107 extracted bits (figure 5.28).

Figure 5.23: Tampered image with 6-image collage.

Figure 5.24: Tamper map for the tampered image with 6-image collage.

Note that this attack is successful on the full image. Thus, although not shown, all resolution and quality scaled versions of the collage attacked image will also be successful (provided the scaling is not so great as to remove the modification or reduce the image quality below the minimum for acceptability).

Although the collage attack is simpler if all images used have the same dimensions this is not required, provided the indices of the coefficients are correctly matched. The images used to produce the 640×480 image in figure 5.27 ranged in size from 250×250 to 700×1030.

Figure 5.25: Tampered image with 21-image collage.

Figure 5.26: Tamper map for the tampered image with 21-image collage.

Figure 5.27: Tampered image with 81-image collage.

Figure 5.28: Tamper map for the tampered image with 81-image collage.

5.3 Conclusion

Ideally, in blind watermarking, there will be an exact match between candidate and extracted watermarks if no processing has occurred. Normally, processing such as resolution or quality scaling will damage the extracted watermark, causing an inexact match or even a mismatch. For a scalable watermarking algorithm, it is desirable to maintain that exact match despite resolution and quality scaling.

Although it is not possible to reverse the effects of scaling, the algorithm described in section 5.1 is able to adapt the candidate watermark, using information available to the detector to identify which bits of the image have been lost. Thus, provided no other processing has occurred, an exact match between the candidate and extracted watermarks can still be achieved at all levels of scaling.

Testing indicates that the exact match property described in section 5.1 has been attained. Experiments on untampered watermark images have shown the match between extracted and candidate watermarks produces 0 bit errors at all tested levels of resolution and quality scaling (section 5.2.3.3).

The algorithm is resolution scalable to $\frac{1}{1024}$th the original area, and quality scalable to $\frac{1}{100}$th the original file size (section 5.2.2.2). Although the exact match property was retained for all tested levels of resolution and quality scaling, this algorithm should not be considered quality scalable beyond $\frac{1}{100}$th the original file size, as too few watermark bits could be extracted in tests of quality scaling beyond this level.

The remaining tests in section 5.2 (page 158) considered the suitability of the algorithm to an image authentication scenario, in which resolution and quality scaling were applied routinely during distribution, but images should otherwise be received unaltered. The sensitivity of the algorithm was examined with respect to an incorrect detection key, image processing and simple attacks, as well as more sophisticated deliberate attacks.

The pseudorandom number generator used in the implementation of the algorithm was inadequate, resulting in reduced sensitivity to certain changes in detection key (section 5.2.1.3). This will be remedied in the following chapter, but does not suggest a flaw in the algorithm design itself.

Although robust to resolution and quality scaling, the watermark is essentially fragile to other changes, detecting manipulations such as recompression and assorted geometric and nongeometric attacks (sections 5.2.1.4, 5.2.1.5 and 5.2.1.6). The watermark detection algorithm emphasizes this fragility, marking as inauthentic any image with even a single bit mismatch. This allows detection of even slight changes, at the cost of excluding applications in which processing other than JPEG2000 scaling is applied before authentication. Simple forms of tampering are detected by the watermarking algorithm even after severe

resolution and quality scaling, in both spatial (section 5.2.3.4) and wavelet (section 5.2.3.5) domains, even when the tampered image differs by a single pixel (appendix D.4.3).

However, the watermark is block wise independent, with single-coefficient blocks, making it vulnerable to Holliman-Memon style attacks (section 5.2.3.6), most notably the collage attack (section 5.2.3.8), and the watermark in each block is reasonably predictable from the watermarked block, making it vulnerable to mark transfer attacks (section 5.2.3.7).

Although these attacks are not universally effective—the mark transfer attack requires a high level of quality scaling, to remove the remaining watermark errors, which may render the tampered image of unacceptable quality, and the collage attack requires a large number of identically watermarked images, to produce a tampered region with acceptable quality, which may be impractical—if this watermarking algorithm is to be successfully used for image authentication, this vulnerability must be addressed.

In attempting to remove the vulnerability to Holliman-Memon style attacks, it is important that any changes to the algorithm preserve its scalability, allowing effective authentication for low resolution and low quality versions as well as for the full image.

Chapter 6

A Blind Scalable Watermark for JPEG2000: with Improved Security

Securing the algorithm of the previous chapter against watermark transfer and collage attacks requires substantial changes to the watermark generation process, along with minor changes in other areas. These changes are designed to ensure that the watermark is image dependent, in a way that cannot be easily determined by an attacker, and that blockwise independence is removed, yet also to maintain much of the scalability of the original algorithm.

Design Objectives

In order to operate effectively in the authentication scenario described in section 5.2 (page 158), a secure, blind scalable watermarking algorithm should have a number of properties.

Like the basic algorithm in chapter 5, the secured algorithm should be scalable, satisfying both detectability and graceful improvement. When the detection algorithm is applied to an untampered watermarked image there should be no mismatches between the candidate and extracted bits. Because JPEG2000 scaling is considered an acceptable modification, this property should also hold for any resolution and/or quality scaled subimage of acceptable quality.

In contrast, the watermark should be damaged by, and thus enable the detection of, other changes to the watermarked image. The approximate locations of these changes should be deducible from the locations of the mismatched watermark bits. As noted in the previous chapter, this should hold for all resolution and quality scaled subimages of acceptable quality in which the tampering is still visible.

The secured algorithm will be evaluated according to the attack model developed in the previous chapter (section 5.2.3.1, page 192). It is assumed that the attacker has access to one or more watermarked images and the associated compression parameters, but does not have access to the unwatermarked originals or a watermark detector containing the secret key. The attacker should not be able to produce a meaningful alteration to

the watermarked image that is of acceptable visual quality without causing mismatches between candidate and extracted bits.

In the secure watermark, the tamper detection properties should remain, even when sophisticated attacks are used. Most notably, the watermark transfer (section 5.2.3.7, page 201) and collage attacks (section 5.2.3.8, page 206), which successfully defeated the unsecured algorithm, should be detected by this algorithm.

6.1 Design of a Secured Algorithm

The secured watermarking algorithm consists of $\texttt{Embed}(I, sk, \Lambda)$ and $\texttt{Detect}(I^*, sk, \Lambda)$ algorithms, where I represents the (wavelet transformed) original image, I^* a potentially altered image, sk a secret key, and Λ a set of additional parameters.

As in the basic algorithm, the watermark is embedded in selected coefficients of the wavelet transformed original image I, obtained using JPEG2000 compression, immediately following the quantization step. The set of selected coefficients contains all coefficients in I with magnitudes greater than or equal to a threshold t. Each coefficient in the image I is assigned an index i which is used in the generation of any pseudorandom numbers for the coefficient v_i. In each selected coefficient v_i, a watermark element u_i is embedded.

At this point, the design deviates from the basic algorithm, which uses a pseudorandomly generated watermark element. In the improved algorithm, each watermark element is treated as a sequence of individual watermark bits. The number of watermark bits to be embedded depends on a pseudorandom number, a global strength parameter, and a local embedding strength.

Each watermark bit is constructed using a sequence of pseudorandomly selected image coefficients, termed a 'feature sequence'. The watermark bit $u_{i,\kappa}$ is constructed according to the signs and relative magnitudes of quantized versions of the coefficients belonging to the feature sequence $V_{i,\kappa}$.

The coefficient selection, indexing and local strength adjustment procedures are retained from the basic design, so that synchronization is maintained when resolution or quality scaling is applied. Thus, the number of embedded bits and the selected feature sequence locations are easily recoverable in the scaled image. Corresponding candidate and extracted watermark bits are compared during detection only if both can be correctly recovered. This allows us to ensure that for any untampered resolution or quality scaled image, there will be an exact match between all compared candidate and extracted watermark bits, thereby eliminating false alarms.

To provide detectability, some features include (and are embedded in) only the lowest resolution or quality layers. To allow graceful improvement, other features include (and

are embedded in) higher resolution or quality layers. The use of both sign and magnitude information in feature formation improves the sensitivity of the watermark to tampering, and the quantization step sizes are chosen to maximize feature sensitivity, while minimizing the effects of scaling.

Because each watermark bit depends on all coefficients in a feature sequence, tampering with an image coefficient is likely to be detected in several watermark bits. The pseudorandom selection of the feature sequence ensures that the dependencies between coefficients and and watermark bits cannot be easily determined by an attacker, preventing mark transfer attacks. Many feature sequences span multiple blocks, removing blockwise independence and defeating the basic Holliman-Memon attack. The feature sequence selection also depends on the image content, and thus the collage attack is defeated.

The following is a description of the revised design, noting what changes have been made and how they relate to the security and scalability of the algorithm.

6.1.1 Coefficient Selection and Embedding

The coefficient selection process, for both embedding and detection, is unchanged from the earlier design and retains all the advantages described in section 5.1.1 (page 140). The set of selected coefficients is formed from all quantized coefficients in the wavelet transformed image whose magnitudes meet or exceed the given threshold parameter:

$$V^* = \mathsf{Select}(I^*, \Lambda^*) \tag{5.2a}$$

$$= \{v^* \in I^* : |v^*| \geq t^* = 2^{n^*}, n^* \in \mathbb{N}\}, \tag{5.2b}$$

The watermark embedding process is also unchanged from the previous chapter. The selected coefficient $v \in V$ is quantized with step size 2^j

$$Q_{2^j}(v) = \left\lfloor \frac{|v|}{2^j} \right\rfloor 2^j \tag{5.3a}$$

and the coefficient bits removed by quantization are replaced by adding j watermark bits

$$u \in \mathbb{Z} : 2^j > u \geq 0 \tag{5.3b}$$

to produce the watermarked coefficient

$$v' = sign(v)(Q_{2^j}(v) + u), \tag{5.3c}$$

so that most significant bit of v is not overwritten and there is just enough space to embed the watermark element u. If the coefficient is not selected $v \notin V$ then no watermark is embedded and the watermarked coefficient $v' \in I'$ has the value $v' = v$.

Let $I' = \mathsf{Embed}(I, \Lambda)$ denote the watermarked image and $I^{\mathcal{Q}} = \mathcal{Q}(I')$ and $I^{\mathcal{R}} = \mathcal{R}(I')$ represent arbitrarily quality scaled and resolution scaled versions respectively. The coefficient selection and embedding rules ensure that, given a potentially scaled but otherwise

unmodified watermarked image and the correct watermarking parameters, all and only coefficients which were selected for embedding and not lost completely due to scaling are selected for detection:

- for the unscaled watermarked image, where v' is the coefficient corresponding to $v \in I$,

$$v' \in V' \iff v \in V; \tag{5.4a}$$

- for the resolution scaled image, where $v^{\mathcal{R}}$ is the coefficient corresponding to $v' \in I'$

$$v^{\mathcal{R}} \in V^{\mathcal{R}} \iff v' \in V' \wedge v^{\mathcal{R}} \in I^{\mathcal{R}}; \tag{5.4b}$$

- for the quality scaled image, where $v^{\mathcal{Q}}$ is the quality scaled coefficient corresponding to $v' \in I'$,

$$v^{\mathcal{Q}} \in V^{\mathcal{Q}} \iff v' \in V' \wedge v^{\mathcal{Q}} \neq 0; \tag{5.4c}$$

provided that, for any watermark element u embedded in a coefficient v,

$$\exists j \in \mathbb{N} \text{ s.t. } 0 \leq u < 2^j \leq |v|. \tag{5.5}$$

(This requirement is satisfied in section 6.1.3). The proof is identical to that of the previous algorithm and can be found in section D.3.1 (page 383).

6.1.2 Indexing

The coefficient indexing structure is also unchanged; it follows a raster scan of each subband of each component, starting at the lowest resolution and proceeding, in order, to the highest resolution (section 5.1.4.1, page 148).

Let $v^* \in V^*$ be a selected coefficient in an image I^* with coordinates x^* y^* in subband $s^* = (r^*, o^*) \in \{0, 1, 2, \ldots, R^* - 1\} \times \{0, 1, 2\}$ in component $c^* \in 0, 1, 2, \ldots, C^* - 1$, having dimensions X^*, Y^*. The coefficient v^* is assigned the index

$$\begin{aligned} i^* &= \mathsf{Index}(v^*, I^*) \\ &= c^* X^*[r^*] Y^*[r^*] + (C^* - c^*) X^*[r^* - 1] Y^*[r^* - 1] + \\ &\qquad \sum_{a=0}^{a=o^*-1} X^*[r^*, a] Y^*[r^*, a] + y^* X^*[r^*, o^*] + x^* \end{aligned} \tag{5.12}$$

so that, given the same watermarking parameters, the index of any received coefficient in a scaled but otherwise unmodified watermarked image image is identical to that of the corresponding coefficient in the original image.

Given the correct embedding parameters $\Lambda^* = \Lambda$, the indexing formula allows the following requirements to be satisfied:

- for the unscaled watermarked image $I' = \mathsf{Embed}(I, \Lambda)$, where $v'_{i'} \in I'$ is the coefficient corresponding to $v_i \in I$

$$i' = i; \tag{5.13a}$$

- for the resolution scaled image, where $v^{\mathcal{R}}_{i^{\mathcal{R}}} \in I^{\mathcal{R}}$ is the coefficient corresponding to $v'_{i'} \in I'$

$$i^{\mathcal{R}} = i'; \tag{5.13b}$$

- for the quality scaled image, where $v^{\mathcal{Q}}_{i^{\mathcal{Q}}} \in I^{\mathcal{Q}}$, is the quality scaled coefficient corresponding to $v'_{i'} \in I'$

$$i^{\mathcal{Q}} = i'. \tag{5.13c}$$

The proof is identical to that of the previous scheme and can be found in section section D.3.3 (page 392).

6.1.3 Watermark Element Generation

The watermark element $u^*_{i^*}$ generated for a coefficient $v^*_{i^*} \in V^*$ is no longer simply the strength-adjusted pseudorandomly generated number $G_{i^*} = \mathsf{G}(v^*_{i^*}, i^*, \Lambda^*, I^*)$ but is a concatenation of $j^*_{i^*}$ watermark bits $u^*_{i^*,\kappa^*}$

$$u^*_{i^*} = \sum_{\kappa^*=0}^{j^*_{i^*}-1} u^*_{i^*,\kappa^*} 2^{\kappa^*}, \tag{6.1}$$

where $\kappa^* \in \mathbb{Z}, 0 \leq \kappa^* < j^*_{i^*}$ and each watermark bit $u^*_{i^*,\kappa^*}$ is constructed from a pseudorandomly generated sequence of feature coefficients (section 6.1.3.1) using the signs and relative magnitudes of quantized feature coefficients (section 6.1.3.2).

Using feature coefficients in the watermark generation process, rather than pseudorandom numbers, results in dependencies between the watermark element and the feature coefficient values (section 6.1.3.2). These dependencies cause the watermark to be damaged when the coefficient values are altered, even if the watermark element itself is unchanged, thus thwarting mark transfer attacks. Careful selection of the feature coefficients (section 6.1.3.1) will ensure that the watermark will also be damaged when entire blocks of coefficients are rearranged or copied from other valid images, thwarting Holliman-Memon and collage attacks.

To ensure an exact match between candidate and extracted watermark bits, it must be possible to adequately predict the number of watermark bits that should be extracted from a scaled image. This requires the number of watermark bits that were embedded to be deducible regardless of resolution or quality scaling. As a result, the method used to determine the number of bits $j^*_{i^*} \in \mathbb{Z},\ 0 \leq j^*_{i^*} < \log_2 |v^*_{i^*}|$ to be embedded in the

coefficient $v^*_{i^*} \in V^*$ is taken largely from the previous algorithm (section 5.1.7, page 155). The only difference is that, in the improved algorithm, the Φ^* most significant bit planes of the lowest resolution layer, where $\Phi^* \in \Lambda^*$ is a given parameter, will not be modified by embedding.

Let $v^*_{i^*} \in V^*$ be a selected coefficient in an image I^* with C^* components and a lowest resolution layer with horizontal dimension $X^*[0]$ and vertical dimension $Y^*[0]$. Then the number of bits embedded (also the quantization step size exponent) is

$$j^*_{i^*} = \begin{cases} \log_2\left(\bar{G_{i^*}}\right) & \text{for } i^* \geq C^*X^*[0]Y^*[0] \\ \min\left(\log_2\left(\bar{G_{i^*}}\right), \log_2\left(\max_{x=0}^{C^*X^*[0]Y^*[0]-1} \bar{v}^*_x\right) - \Phi^*\right) & \text{otherwise} \end{cases}, \tag{6.2}$$

where $G_{i^*} = \mathsf{G}(v^*_{i^*}, i^*, \Lambda^*, I^*)$ is generated, as for the watermark element of the previous chapter, using:

$$\mathsf{G}(v^*_{i^*}, i^*, \Lambda^*, I^*) = \lfloor \alpha^* 2^{-(h+1)} \bar{v}^*_{i^*} w^*_{i^*} \rfloor, \tag{5.18}$$

where $w^*_{i^*} = \mathsf{g}(sk^*, i^*)$ (5.15), is in the range $0 \leq w^*_i < 2^h$ for some $h \in \mathbb{N}$, is pseudo-randomly generated using the key $sk^* \in \Lambda^*$ and the index i^*, where $\bar{v}^*_{i^*}$ is the smallest non-negative-integer power of two that exceeds $|v^*_{i^*}|$ (5.16) and where $\alpha^* \in \Lambda^*$ is a global strength parameter in the range $0 \leq \alpha^* < 1$. For coefficients that are not selected, no watermark bits are generated

$$j^*_{i^*} = 0 \text{ if } v^*_{i^*} \notin V^*. \tag{6.3}$$

The number of bits embedded in a coefficient v_i in the lowest resolution layer ($i < CX[0]Y[0]$) is reduced relative to that of the previous algorithm whenever this is necessary to ensure that the Φ most significant bit planes of the lowest resolution layer are not disturbed by watermark embedding. The most significant bit planes of the lowest resolution layer are likely to be present in all scaled versions of the image which are of acceptable quality, thus these bit planes can be hashed to produce an image-dependent secret key for feature sequence formation, which provides security against collage attacks. Note that all Φ bit planes must be received intact before the correct hash value can be computed. Thus the parameter Φ adjusts the tradeoff between image dependence and scalability; for larger values of Φ the hash value will be more sensitive to changes in the lowest resolution layer, while for smaller values of Φ fewer bit planes are required before watermark detection can occur.

Because the number of embedded bits for each coefficient is determined in a similar manner, many of the properties provided by the original watermark element generation process (section 5.1.4.2, page 150) are preserved. Fine adjustment of the watermark strength is still possible due to the real-valued w_i and global strength parameter α, and

the correct number of embedded bits can still be determined for any received coefficient in the scaled watermarked image. The local strength α_i is still approximately proportional to the coefficient magnitude; however, for coefficients in the lowest resolution layer, fewer watermark bits might be embedded.

The watermark generation procedure described here ensures that the number of embedded bits j_i satisfies the requirement of the proof in section D.3.1 (page 383), which is

- given the original image, where $v_i \in V$ is a selected coefficient

$$j_i \in \mathbb{N} \text{ s.t. } 0 \leq u_i < 2^{j_i} \leq |v_i| \tag{5.5}$$

and that given the correct watermarking parameters $\Lambda^* = \Lambda$ and a potentially scaled but otherwise untampered watermarked image, of which at least the Φ most significant bit planes of the lowest resolution layer have not been lost, the number of watermark bits can be correctly determined for any selected coefficient:

- for the unscaled watermarked image, where $v'_i \in I'$ is the coefficient corresponding to $v_i \in I$

$$j'_i = j_i; \tag{6.4a}$$

- for the resolution scaled watermarked image, where $v_i^{\mathcal{R}} \in I^{\mathcal{R}}$ is the coefficient corresponding to $v'_i \in I'$ (and has not been completely lost $v_i^{\mathcal{R}} \in I^{\mathcal{R}}$)

$$j_i^{\mathcal{R}} = j_i; \tag{6.4b}$$

- for the quality scaled watermarked image, where $v_i^{\mathcal{Q}} \in I^{\mathcal{Q}}$ is the coefficient corresponding to $v'_i \in I'$ and has not been completely lost $v^{\mathcal{Q}} \neq 0$

$$j_i^{\mathcal{Q}} = j_i. \tag{6.4c}$$

The proof can be found in section E.1.1 (page 487).

6.1.3.1 Feature Sequence Formation

Each bit κ^* of the watermark element $u^*_{i^*}$ is calculated from a different pseudorandomly selected feature sequence of η^* coefficients $V^*_{i^*,\kappa^*}$, where $\eta^* \in \Lambda^*$ is the given feature sequence length parameter.

$$V^*_{i^*,\kappa^*}(i^*, \kappa^*, \Lambda^*, I^*) = \{v^*_{\mathfrak{f}^*(i^*,\kappa^*,0)}, v^*_{\mathfrak{f}^*(i^*,\kappa^*,1)}, \ldots, v^*_{\mathfrak{f}^*(i^*,\kappa^*,\eta^*-1)}\}. \tag{6.5}$$

The sequence $V^*_{i^*,\kappa^*}$ is either an *intra-codeblock sequence*, in which case all feature coefficients $v^*_{\mathfrak{f}^*(i^*,\kappa^*,0)} \cdots v^*_{\mathfrak{f}^*(i^*,\kappa^*,\eta^*-1)}$ are restricted to codeblock $b^*_{i^*}$ of subband $s^*_{i^*} = (o^*_{i^*}, r^*_{i^*})$

of component $c^*_{i^*}$ or an *intra-resolution sequence*, in which case all feature coefficients are restricted to resolution r_{i^*} but may be chosen from any component or subband. The sequence type is determined by the κth bit $\Psi_{i^*,\kappa^*} \in \{0,1\}$ of the pseudorandomly generated number Ψ_{i^*}

$$\Psi_{i^*,\kappa^*}(i^*,\kappa^*,\Lambda^*,I^*) = \mathsf{g}(\mathsf{H}(sk^*,\Phi^*,0LL^*),i^*)_{\kappa^*}, \tag{6.6}$$

where $\mathsf{H}(sk^*,\Phi^*,0\mathrm{LL}^*)$ is a hash of the secret key sk^* and the first Φ^* significant bit planes of the lowest resolution layer $0\mathrm{LL}^*$ of I^* and their sign bits.[1]

Given a selected coefficient $v^*_{i^*} \in V^*$ and a bit position $\kappa^* \in \mathbb{Z}, 0 \le \kappa^* < j^*_{i^*}$, the sequence $\mathfrak{I}^*{}_{i^*,\kappa^*}$ denotes set of indices from which a feature coefficient index $\mathfrak{f}^*(i^*,\kappa^*,x^*)$ may be chosen, arranged in increasing order. When $\Psi^*_{i^*,\kappa^*} = 0$, the set consists of the indices of all coefficients in the resolution layer $r^*_{i^*}$. When $\Psi^*_{i^*,\kappa^*} = 1$, it consists of the indices of all coefficients in the codeblock $b^*_{i^*}$.

$$\mathfrak{I}^*{}_{i^*,\kappa^*}(i^*,\kappa^*,\Lambda^*,I^*) = \begin{cases} \left[C^*X^*[r^*_{i^*}-1]Y^*[r^*_{i^*}-1], C^*X^*[r^*_{i^*}]Y^*[r^*_{i^*}]\right) \\ \qquad\qquad \textbf{if } \Psi^*_{i^*,\kappa^*} = 0 \\ \bigcup_{n^*_x,n^*_y} \Big\{ c^*_{i^*}X^*[r^*_{i^*}]Y^*[r^*_{i^*}] \\ \qquad +(c^*_{i^*}-c^*_{i^*})X^*[r^*_{i^*}-1]Y^*[r^*_{i^*}-1] \\ \qquad +\sum_{a=0}^{a=o^*_{i^*}-1} X^*[r^*_{i^*},a]Y^*[r^*_{i^*},a] \\ \qquad +(n^*_y-\mathrm{tby0}^*)X[r^*_{i^*},o^*_{i^*}] \\ \qquad +n^*_x-\mathrm{tbx0}^* \Big\} \\ \qquad\qquad \textbf{if } \Psi^*_{i^*,\kappa^*} = 1 \end{cases} \tag{6.7}$$

where $n^*_x, n^*_y \in \mathbb{Z}$ such that

$$\max\left(\left\lfloor \frac{\mathrm{tbx0}^*+x^*_{i^*}}{2^{\mathrm{xcb}'^*}}\right\rfloor 2^{\mathrm{xcb}'^*}, \mathrm{tbx0}^*\right) \le n^*_x < \min\left(\left\lceil \frac{\mathrm{tbx0}^*+x^*_{i^*}}{2^{\mathrm{xcb}'^*}}\right\rceil 2^{\mathrm{xcb}'^*}, \mathrm{tbx1}^*\right) \tag{6.8a}$$

and

$$\max\left(\left\lfloor \frac{\mathrm{tby0}^*+y^*_{i^*}}{2^{\mathrm{ycb}'^*}}\right\rfloor 2^{\mathrm{ycb}'^*}, \mathrm{tby0}^*\right) \le n^*_y < \min\left(\left\lceil \frac{\mathrm{tby0}^*+y^*_{i^*}}{2^{\mathrm{ycb}'^*}}\right\rceil 2^{\mathrm{ycb}'^*}, \mathrm{tby1}^*\right) \tag{6.8b}$$

where $(x^*_{i^*}, y^*_{i^*})$ denotes the location of the sample $v^*_{i^*}$ $(\mathrm{tbx0}^*, \mathrm{tby0}^*)$ the location of the top left-hand sample, $(\mathrm{tbx1}^*-1, \mathrm{tby1}^*-1)$ the bottom right-hand sample, and xcb'^* and ycb'^* the codeblock size exponents, in the subband $s^*_{i^*} = (o^*_{i^*}, r^*_{i^*})$ and component $c^*_{i^*}$.

[1] A sign bit of 0 represents both positively signed coefficients and coefficients which are not significant in the Φ^* most significant bit planes of $0\mathrm{LL}^*$, while a sign bit of 1 represents negatively signed (significant) coefficients

Each feature coefficient $v^*_{\mathfrak{f}^*(i^*,\,\kappa^*,x^*)}$, where $x \in \mathbb{Z}, 0 \leq x^* < \eta^*$, is chosen uniformly from this sequence of allowed coefficients by generating a pseudorandom number $\mathfrak{r}_{i^*,\,\kappa^*,x^*}$, between zero and the cardinality of the set of allowed feature coefficient indices $0 \leq \mathfrak{r}_{i^*,\,\kappa^*,x^*} < |\mathfrak{I}_{i^*,\,\kappa^*}|$

$$\mathfrak{r}^*{}_{i^*,\,\kappa^*,x^*}(i^*,\kappa^*,x^*,\Lambda^*,I^*) = \mathsf{g}(\mathsf{G}(v_i,i,\Lambda,I),\mathsf{H}(sk^*,\Phi^*,0LL^*),i^*,\kappa^*,x^*) \tag{6.9}$$

and choosing the feature coefficient with index

$$\mathfrak{f}^*(i^*,\,\kappa^*,x^*) = \mathfrak{I}_{i^*,\,\kappa^*}(\mathfrak{r}_{i^*,\,\kappa^*,x^*}). \tag{6.10}$$

Because intra-resolution sequences are formed from pseudorandomly selected coefficients across the entire resolution, a watermark bit for a coefficient v_i in resolution r_i may depend on any η coefficients in r_i. This removes the blockwise independence of the watermark and thus defeats Holliman-Memon attacks.

However, we have no guarantees, at the time of embedding, that coefficients in the same resolution layer as the selected coefficient v_i will be included in a given scaled image. Thus not all of an intra-resolution sequence $V^*_{i^*,\,\kappa^*}$ will necessarily be available whenever the corresponding watermark bit is available. If too few bits are available from any feature coefficient, it will be impossible to compute the watermark bit $u^*_{i^*,\kappa^*}$.

Intra-codeblock sequences do not have this drawback as we are at least guaranteed to have received all but the κ^* least significant bit planes of all coefficients in the sequence[2] at the point where the watermark bit $u^*_{i^*,\kappa^*}$ has been received. However, intra-codeblock sequences are blockwise independent with codeblock-sized blocks, so a scheme composed entirely of intra-codeblock sequences would be vulnerable a Holliman-Memon style attack with codeblock-sized blocks. Using both intra-resolution and intra-codeblock sequences protects against Holliman-Memon style attacks while still ensuring that enough sequences are received to allow watermark bit construction from a scaled image.

Because each sequence $V_{i,\,\kappa}$ and sequence type $\Psi_{i,\kappa}$ is formed using a hash of the secret key and the Φ most significant bit planes of the lowest resolution layer, the feature sequences are unknown to the attacker and image dependent. This prevents vulnerability to collage attacks, because the watermark is dependent not only on the secret key but also the image itself, so it cannot be determined without access to a large number of images not only watermarked with the same secret key but also for which the Φ most significant bit planes of the lowest resolution layer are identical.

[2] The fractional bit-plane encoding used in JPEG2000 (section 2.2.3.6, page 47) includes all more significant bit planes in a codeblock before including any bits from a less significant bit-plane. Thus, if the watermark bit $u^*_{i^*,\kappa^*}$, which lies in bit-plane κ^*, has been received, then all bits from the most significant bit-plane to bit plane $\kappa^* + 1$ must also have been received.

Restricting all feature coefficient sequences $V_{i,\kappa}$ to the resolution r_i provides resolution scalability. Each feature sequence and its corresponding watermark bit is contained within a single resolution thus is unaffected by the loss of higher resolution layers due to scaling. Low-resolution layers can be used to authenticate low-resolution features and the addition of subsequent resolution layers allows the authentication of increasingly higher frequency features. Although this restriction does allow an entire resolution layer of a watermarked image to be replaced by the corresponding resolution layer from a different image (watermarked with the same secret key and having exactly the same Φ most significant bit planes in the lowest resolution layer) without producing a detection error, such an attack is unlikely to produce an image which is meaningfully different from the original or of acceptable visual quality and thus is not considered successful.

The feature sequence formation procedure described here ensures that given the correct watermarking parameters $\Lambda^* = \Lambda$ and a potentially scaled but otherwise untampered watermarked image I^*, of which at least the Φ most significant bit planes of the lowest resolution layer have not been lost, corresponding feature sequences are composed of corresponding coefficients:

- for the unscaled watermarked image, where $v'_i \in V'$ is the coefficient corresponding to $v_i \in V$

$$\mathfrak{f}'(i,\,\kappa,x) = \mathfrak{f}(i,\,\kappa,x) \tag{6.11a}$$

$$v'_{\mathfrak{f}'(i,\,\kappa,x)} \in V'_{i,\,\kappa} \iff v_{\mathfrak{f}(i,\,\kappa,x)} \in V_{i,\,\kappa}; \tag{6.11b}$$

- for the resolution scaled watermarked image, where $v_i^{\mathcal{R}} \in V^{\mathcal{R}}$ is the coefficient corresponding to $v'_i \in V'$

$$\mathfrak{f}^{\mathcal{R}}(i,\,\kappa,x) = \mathfrak{f}(i,\,\kappa,x) \tag{6.11c}$$

$$v^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\,\kappa,x)} \in V^{\mathcal{R}}_{i,\,\kappa} \iff v_{\mathfrak{f}(i,\,\kappa,x)} \in V_{i,\,\kappa}; \tag{6.11d}$$

- for the quality scaled watermarked image, where $v_i^{\mathcal{Q}} \in V^{\mathcal{Q}}$ is the coefficient corresponding to $v'_i \in V'$

$$\mathfrak{f}^{\mathcal{Q}}(i,\,\kappa,x) = \mathfrak{f}(i,\,\kappa,x) \tag{6.11e}$$

$$v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\,\kappa,x)} \in V^{\mathcal{Q}}_{i,\,\kappa} \iff v_{\mathfrak{f}(i,\,\kappa,x)} \in V_{i,\,\kappa}. \tag{6.11f}$$

The proof can be found in section section E.1.2 (page 495).

6.1.3.2 Watermark Bit Construction

Each watermark bit $u^*_{i^*,\,\kappa^*}$ is constructed by taking the exclusive or of the signs and relative magnitudes of adaptively quantized feature coefficients in the sequence $V^*_{i^*,\,\kappa^*}$

$$u^*_{i^*,\,\kappa^*} = \bigoplus_{x^*=0}^{\eta^*-1} \mathfrak{S}(v^*_{\mathfrak{f}^*(i^*,\,\kappa^*,\,x^*)}) \oplus \mathfrak{M}(v^*_{\mathfrak{f}^*(i^*,\,\kappa^*,\,x^*-1 \mod \eta^*)}, v^*_{\mathfrak{f}^*(i^*,\,\kappa^*,\,x^*)}) \quad (6.12a)$$

where

$$\mathfrak{S}(v^*_{\mathfrak{f}^*(i^*,\,\kappa^*,\,x^*)}) = \begin{cases} 0 & \text{if } \operatorname{sign}(v^*_{\mathfrak{f}^*(i^*,\,\kappa^*,\,x^*)}) Q_{2^{q^*_{i^*,\,\kappa^*,\,x^*}}}(v^*_{\mathfrak{f}^*(i^*,\,\kappa^*,\,x^*)})) \geq 0 \\ 1 & \text{if } \operatorname{sign}(v^*_{\mathfrak{f}^*(i^*,\,\kappa^*,\,x^*)}) Q_{2^{q^*_{i^*,\,\kappa^*,\,x^*}}}(v^*_{\mathfrak{f}^*(i^*,\,\kappa^*,\,x^*)})) < 0 \end{cases} \quad (6.12b)$$

represents the sign information, and

$$\mathfrak{M}(v^*_{\mathfrak{f}^*(i^*,\,\kappa^*,\,t^*)}, v^*_{\mathfrak{f}^*(i^*,\,\kappa^*,\,x^*)}) = \begin{cases} 0 & \text{if } Q_{2^{q^*_{i^*,\,\kappa^*,\,t^*}}}(v^*_{\mathfrak{f}^*(i^*,\,\kappa^*,\,t^*)}) \leq Q_{2^{q^*_{i^*,\,\kappa^*,\,x^*}}}(v^*_{\mathfrak{f}^*(i^*,\,\kappa^*,\,x^*)}) \\ 1 & \text{if } Q_{2^{q^*_{i^*,\,\kappa^*,\,t^*}}}(v^*_{\mathfrak{f}^*(i^*,\,\kappa^*,\,t^*)}) > Q_{2^{q^*_{i^*,\,\kappa^*,\,x^*}}}(v^*_{\mathfrak{f}^*(i^*,\,\kappa^*,\,x^*)}) \end{cases} \quad (6.12c)$$

the magnitude information, from the feature coefficients quantized with their respective step size exponents $q_{i^*,\,\kappa^*,x^*}$,$q_{i^*,\,\kappa^*,t^*}$, $0 \leq x^*, t^* < \eta^*$.

The sign and magnitude features are designed to produce a single bit that is sensitive to tampering. The sign feature is sensitive to a change in the signs of the feature coefficients and will change in value if an odd number of quantized coefficients in the sequence have a different sign. The magnitude feature is sensitive to changes in the magnitudes of the quantized feature coefficients; it establishes a pair of inequalities for each coefficient in the feature sequence and will change in value if an odd number of these inequalities are violated.

This defeats mark transfer attack (section 5.2.3.7, page 201). This is because each watermark bit has been derived from image features, so when the image is changed many watermark bits should also change, and each watermark bit is derived from a secret sequence of coefficients, so it is difficult for an attacker to determine how the watermark might be altered to fit the new image.

To tamper directly with a single coefficient v_i undetected the attacker must determine which watermark bits will be changed by the tampering. However, as was noted earlier, watermark bits are constructed from a sequence containing v_i is derived from a hash of the secret key and the Φ most significant bits of the lowest resolution layer, so an attacker does not know which bits are sensitive to changes in v_i, and this information difficult to derive without access to multiple images with identical hash results.

The feature quantization step size $2^{q^*_{i^*,\kappa^*,x^*}}$ is adjusted for each feature coefficient $v^*_{\mathfrak{f}^*(i^*,\kappa^*,x^*)}$ in the sequence $V^*_{i^*,\kappa^*}$

$$q^*_{i^*,\kappa^*,x^*} = \begin{cases} \max(M^*_{s^*_{\mathfrak{f}^*(i^*,\kappa^*,x^*)}} - (1 + \lfloor \frac{j^*_{i^*}-1-\kappa^*}{a^*} \rfloor), j^*_{\mathfrak{f}^*(i^*,\kappa^*,x^*)}) & \text{if } \Psi_{i^*,\kappa^*} = 0 \\ \max(\kappa^*+1, j^*_{\mathfrak{f}^*(i^*,\kappa^*,x^*)}) & \text{if } \Psi_{i^*,\kappa^*} = 1 \end{cases} \tag{6.13}$$

it depends on the bit-plane κ^* in which the resulting feature will be embedded, the number of embedded bits $j^*_{\mathfrak{f}^*(i^*,\kappa^*,x^*)}$, the maximum number of significant bit planes $M_{s_{\mathfrak{f}^*(i^*,\kappa^*,x^*)}}$ for the feature coefficient's subband, the sequence type Ψ_{i^*,κ^*}, and a feature robustness parameter $a^* \in \Lambda^*$.

The quantization step size provides a tradeoff between feature sensitivity and quality scalability. A smaller step size captures more detail for the feature while a larger step size allows more quality scaling.

The intra-codeblock quantization step size is chosen to be as sensitive as possible while ensuring that all bits required to compute the watermark bit will be present as soon as the watermark bit is received. It exploits the feature of JPEG2000 codestreams that if any bit from a given bit plane in a codeblock has been received, then all more significant bit planes in that codeblock have been received (section 2.2.3.6, page 47). If the watermark bit $u^*_{i^*,\kappa^*}$, contained in the bit plane κ^* of codeblock b_{i^*}, exists then all bit planes greater than κ^* are present, so all unwatermarked bits in these bit planes are included in the feature.

Intra-resolution sequences use a large quantization step size, enabling watermark bit computation with only a few received bits from each coefficient, because there is no guarantee of the number of received bits for each feature coefficient. The constant feature robustness parameter a allows control over tradeoff between quality scalability and sensitivity; a large a value will result in coarse features which should be available at most levels of quality scaling while a small a value will produce increasingly sensitive features, which may not be computable until higher quality layers, as κ decreases.

For both sequence types the quantization step size for each coefficient is set large enough to remove all $j^*_{\mathfrak{f}^*(i^*,\kappa^*,x^*)}$ embedded bits, so that embedding in a coefficient that has already been included in a feature sequence has no effect on the candidate watermark bit computed using that sequence.

Given the correct watermarking parameters $\Lambda^* = \Lambda$ and a potentially scaled but otherwise untampered watermarked image, of which at least the Φ most significant bit planes of the lowest resolution layer have not been lost, quantization step size exponents for corresponding feature coefficients are identical provided the feature coefficient has not been completely lost due to scaling.

- For the unscaled watermarked image, where $v'_i \in V'$ is the coefficient corresponding to $v_i \in V$

$$\forall v'_{\mathfrak{f}'(i,\kappa,x)} \in V'_{i,\kappa}, q'_{i,\kappa,x} = q_{i,\kappa,x}. \tag{6.14a}$$

- For the resolution scaled watermarked image, where $v_i^{\mathcal{R}} \in V^{\mathcal{R}}$ is the coefficient corresponding to $v'_i \in V'$

$$\forall v^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\kappa,x)} \in V^{\mathcal{R}}_{i,\kappa}, q^{\mathcal{R}}_{i,\kappa,x} = q_{i,\kappa,x}. \tag{6.14b}$$

- For the quality scaled watermarked image, where $v_i^{\mathcal{Q}} \in V^{\mathcal{Q}}$ is the coefficient corresponding to $v'_i \in V'$

$$\forall v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)} \neq 0 \in V^{\mathcal{Q}}_{i,\kappa}, q^{\mathcal{Q}}_{i,\kappa,x} = q_{i,\kappa,x}. \tag{6.14c}$$

The proof can be found in section section E.1.3 (page 508).

6.1.4 Calculating the Number of Missing Bits

As was the case with the previous algorithm, the candidate generation and watermark extraction procedures (sections 6.1.5 and 6.1.6) rely on the correct calculation of the number $m^*_{i^*}$ of least significant magnitude bits which were lost during scaling to produce the scaled coefficient $v^*_i \in V^*$ from the coefficient $v'_i \in V'$ where $I^* = I^{\mathcal{F}}$ is an untampered scaled image.

The number of missing bits is still calculated from the maximum number of bit planes M_{s_i} in the subband s_i of resolution r_i in component c_i, the number Z_{b_i} of most significant bit planes in codeblock b_i that are taken to be all zero, and the number P_{b_i} of received fractional bit plane passes for the codeblock b_i.

However, now we must correctly calculate the number of missing bits not only for all selected coefficients $v^*_i \in V^*$, which, due to the coefficient selection procedure, are necessarily non-zero, but also for feature coefficients $v^*_i \in V^*_{h^*,\kappa^*}$, which may well have the value zero.

The general procedure is identical to that of section 5.1.6 (page 153) but we can no longer ignore the case where $P_{b_i} \equiv 0 \pmod 3$ and $v^*_i = 0$, for which the number of received magnitude bits $X_i = 1 + \left\lfloor \frac{P_{b_i}-1}{3} \right\rfloor$ or, equivalently, for which $X_i = 1 + \left\lfloor \frac{P_{b_i}}{3} \right\rfloor - 1$. Thus

$$X_i = 1 + \left\lfloor \frac{P_{b_i}}{3} \right\rfloor + \begin{cases} -1 & \text{if } P_{b_i} \equiv 0 \pmod 3 \text{ and } v^*_i = 0 \\ 1 & \text{if } P_{b_i} \equiv 2 \pmod 3 \text{ and } \bar{v}^*_i = 2^{(M_{s_i} - Z_{b_i} - 1 - \lfloor \frac{P_{b_i}}{3} \rfloor)} \\ 0 & \text{otherwise} \end{cases}. \tag{6.15a}$$

So the number of missing magnitude bits is

$$m_{i^*}^* = M_{s_{i^*}^*} - Z_{b_{i^*}^*} - 1 - \left\lfloor \frac{P_{b_{i^*}^*}}{3} \right\rfloor - \begin{cases} -1 & \text{if } P_{b_i} \equiv 0 \pmod 3 \text{ and } v_i^* = 0 \\ 1 & \text{if } P_{b_i} \equiv 2 \pmod 3 \text{ and } \bar{v}_i^* = 2^{(M_{s_i} - Z_{b_i} - 1 - \lfloor \frac{P_{b_i}}{3} \rfloor)} \\ 0 & \text{otherwise} \end{cases} \tag{6.15b}$$

Given a scaled, but otherwise untampered, watermarked image $I^{\mathcal{F}}$, the marker segment and packet header information accurately represents the scaling of v_i' to produce $v_{i^{\mathcal{F}}}^{\mathcal{F}}$. Thus the number of least significant magnitude bits $m_{i^{\mathcal{F}}}^{\mathcal{F}}$, missing from the representation of the coefficient $v_{i^{\mathcal{F}}}^{\mathcal{F}}$, will be identical to the number of bits that were lost from the watermarked coefficient v_i' to produce $v_{i^{\mathcal{F}}}^{\mathcal{F}}$, that is $m_{i^{\mathcal{F}}}^{\mathcal{F}} = m_i$.

Note that, because $v_i^* \in I^*$ can be zero, it is possible that the codeblock containing v_i^* has not been received and thus that the number Z_{b_i} of most significant bit planes in the codeblock b_i that are taken to be all zero, which is contained in the first packet header for b_i, is not explicitly given. In this case, no most significant bit planes are taken to be all zero, so $Z_{b_i} = 0$ and the number of received passes is zero $P_{b_i} = 0$ thus, using the above formula, we correctly obtain the result that all (potentially significant) bits of v_i^* have not been received $m_i^* = M_{s_i} = m_i$.

6.1.5 Candidate Generation

Because the generation process for u_{i^*,κ^*} uses all but the q_{i^*,κ^*,x^*} least significant bits of each feature coefficient $v_{i^*,\kappa^*,x^*}^* \in V_{i^*,\kappa^*}^*$, the corresponding candidate watermark bit u_{i^*,κ^*}^c can only be correctly constructed from the scaled image if none of those bits are missing. That is if, for all feature coefficients $v_{i^*,\kappa^*,x^*}^* \in V_{i^*,\kappa^*}^*$, the number of missing bits does not exceed the quantization step size exponent: $m_{\mathfrak{f}^*(i^*,\kappa^*,x^*)}^* \leq q_{i^*,\kappa^*,x^*}^*$.

Thus, candidate generation uses the generation process described in section 6.1.3 (page 217), but a candidate bit is only generated provided none of the required feature bits have been lost due to scaling

$$u_{i^*,\kappa^*}^c = \begin{cases} u_{i^*,\kappa^*}^* & \text{if } \exists v_{\mathfrak{f}^*(i^*,\kappa^*,x^*)}^* \wedge m_{\mathfrak{f}^*(i^*,\kappa^*,x^*)}^* \leq q_{i^*,\kappa^*,x^*}^*, \forall x^* \in \mathbb{Z} : 0 \leq x^* < \eta^* \\ \nexists & \text{if } \exists x^* \in \mathbb{Z}, 0 \leq x^* < \eta^*, \nexists v_{\mathfrak{f}^*(i^*,\kappa^*,x^*)}^* \vee m_{\mathfrak{f}^*(i^*,\kappa^*,x^*)}^* > q_{i^*,\kappa^*,x^*}^* \end{cases} \tag{6.16}$$

where $v_{i^*}^* \in V^*$ is a selected coefficient, $\kappa^* \in \mathbb{Z}, 0 \leq \kappa^* < j_{i^*}^*$, and $\nexists$ denotes that no candidate watermark bit is generated.

Given the correct watermarking parameters $\Lambda^* = \Lambda$ and a potentially scaled but otherwise untampered watermarked image, of which at least the Φ most significant bit

planes of the lowest resolution layer have not been lost, then if a candidate watermark bit exists it will be identical to the corresponding embedded watermark bit:

- for the unscaled watermarked image, where $v'_i \in V'$ is the coefficient corresponding to $v_i \in V$

$$\text{if } \exists u^c_{i,\kappa} \text{ then } u^c_{i,\kappa} = u_{i,\kappa}; \tag{6.17a}$$

- for the resolution scaled watermarked image, where $v^{\mathcal{R}}_i \in V^{\mathcal{R}}$ is the coefficient corresponding to $v'_i \in V'$

$$\text{if } \exists u^c_{i,\kappa} \text{ then } u^c_{i,\kappa} = u_{i,\kappa}; \tag{6.17b}$$

- for the quality scaled watermarked image, where $v^{\mathcal{Q}}_i \in V^{\mathcal{Q}}$ is the coefficient corresponding to $v'_i \in V'$

$$\text{if } \exists u^c_{i,\kappa} \text{ then } u^c_{i,\kappa} = u_{i,\kappa}. \tag{6.17c}$$

The proof can be found in section E.1.4 (page 511).

6.1.6 Watermark Extraction

The watermark extraction process is essentially the same as in the previous design (section 5.1.3, page 144); however, in this design we are more interested in individual bits

$$u^d_{i^*,\kappa^*} = \begin{cases} \left\lfloor \frac{|v^*_{i^*}| - Q_{2^{(\kappa^*+1)}}(v^*_{i^*}) - \left\lfloor 2^{m^*_{i^*}} \mathrm{r} \right\rfloor}{2^{\kappa^*}} \right\rfloor & \text{if } m^*_{i^*} \le \kappa^* < j^*_{i^*} \\ \nexists & \text{if } \kappa^* < m^*_{i^*} \text{ or } \kappa \ge j^*_{i^*} \end{cases} \tag{6.18}$$

where $v^*_{i^*} \in V^*$ is a selected coefficient, $\kappa^* \in \mathbb{Z}, 0 \le \kappa^* < j^*_{i^*}$, r is the coefficient reconstruction parameter and the symbol $\nexists$ is used to indicate that no value is assigned to $u^d_{i^*,\kappa^*}$ because the associated watermark bit is not extracted.

Given the correct watermarking parameters $\Lambda^* = \Lambda$ and a potentially scaled but otherwise untampered watermarked image, of which at least the Φ most significant bit planes of the lowest resolution layer have not been lost, then if an extracted bit exists, it will be identical to the corresponding embedded bit:

- for the unscaled watermarked image, where $v'_i \in V'$ is the coefficient corresponding to $v_i \in V$

$$\text{if } \exists u^d_{i,\kappa} \text{ then } u^d_{i,\kappa} = u_{i,\kappa}; \tag{6.19a}$$

- for the resolution scaled watermarked image, where $v^{\mathcal{R}}_i \in V^{\mathcal{R}}$ is the coefficient corresponding to $v'_i \in V'$

$$\text{if } \exists u^d_{i,\kappa} \text{ then } u^d_{i,\kappa} = u_{i,\kappa}; \tag{6.19b}$$

- for the quality scaled watermarked image, where $v_i^{\mathcal{Q}} \in V^{\mathcal{Q}}$ is the coefficient corresponding to $v'_i \in V'$

$$\text{if } \exists u^d_{i,\kappa} \text{ then } u^d_{i,\kappa} = u_{i,\kappa}. \tag{6.19c}$$

The proof can be found in section E.1.5 (page 520).

6.1.7 Detection Output

Provided that $u^d_{i,\kappa}$ can be extracted and $u^c_{i,\kappa}$ can be generated, i.e. they both exist, they are compared. If every pair of corresponding candidate and extracted watermark bits $u^c_{i\kappa}$ and $u^d_{i\kappa}$ that is compared produces a match, we may conclude that the image has not been modified. If one or more pairs do not match then we conclude that the image has been tampered with

$$\text{Output} = \begin{cases} \text{True} & \text{if } \forall i \text{ s.t. } \exists u^c_i, u^d_i, \quad u^c_i = u^d_i \\ \text{False} & \text{if } \exists u^c_i, u^d_i, \quad u^c \neq u^d \end{cases}. \tag{6.20}$$

Given a scaled but otherwise untampered image $I^{\mathcal{F}}$, of which at least the Φ most significant bit planes of the lowest resolution layer have not been lost, and the correct watermarking parameters $\Lambda^{\mathcal{F}} = \Lambda$ then if the extracted bit $u^d_{i^{\mathcal{F}},\kappa^{\mathcal{F}}}$ exists, it will be identical to the corresponding embedded bit $u_{i^{\mathcal{F}},\kappa}$.

- For the unscaled watermarked image, where $v'_i \in V'$ is the coefficient corresponding to $v_i \in V$

$$\text{if } \forall i \text{ s.t. } \exists u^d_{i,\kappa} \wedge \exists u^c_{i,\kappa} \text{ then } u^d_{i,\kappa} = u^c_{i,\kappa}. \tag{6.21a}$$

- For the resolution scaled watermarked image, where $v_i^{\mathcal{R}} \in V^{\mathcal{R}}$ is the coefficient corresponding to $v'_i \in V'$

$$\text{if } \forall i \text{ s.t. } \exists u^d_{i,\kappa} \wedge \exists u^c_{i,\kappa} \text{ then } u^d_{i,\kappa} = u^c_{i,\kappa}. \tag{6.21b}$$

- For the quality scaled watermarked image, where $v_i^{\mathcal{Q}} \in V^{\mathcal{Q}}$ is the coefficient corresponding to $v'_i \in V'$

$$\text{if } \forall i \text{ s.t. } \exists u^d_{i,\kappa} \wedge \exists u^c_{i,\kappa} \text{ then } u^d_{i,\kappa} = u^c_{i,\kappa}. \tag{6.21c}$$

The proofs can be found in section E.1.6 (page 525).

6.1.8 Blind Scalable Watermarking Algorithm with Improved Security

The embedding algorithm:

Embed
Input: $I, \Lambda = \{\alpha, t, sk, \eta, \Phi, a\}$ **Output:** I'
◦ *Coefficient Selection* $V = \{v \in I : \lvert v\rvert \geq t = 2^n\} \qquad n \in \mathbb{N}$ ◦ *Indexing* $i = cX[r]Y[r] + (C-c)X[r-1]Y[r-1] + \sum_{o=0}^{o=s-1} X[r,o]Y[r,o] + yX[r,s] + x$ ◦ *Embedded Bits Calculation* $j_i = \begin{cases} 0 & \text{if } v_i \notin V, \text{ or} \\ \log_2\left(\bar{G}_i\right)\big) & \text{for } i \geq CX[0]Y[0] \\ \min\left(\log_2\left(\bar{G}_i\right), \log_2\left(\max_{x=0}^{CX[0]Y[0]-1} \bar{v}_x\right) - \Phi\right) & \text{for } i < CX[0]Y[0] \end{cases}$ $G_i = \lfloor \alpha 2^{(l+1)} \bar{v}_i \mathrm{g}(sk, i) \rfloor \qquad 0 \leq \alpha < 1, \quad 0 \leq \mathrm{g}(i, sk) < 2^l, \quad l \in \mathbb{Z}$ ◦ *Feature Sequence Formation* $V_{i,\kappa}(i, \kappa, \Lambda, I) = \{v_{\mathrm{f}(i,\kappa,0)}, v_{\mathrm{f}(i,\kappa,1)}, \ldots, v_{\mathrm{f}(i,\kappa,\eta-1)}\}$ where coefficients $v_{\mathrm{f}(i,\kappa,x)}$ are chosen pseudorandomly using the hash $H(sk, \Phi, 0LL)$ from the same resolution layer (if $\Psi_{i,\kappa} = 0$) or codeblock (if $\Psi_{i,\kappa} = 1$) as v_i where $\Psi_{i,\kappa}(i, \kappa, \Lambda, I) = \mathrm{g}(\mathrm{H}(sk, \Phi, 0LL), i)_\kappa$ ◦ *Watermark Bit Construction* $u_{i,\kappa} = \bigoplus_{x=0}^{\eta-1} \mathfrak{S}(v_{\mathrm{f}(i,\kappa,x)}) \bigoplus \mathfrak{M}(v_{\mathrm{f}(i,\kappa,x-1 \mod \eta)}, v_{\mathrm{f}(i,\kappa,x)})$ where $\mathfrak{S}(v_{\mathrm{f}(i,\kappa,x)})$ represents the sign and $\mathfrak{M}(v_{\mathrm{f}(i,\kappa,(x-1 \mod \eta))}, v_{\mathrm{f}(i,\kappa,x)})$ the relative magnitudes of the coefficients after quantization using step size $2^{q_{i,\kappa,x}}$ ◦ *Watermark Element Generation* $u_i = \sum_{\kappa=0}^{j_i-1} u_{i,\kappa} 2^\kappa$ ◦ *Watermark Embedding* $v_i' = \mathrm{sign}(v_i)(Q_{2^{j_i}}(v_i) + u_i)$ ◦ $I'(c, r, s, x, y) = \begin{cases} v_i' & v_i \in V \\ v_i & v_i \notin V \end{cases}$

and detection algorithm:

Detect

Input: $I^*, \Lambda^* = \{\alpha^*, t^*, sk^*, \eta^*, \Phi^*, a^*\}$

Output: True / False

◦ *Coefficient Selection* $V^* = \{v \in I^* : |v| \geq t^* = 2^{n^*}\} \qquad n^* \in \mathbb{N}$

◦ *Indexing* $i^* = c^* X^*[r^*]Y^*[r^*] + (C^* - c^*)X^*[r^* - 1]Y^*[r^* - 1] + \sum_{o=0}^{o=s^*-1} X^*[r^*, o]Y^*[r^*, o] + y^* X^*[r^*, s^*] + x^*$

◦ *Embedded Bits Calculation*

$$j_{i^*}^* = \begin{cases} 0 & \text{if } v_{i^*}^* \notin V^*, \text{ or} \\ \log_2\left(\bar{G_{i^*}}\right) & \text{for } i^* \geq C^* X^*[0]Y^*[0] \\ \min\left(\log_2\left(\bar{G_{i^*}}\right), \log_2\left(\max_{x=0}^{C^*X^*[0]Y^*[0]-1} \bar{v}_x^*\right) - \Phi^*\right) & \text{for } i^* < C^* X^*[0]Y^*[0] \end{cases}$$

$$G_{i^*} = \lfloor \alpha^* 2^{(l+1)} \bar{v}_{i^*}^* \mathsf{g}(sk^*, i^*) \rfloor \qquad 0 \leq \alpha^* < 1, \quad 0 \leq \mathsf{g}(i^*, sk^*) < 2^{l^*}, \quad l^* \in \mathbb{Z}$$

◦ *Feature Sequence Formation*

$V_{i^*,\kappa^*}^*(i^*, \kappa^*, \Lambda^*, I^*) = \{v_{\mathfrak{f}^*(i^*,\kappa^*,0)}^*, v_{\mathfrak{f}^*(i^*,\kappa^*,1)}^*, \ldots, v_{\mathfrak{f}^*(i^*,\kappa^*,\eta^*-1)}^*\}$

where coefficients $v_{\mathfrak{f}^*(i^*,\kappa^*,x^*)}^*$ are chosen pseudorandomly using the hash $H(sk^*, \Phi^*, 0LL^*)$ from the same resolution layer (if $\Psi_{i,\kappa} = 0$) or codeblock (if $\Psi_{i,\kappa} = 1$) as v_i, where $\Psi_{i^*,\kappa^*}(i^*, \kappa^*, \Lambda^*, I^*) = \mathsf{g}(\mathsf{H}(sk^*, \Phi^*, 0LL^*), i^*)_{\kappa^*}$

◦ *Watermark Bit Construction*

$u_{i^*,\kappa^*}^* = \bigoplus_{x^*=0}^{\eta^*-1} \mathfrak{S}(v_{\mathfrak{f}^*(i^*,\kappa^*,x^*)}^*) \bigoplus \mathfrak{M}(v_{\mathfrak{f}^*(i^*,\kappa^*,x^*-1 \mod \eta^*)}^*, v_{\mathfrak{f}^*(i^*,\kappa^*,x^*)}^*)$

where $\mathfrak{S}(v_{\mathfrak{f}^*(i^*,\kappa^*,x^*)}^*)$ represents the sign and $\mathfrak{M}(v_{\mathfrak{f}^*(i^*,\kappa^*,x^*-1 \mod \eta^*)}^*, v_{\mathfrak{f}^*(i^*,\kappa^*,x^*)}^*)$ the relative magnitudes of the coefficients after quantization, using step size $2^{q_{i^*,\kappa^*,x^*}^*}$

◦ *Watermark Element Generation* $u_{i^*}^* = \sum_{\kappa^*=0}^{j_{i^*}^*-1} u_{i^*,\kappa^*}^* 2^{\kappa^*}$

◦ *Candidate Truncation*

$$u_{i^*,\kappa^*}^c = \begin{cases} u_{i^*,\kappa^*}^* & \text{if } \exists v_{\mathfrak{f}^*(i^*,\kappa^*,x^*)}^* \wedge m_{\mathfrak{f}^*(i^*,\kappa^*,x^*)}^* \leq q_{i^*,\kappa^*,x^*}^*, \ \forall x^* \in \mathbb{Z} : 0 \leq x^* < \eta^* \\ \nexists & \text{if } \exists x^* \in \mathbb{Z}, 0 \leq x^* < \eta^*, \text{ s.t. } \nexists v_{\mathfrak{f}^*(i^*,\kappa^*,x^*)}^* \vee m_{\mathfrak{f}^*(i^*,\kappa^*,x^*)}^* > q_{i^*,\kappa^*,x^*}^* \end{cases}$$

◦ *Missing Bits Calculation*

$$m_{i^*}^* = M_{s_{i^*}^*}^* - Z_{b_{i^*}^*}^* - 1 - \left\lfloor \frac{P_{b_{i^*}^*}^*}{3} \right\rfloor - \begin{cases} 1 & \text{if } P_{b_{i^*}^*}^* \equiv 2 \pmod 3 \text{ and } \bar{v}_{i^*}^* = 2^{(M_{s_{i^*}^*}^* - Z_{b_{i^*}^*}^* - 1 - \lfloor \frac{P_{b_{i^*}^*}^*}{3} \rfloor)} \\ -1 & \text{if } P_{b_{i^*}^*}^* \equiv 0 \pmod 3 \text{ and } v_{i^*}^* = 0 \\ 0 & \text{otherwise} \end{cases}$$

◦ *Watermark Extraction*

$$u_{i^*,\kappa^*}^d = \begin{cases} \left\lfloor \frac{|v_{i^*}^*| - Q_{2^{\kappa^*+1}}(v_{i^*}^*) - \left\lfloor 2^{m_{i^*}^*} \mathrm{r} \right\rfloor}{2^{\kappa^*}} \right\rfloor & \text{if } m_{i^*}^* \leq \kappa^* < j_{i^*}^* \\ \nexists & \text{if } \kappa^* < m_{i^*}^* \text{ or } \kappa \geq j_{i^*}^* \end{cases}$$

◦ *Output* $= \begin{cases} \text{True} & \text{if } \forall i \text{ s.t. } \exists u_i^c, u_i^d, \quad u_i^c = u_i^d \\ \text{False} & \text{if } \exists u_i^c, u_i^d, \quad u^c \neq u^d \end{cases}$

6.1.9 Design Outcomes

The secured watermarking algorithm allows JPEG2000 resolution and quality scaling of the watermarked image, provided the Φ most significant bit planes of the lowest resolution layer remain undisturbed. For such images, provided no other modifications are applied, the exact match between candidate and extracted watermarks is retained.

The watermark is sensitive to other changes to the image. Any meaningful change, in which no attempt is made to defeat the watermark, is likely to alter either the values of the extracted watermark bits in the tampered region, or of the candidate watermark bits computed using quantized feature coefficients[3] from the tampered region, or both. All such alterations can be recorded as an error at the location of the tampered coefficient, allowing a degree of tamper location.

The design provides security against mark transfer, Holliman-Memon and collage attacks.

Security against simple mark transfer attacks is provided by a watermark generation procedure that is image dependent (section 6.1.3). More sophisticated mark transfer attacks, in which the attacker modifies the watermark bits to match the features of the tampered image, will also be defeated, because the pseudorandom numbers used in feature sequence selection are unknown to the attacker (section 6.1.3.1). Finally, it is difficult for the attacker to derive the feature sequences from the observation of watermarked images, because the number of watermark bits is chosen according to the secret key, and the feature sequences are chosen according to the hash of both the secret key and the Φ most significant bit planes of the lowest resolution layer. So the attacker would require many images for which both the secret key and the hash results were identical.

Security against collage attacks is provided by ensuring that the watermark is neither blockwise independent nor image independent. Roughly half of the watermark bits are derived from intra-resolution feature sequences, which are chosen from within an entire resolution layer. Thus any copying from another valid watermarked image of a block of size less than an entire resolution layer[4] is likely to produce errors. These errors would occur both for watermark bits inside the block with feature coefficients outside the block, and for watermark bits outside the block with feature coefficients inside the block. More errors

[3] Note that intra-codeblock feature coefficients are more sensitive to manipulations of this kind, as they use a quantization step size that is as small as possible while still ensuring that the candidate watermark bit can be correctly generated whenever the corresponding watermark bit can be extracted, so these sequences and the embedded watermark bits will frequently also detect more minor changes.

[4] Note that it is possible to substitute a whole resolution layer from a valid watermarked image (provided the hash is identical to that for the watermarked image). We do not consider this to be a weakness; however, if detection of collage attacks using entire resolution layers was required, the algorithm could be modified to also include all preceding resolution layers, maintaining resolution scalability, or all resolution layers, with some loss of resolution scalability.

are likely to occur if the block does not correspond to an integer number of codeblocks, in which case not only intra-resolution but also intra-codeblock feature sequences will be affected. The same interdependence ensures that other Holliman-Memon style attacks will also be detected.[5]

The required security against deliberate tampering has been added while maintaining most of the scalability; however, some tradeoffs have been necessary to achieve this. The major sacrifice has been to the tamper location ability of the algorithm. A mismatch between candidate and extracted watermark bits no longer necessarily indicates that the watermarked coefficient has been altered. Instead, such a mismatch could be the result of changes to the watermarked coefficient or any of the associated η feature coefficients, and as such, all $\eta + 1$ coefficients must be marked as potential errors. This drawback is mitigated somewhat by the fact that deliberate tampering generally affects many coefficients in a concentrated spatial region, so the concentration of potential errors can be used to distinguish tampered areas from the more randomly spread potential errors at untampered coefficients. The ability to apply arbitrary amounts of quality scaling has also been sacrificed, as the algorithm requires the Φ most significant bit-planes of the lowest resolution layer to be correct. This is unlikely to be a problem, however, as the most significant bit-planes of the lowest resolution layer typically contribute substantially to image quality and can be reasonably expected to be present in any image of acceptable quality.

6.2 Related Work

Although there does exist a digital signature based authentication scheme by Peng et al. [137] to protect images at all levels of scaling, this solution involves separate transmission of the signature data and is not watermark based. There appear to be no watermarking algorithms for authentication that are both resolution and quality scalable, by the definition in section 3.1.1 (page 52), while providing security against mark transfer and collage attacks.

With one exception [174], image watermarking algorithms specifically focused on scalability [201, 178, 25, 114, 163] have been for copyright protection, and thus are unsuitable for image authentication. Although Seo and Park [163] refer to authentication in their paper, this is ownership authentication, which uses robust watermarks for copyright protection, and not image (integrity) authentication.

[5] This includes block transplantation attacks, as discussed by Barreto et al. [10], because the set of "neighbouring blocks" (the feature sequence) from which the watermark bit is computed are unknown to the attacker so the expressions involving them cannot be calculated.

The authentication algorithm of Steinder et al. [174], specifically designed for resolution scalability with the SPIHT compression algorithm [155], embeds an edge map, constructed from the LL subband, into the LL subband. This allows detection immediately upon receipt of the lowest resolution layer, of size $\frac{1}{64}$th the original image area, satisfying the detectability property. However, only the lowest resolution image is used for authentication, regardless of the resolution of the received image. In reference to the effects of this for highly scaled images, they remark that "although the LL band will be verified as valid, the receiving user may be shown a tampered image".

Many semi-fragile watermarking algorithms have been designed to allow compression, and, as noted by Meerwald and Uhl [123], any algorithm robust to compression will support some degree of quality scalability; however, the vast majority of these compression tolerant algorithms provide detectability only and do not protect higher layers. For a scalable watermarking scheme, the graceful improvement property is necessary to ensure that the enhancement layers, which comprise the majority of the image, are also protected.

Although they did not specifically consider scalability, Lin et al. [111] produce a watermark that modifies both the bits that will not be lost during compression and those that will, and so does provide both quality detectability and graceful improvement. However, the watermark is blockwise independent and image independent, which allows both collage and mark transfer attacks.

Eggers and Girod [43] embed using a quantization step size that is not matched to JPEG quantization tables, and so may provide some quality graceful improvement as well as detectability, but they note that JPEG compression can remove the watermark in smooth regions, causing false alarms. Again, the watermark is blockwise and image independent, allowing collage and mark transfer attacks.

There do exist compression tolerant algorithms that provide both similar and alternative solutions to securing their watermark against Holliman-Memon style attacks, which has been the foremost concern in the design of this algorithm. So the remainder of this section focuses on these solutions and their strengths and weaknesses in a scalable watermarking context.

6.2.1 Fragile Watermarking

Much of the development towards better solutions to Holliman-Memon attacks occurred in the fragile watermarking domain. These methods are typically based on hashing each block and are too fragile for use with scalable compression, as they require all image bits to be present before the hash can be verified. While it may be possible to construct a scalable hash-based watermark by using, for example, hashes of the more significant bit planes of coefficients from codeblocks in each wavelet subband, it would still be difficult

to both invisibly embed the watermark and ensure that sufficient bits of the appropriate hashes would be present in the scaled image to allow authentication.

Two different solution techniques are used in fragile watermarking, the inclusion of additional blocks and the use of an image index. The feature selection process used in this algorithm (section 6.1.3.1, page 219) shares some similarities with both methods. The use of intra-resolution coefficients extends the feature sequences beyond the individual codeblock, while the pseudoranom feature coefficient selection process makes use of an image dependent, hash based key.

6.2.1.1 Inclusion of Additional Blocks

Holliman and Memon, when presenting their attack, suggest that it may be prevented by making the watermark to be embedded in any given block dependent upon both that block and at least one other image block, thereby removing the blockwise independence of the watermark. This increases security at the cost of decreased tamper localization. The same concept is present some form in many solutions, including our own, where the feature sequence contains η coefficient-sized blocks.

Coppersmith et al. [28] suggest hashing a larger surrounding area but embedding this into the smaller contained block. This removes blockwise independence at the cost of decreased tamper-localization.

Celik et al. [193] generate hierarchical partitions of the image with each partition at 1/4 the granularity of the previous partition. Each block at each level of the hierarchy is hashed and encrypted and the resulting signature is divided amongst the lowest-level blocks used to form it. The concatenation of all partial signatures assigned to each lowest-level block is embedded in that block. This removes blockwise independence with a graceful loss of tamper localization. Simple changes to the image will be localised at the finest granularity, while counterfeit blocks of one granularity will be detected at a coarser granularity. This has some commonality with the division of our watermark between intra-codeblock features, which are sensitive to simple changes, and intra-resolution features, which have less sensitivity but allow detection of counterfeit codeblocks.

Baretto et al. [10] state that simply making the hashes depend on overlapping regions is not sufficient. They show for a simple case, where the hash for any given block depends on that and the preceding block, a block 'transplantation' attack whereby if two different images have blocks X_a, X_b, X_c, X_d and Y_a, Y_b, Y_c, Y_d respectively, whose features only differ at position b then blocks X_b and X_c and their signatures can be replaced, by Y_b and Y_c and their signatures, without affecting the authentication result. They state that such attacks are also effective even against algorithms with several dependencies per block and suggest

that a nondeterministic signature be included in the hash to prevent transplantation (and the similar but more complex 'improved birthday') attacks.

Li and Si [103] use DWT domain blocks, sharing elements of both the Coppersmith et al. and Celik et al. techniques. In selected coefficients in the HH band at the highest resolution layer, they calculate their own single-bit hash of that coefficient with its 8-connected neighbours and the corresponding coefficient neighbourhoods in the other subbands at the highest resolution, the coefficient's ancestors in all lower resolution layers and their corresponding coefficients at all other orientations and an image sized secret key. Experiments suggest security against Holliman-Memon attacks and block transplantation attacks, the security of the hash against mark transfer attacks is not demonstrated.

6.2.1.2 Image Indexing

Wong and Memon [208] propose hashing each block with the image dimensions and the image and block indices using a cryptographic hash function and embedding the result into the least significant bits of the block. Because it does not require the inclusion of additional blocks, this technique does not involve any decrease in tamper localization, but it does require that the image index is known at the detector, which may not be practical.

Fridrich [52] solves this problem by separating the image index from the hash function. For each 128-pixel block a logo is constructed which contains the image index and block position in bits 1-52 and the same information repeated in bits 53-104, with other information such as the image dimensions, camera ID and author in the final 24 bits. The hash of each block is XORed with the corresponding logo, then encrypted and embedded in the least significant bit plane. This allows bits 1-104 of the hash to be verified using the symmetry of the logo alone, while block position and image index can be read from an untampered block.

Wang and Chen [202] use a similar approach. They tile a hash of the image index and timestamp across the image in the 3 least significant bit planes of 2x2 colour pixel blocks, so that the image index and timestamp need not be known at the detector but can be verified by symmetry. Sample values in each block and their parity bits are XORed with the hash to produce the authentication data for that block, recovery data for tampered blocks is also added. This algorithm defeats the collage attack but is vulnerable to mark transfer attacks.

6.2.2 Semi-Fragile Watermarking

The inclusion of aditional blocks is the main technique used for semi-fragile watermarking algorithms designed specifically to survive compression. The solutions used in these algorithms are, as one might expect, quite similar to the solution proposed in this design.

6.2.2.1 Inclusion of Additional Blocks

Sun et al. [181] propose a semi-fragile algorithm which, like our own algorithm, is designed for use with JPEG2000. It uses output from the EBCOT algorithm to obtain a robust feature code for each block, which is error correction coded and the parity bits are embedded in the block. The codes are also concatenated, hashed and encrypted to form a global signature which is used to check the authenticity of the image. This algorithm does not satisfy the definition of scalability in this thesis, however, because the features are chosen according to the quality layers present at a lowest authenticable bit rate (LABR) of $\frac{1}{96}$th the original file size. Because all chosen features are required for global authentication, quality layers above this rate are unprotected while subimages scaled below this rate cannot be authenticated. The global signature secures this algorithm against Holliman-Memon style attacks; however, the individual block watermarks are unsecured, which allows an attacker to recompute the correct block watermarks after tampering, removing all tamper location capability. Sun and Chang [180] present a similar digital signature based algorithm, in which watermarking is only used to support error correction coding for improved feature robustness. Their algorithm also uses a least authenticable bit rate and so does not satisfy the graceful improvement property.

Ho and Li [70] use a secret partitioning of Coppersmith et al. blocks and an image sized secret logo to prevent mark transfer, collage, and transplantation attacks; however, transplantation attacks may still be possible between identically positioned blocks in different images. From a JPEG quality 50 image, they select the four highest frequency non-zero coefficients in each 8x8 DCT block to watermark; some of the remaining coefficients from that and neighbouring blocks are chosen according to the secret logo and are partitioned into four sequences, each associated with one selected coefficient. Feature codes are constructed based on the relative signs (same or different) and magnitudes (smaller or larger) of coefficients in each sequence and the parity of the XORed sign and magnitude features is embedded in the corresponding selected coefficient. This is similar to the watermark construction method in section 6.1.3.2 (page 223) of this thesis, with two differences:

- this algorithm uses the sign of each coefficient, which is sensitive to any change in the sign of a single coefficient, their algorithm uses relative signs between each coefficient and the next coefficient, which will only be sensitive to sign changes for coefficients at the ends of the feature sequence;
- most importantly, this algorithm provides graceful improvement: features are constructed using an adaptive quantization step size, which allows tamper sensitivity to increase for higher quality images. Ho and Li use quantization at a fixed quality, meaning that all layers above JPEG quality 50 are unprotected.

Lin and Chang [107] calculate single-bit features, the sign of the difference between secretly selected corresponding coefficient pairs in different JPEG blocks, that are invariant under recompression at a higher quality (assumed to be greater than JPEG quality 50). They quantize secretly selected coefficients to robustly embed the feature bits. Because the coefficient sequences used for feature generation and embedding are unknown to the attacker, Holliman-Memon, block transplantation and mark transfer attacks are all resisted. However, because coefficient absolute differences below some threshold are accepted as authentic without reference to the calculated features, manipulations like object removal in images with large flat backgrounds may go undetected.

Maeno et al. [116] improve the Lin and Chang technique [107] by limiting the feature independent acceptance region, using two different methods. The first method uses a secret random bias sequence to move the feature independent acceptance region to a different location for each coefficient pair, making it hard to find a change to an object that will move all coefficients into their acceptance regions. The second method categorizes the coefficient difference into more than two regions, resulting in multi-bit features (which are then compressed); this allows acceptance to always be feature dependent, while still allowing a margin of overlap between adjacent acceptance regions. Quality detectability is achieved for JPEG2000 compression, down to $\frac{1}{96}$th the original file size. As with the Lin and Chang algorithm, these algorithms will resist Holliman-Memon, block transplantation and mark transfer attacks.

Like the Lin and Chang and Maeno et al. algorithms, the algorithm presented in this chapter uses secret selection of feature sequences to defeat Holliman-Memon, transplantation and mark transfer attacks. However, it additionally prevents collage attacks by using the hash of the secret key and the Φ most significant bit planes of the lowest resolution layer. Furthermore, each sequence is chosen from a single resolution layer, which provides resolution detectability not present in the Lin and Chang or Maeno et al. algorithms. Finally, their algorithms do not satisfy the graceful improvement property; they use a single (JPEG or JPEG2000, respectively) quality setting for feature generation, so all layers above that quality setting are unprotected.

The fragile 'watermark' in Fan and Tsao's "Dual pyramid watermarking scheme for JPEG-2000" [45] consists of all coefficients in each HH subband. As all HH coefficients are transmitted in full, there is no loss of localization accuracy, and the amount of error detected at different layers is used to characterize the severety of the attacks. Yet security against more sophisticated attacks appears to have been achieved by not embedding this data anywhere within the image but, instead, transmitting it separately. Thus the fragile 'watermark' is perhaps best termed a digital signature, of somewhat less than one third the size of the image.

Schlauweg et al. [161] secure against sophisticated Holliman-Memon attacks such as the collage attack by hashing all blocks in each resolution layer but, in doing so, sacrifice all localization of the tampered region. Although their focus is strongly on security, rather than scalability per se, they, like Steinder et al., recognize the vulnerabilities of algorithms that protect the base resolution layer only. As a result, they provide a truly resolution scalable (down to at least $\frac{1}{64}$th the original image area) semi-fragile algorithm by hashing together the prequantized coefficients in each resolution layer separately. Prequantization allows the watermark to survive modifications other than resolution scaling, including JPEG and JPEG2000 compression, Gaussian blur and contrast changes. However, the prequantization step size for each block is one of two known values, and can be determined from the textural properties of the block. Thus a step size sufficiently strong to allow detectability at low quality layers is also likely to allow tampering with higher bit planes[6].

Somewhat surprisingly, given their observations on security, Schlauweg and Müller return to an approach that watermarks the base resolution layer only. In their more recent paper [160], they present a fragile watermark that has resolution detectability to $\frac{1}{64}$th the original image area but no graceful improvement.

Like the Schlauweg et al. algorithm in [161], the algorithm presented in this chapter is secure against collage attacks, and considers all resolution layers, treating each separately, to provide resolution scalability. However, it is substantially different in other respects. Rather than treating each resolution as a whole and using a public key signature scheme, each resolution is divided into many feature sequences using a secret key, this provides tamper localization at the cost of a substantially restricted set of application scenarios. Finally, the intra-resolution quantization step size used in this algorithm resists tampering with higher bit planes. The Schlauweg et al. algorithm is intended for scenarios in which the standard distribution path includes operations beyond resolution and quality scaling, as a result they use less sensitive and adaptive feature quantization, which allows the modification of higher bit planes.

6.2.2.2 Secret Transform Domain

There is also a block-based semi-fragile algorithm which does not rely on the inclusion of additional blocks. Wu et al. [209] use a secret wavelet parameterization for the image transform in their semi-fragile algorithm. They embed a logo XORed with pseudorandom noise generated using a secret key into the LL subband of the 3-level transformed image by setting the fifth least significant bit of each coefficient. This allows resolution detectability

[6]Specifically, the attack on higher quality layers (figure 6.1c, page 241) could easily be adapted utilize the prequantization step size for each coefficient, as provided by the Schlauweg et al. algorithm, to perform similar wavelet domain modifications targeted precisely to the tolerances allowed by prequantization.

at $\frac{1}{64}$th the area of the original image and at JPEG quality 40. Although the watermark is blockwise independent with coefficient blocks, Holliman-Memon, transplantation and mark transfer attacks are prevented because the secret wavelet transform prevents access to the blocks. This method has the advantage that tamper localization is not sacrificed; however, the authors note that mild compression can mask tampering, allowing the image to pass detection.

6.3 Scalability in Image Authentication

The semi-fragile watermarks discussed in the preceding section were designed specifically to tolerate compression, indeed two were designed for compatability with JPEG2000 compression, yet only the resolution scalable algorithm of Schlauweg et al. [161] provides any graceful improvement.

This might lead us to reconsider our definition of a scalable watermarking algorithm. However, for a semi-fragile watermarking algorithm, even more than for a robust watermarking algorithm, it is important that the enhancement layers are protected in addition to the base layer, to allow tampering with the enhancement layers to be detected. Ideally, the watermark should be divided amongst the layers according to their value in terms of the overall image. That is, a scalable semi-fragile watermarking algorithm should possess not only detectability, but also graceful improvement.

Consider the two scalability properties defined in section 3.1.1:

1. **Detectability**

   The watermark is detectable in any portion of the scaled content which is of acceptable quality;

2. **Graceful Improvement**

   Increased portions of the scaled content provide reduced error in watermark detection, appropriate to the improved content quality.

All of the secure, semi-fragile watermarking algorithms described in the previous section provide the detectability property only. That is, they achieve compression tolerance by watermarking only the lowest resolution or (more usually) quality layer. The graceful improvement property is not satisfied, because the enhancement layers do not contribute to the watermark. Thus, although the watermark will perform correctly in an untampered resolution or quality scaled image, an attacker may tamper freely with all the higher layers of the image.

Such an approach is perfectly acceptable when only mild compression need be tolerated. In such cases, the lowest layer is set to have high perceptual quality, and the higher layers

contribute relatively little to the image. Thus, an attacker cannot make a meaningful change to the image while tampering only with the higher layers.

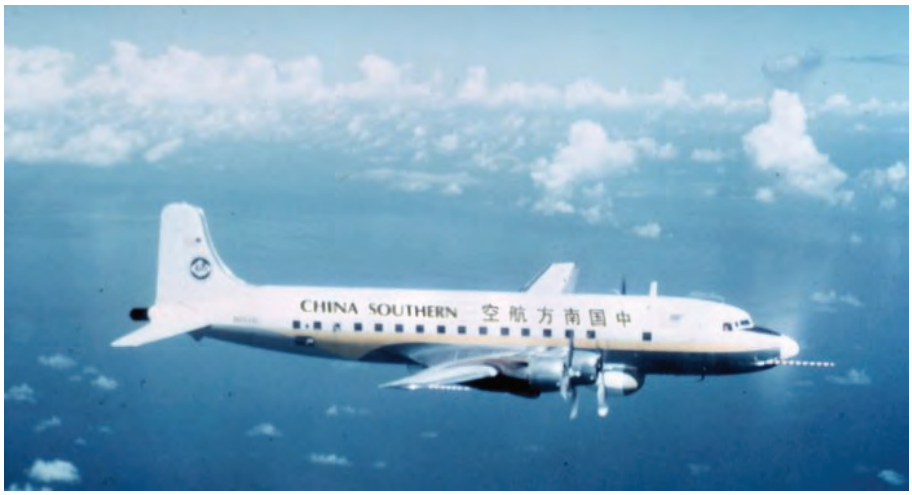
For a scalably compressed image, though, substantially higher levels of compression are permitted. Thus the lowest resolution or quality layer is likely to represent only a very small portion of the total image, while the higher layers contribute a substantial portion. In this case, an attacker may make a meaningful change to an image, which violates its authenticity, by tampering only with the enhancement layers, and leaving the lowest layer untouched.

Figure 6.1a shows an authentic image (Image 14, section A.17, page 325), compressed with 6 resolution layers and 6 quality layers with rates 0.01, 0.02, 0.03, 0.04, 0.05 and 0.9999. The text on the aeroplane reads "UNITED STATES DEPT. OF COMMERCE", and the sky in the lower right-hand corner contains only clouds. By tampering only with the higher resolution layers (leaving the lowest resolution layer untouched) we are able to produce figure 6.1b. The text on the aeroplane has been changed to read "CHINA SOUTHERN". By tampering only with the higher quality layers (leaving the lowest quality layer untouched) we are able to produce figure 6.1c. The sky in the lower right-hand corner now bears the copyright notice "©J. Citizen".

In both cases, the image quality remains acceptable and the changes could reasonably be considered 'meaningful'. The changes were performed easily by tampering with the enhancement layers in the wavelet domain, and would remain undetected by any watermarking algorithm in which the enhancement layers did not contribute to the watermark.

(a)

Figure 6.1: (a) An original image (continued pg. 241).



(b)

(c)

Figure 6.1: (a) An original image. (b) the aeroplane's text is modified to read CHINA SOUTHERN(b) without changing the lowest resolution layer. (c) a copyright mark reading © J. Citizen has been inserted without changing the lowest quality layer. All watermarking algorithms that only authenticate the lowest resolution or quality layer are vulnerable to these attacks.

6.3.1 A Note on Fragility to Modifications Other than Scaling

Focusing on scalability alone results in an authentication watermarking algorithm that lies on a somewhat unusual point in the continuum between bit-sensitive fragile algorithms, which do not allow compression, and image processing insensitive semi-fragile algorithms, which allow compression and a wide range of other modifications. This algorithm is designed to allow high levels of JPEG2000 resolution and quality scaling, but remain sensitive to other modifications. Thus, while technically semi-fragile, in that it allows quite substantial modifications to the image, some may prefer to consider it as a fragile watermark with scalability.

The practice of developing a semi-fragile algorithm in which the allowable modifications consist only of the effects of a particular compression algorithm is not unique (e.g. [108]), and has come under some criticism. Specifically, Fei et al. [48] and [49] state that focusing on tolerance to one particular distortion, limits the portability of the algorithm as tolerance may be required for other distortions. This is, of course, quite true; however, compression is undeniably the most practically important distortion that must be tolerated, and one can easily envisage situations in which JPEG2000 images may be useful without having first been brightened, cropped, blurred and rotated,[7] or, failing that, in which watermark embedding is applied to the processed image immediately prior to uploading and the image is not intended to be modified further.

Furthermore, while portability is certainly desirable, attempts to provide robustness to a range of non-malicious distortions typically involve widening a (threshold- or step size-based) tolerance in the authentication procedure, to the point where it accepts the many allowable modifications. While there is nothing intrinsically wrong with such methods, if the allowable modifications include high levels of resolution and quality scaling then this tolerence becomes such that malicious modifications, such as those described in the previous section, are also tolerated. Indeed it is not even certain that it is possible to permit such a broad set of legitimate modifications while still excluding subtle but malicious changes. This is particularly the case in a scalable compression setting, in which the levels of distortion tolerated (of necessity) by some users are beyond what would be considered acceptable by other users.

[7] Note that some of these operations, which would presumably all be tolerated by a truly portable semi-fragile algorithm, would in fact be considered malicious in certain application scenarios (see, for example, the discussion on blurring in Lin and Hsieh [106]). This is particularly the case with cropping, which is routine in certain distribution scenarios yet is forbidden in others due to the loss of potentially relevant information.

Note also that in the absence of other processing, the purpose of compression is to reduce the image file size, for better transmission or storage performance. This functionality is well provided by resolution and quality scaling, which do not involve any aditional quality degredation such as may result from recompressing an already compressed image, and thus recompression need not be tolerated.

This said, there are undoubtedly a range of other applications in which both scalability and tolerance of processing other than scaling are equally important. It is hoped (see section 7.2.4, page 274) that the methods used to provide scalability in this algorithm, and the attacks that were demonstrated in the previous section, may inform future development of semi-fragile algorithms suitable for such applications.

6.4 Evaluation of the Secured Watermarking Algorithm

The newly designed algorithm should be secure against mark transfer and collage attacks, while still maintaining scalability and fragility.

This evaluation is divided into three parts. Section 6.4.2 considers the correctness and fragility of the algorithm; it demonstrates the exact match property under resolution and quality scaling and examines the sensitivity of the watermark to changes in the detection key and to recompression. Section 6.4.3 evaluates the scalability of the algorithm using measures of detectability and graceful improvement. Finally, section 6.4.4 revisits deliberate tampering using mark transfer and collage attacks, which successfully defeated (section 5.2.3, page 192) the original algorithm,

6.4.1 Experimental Framework

The evaluation process in these experiments is very similar to that used in the experiments for the unsecured algorithm (section 5.2.1.1, page 161). If the testing for a particular property involves any deviations from the method described below, the differences are noted in the section on that property.

Embedding

The watermark is embedded as part of the JPEG2000 compression process, using the algorithm described in section 6.1.8 (page 229). JPEG2000 compression is performed using the JasPer algorithm [1]. We watermark 20 images using 10 embedding keys to produce 200 watermarked images, each compressed using 6 resolution layers and 6 quality layers[8] with compression rates 0.01, 0.02, 0.03, 0.04, 0.05 and 0.9999.

Each feature is formed using $\eta = 4$ pseudorandomly selected coefficients, and $a = 1$ additional bit of each coefficient is included in the generation of intra-resolution features for each additional marked bit below the $(j-1)$th bit plane. The secret key is hashed with the most significant bit plane of the lowest resolution layer $\Phi = 1$.

[8] The evaluation (section 5.2.2.2, page 188) of the previous version of this algorithm suggested that the lowest 2 of 8 quality layers, compression rates of 0.0025 and 0.005, allowed too few bits to be extracted for some images.

For sections 6.4.2 and 6.4.3 (pages 245 and 250), the global strength α is adjusted to ensure that the 99th percentile S-CIELAB CIEDE 2000 error caused by watermark embedding, for a 96dpi Dell 1702FP (Analog) monitor viewed at 46cm is close to, but not exceeding, 4ΔE, so that all full watermarked images have a similar perceptual quality. The mark transfer and collage attacks, performed in section 6.4.4, require multiple images watermarked with the same embedding strength. A fixed strength $\alpha = 0.3580455$ is used for all images in that section, to allow the attacks to be implemented.

In some cases individual images are tested to compare with the results of the previous algorithm. For these images the value of α is chosen to match the watermarking induced perceptual distortion in the corresponding test of the previous algorithm; the chosen value is reported with the experiment.

Attacking

In sections 6.4.2.1, 6.4.2.2 and 6.4.3 no attacks are performed; in the remaining sections each image undergoes an attack. In section 6.4.2.3 the attack consists of decompression into RGB format and recompression using JPEG2000 with the same compression parameters as were used during embedding. A mark transfer attack is performed in section 6.4.4.1, and a mark transfer attack with quality scaling in section 6.4.4.2, while section 6.4.4.3 uses a collage attack.

Detection

For each watermarked image, watermark detection is performed for the full image and each of the five resolution scaled and five quality scaled subimages defined during embedding. In most cases, the detection key is the same as the embedding key.

The watermark is treated as a sequence of bits, and any bit for which the candidate watermark u^c and the extracted watermark u^d do not match is counted as a bit error. Any bit error indicates that the values of either the watermarked coefficient or one of the η feature coefficients used to generate that watermark bit are different to those expected for an untampered image. To calculate the total number of bit errors corresponding bits in the sign-magnitude representations of corresponding candidate and extracted watermark elements are compared, and all mismatching bits are counted.

The number of bit errors and the number of (non-missing) extracted bits is recorded, and the bit error rate calculated, for all subimages at each resolution or quality level. If the BER for any image is above zero, that image is deemed inauthentic.

6.4.2 Correctness and Fragility

The secured algorithm should still retain an exact match between the candidate and extracted watermarks after resolution and quality scaling, when the correct detection key is used. This exact match property is demonstrated in section 6.4.2.1.

When an incorrect detection key is used, there should be a substantial mismatch between candidate and extracted watermarks, even when that key is close to the one used for embedding. This key sensitivity was shown to be poor for the unsecured algorithm (section 5.2.1.3, page 164). The same experiments are repeated for the secured algorithm in section 6.4.2.2.

In section 6.4.2.3 the fragility of the secured algorithm is tested after recompression. Testing of non-geometric and geometric manipulations other than recompression (see section 5.2.1.4, page 166), is not repeated for the secured algorithm as it is deemed unnecessary. Because all such manipulations involve recompression, and so are expected to produce distortion least equal to that caused by recompression,[9] only fragility to recompression need be tested. Provided the results of section 6.4.2.3 show that the algorithm is fragile to recompression, we can reasonably conclude that the algorithm will be fragile to non-geometric and geometric manipulations. Similarly, deliberate tampering in the spatial domain, in which no measures are taken to defeat the watermark detector, is also highly likely to produce error rates similar to (or worse than) recompression, so no explicit tests of spatial tampering are performed.

6.4.2.1 Exact Match under Resolution/Quality Scaling

To test that the algorithm is correctly implemented and that the candidate and extracted watermark sequences match exactly regardless of resolution and quality scaling, the experiments of section 5.2.1.2 (page 162) are repeated. If the exact match property holds we should see a BER of zero, not only for the full image but also for any resolution or quality scaled subimage.

Results and Analysis

The bit error rate for all watermark detections in both resolution (table 6.1) and quality (table 6.2) scaled subimages is zero.

Relative to the previous algorithm, a lower percentage of the watermark is available at low resolution layers and a higher percentage is available at low quality layers but both show zero mismatches between candidate and extracted marks.

[9] That all tested spatial and geometric manipulations produced errors similar to or worse than those for recompression can be seen from the tests performed on the previous unsecured implementation in section 5.2.1.4 (page 166)

Table 6.1: Total bit errors and extracted bits for resolution scaled subimages – data for each row is obtained from the 200 subimages which have the shown number of resolution layers.

resolution layers	bit errors	extracted bits	BER
1	0	364960	0
2	0	787988	0
3	0	1650735	0
4	0	2951445	0
5	0	4460973	0
6	0	5267485	0

Table 6.2: Total bit errors and extracted bits for quality scaled subimages – data for each row is obtained from the 200 subimages which have the shown number of quality layers.

quality layers	bit errors	extracted bits	BER
1	0	164158	0
2	0	352294	0
3	0	544398	0
4	0	859333	0
5	0	1077475	0
6	0	5267485	0

6.4.2.2 Key Sensitivity

To test the sensitivity of the watermark to differences between the embedding and detection keys, the experiments of section 5.2.1.3 (page 164) are repeated.

Experiments are performed using the method described in section 6.4.1 (page 243) with the processing phase skipped so there are no attacks. However, rather than use the same embedding and detection keys, we use a single embedding key $sk_e = 101$ and the detection process is performed using 100 sequential detection keys $sk_d \in \{1, 2, ..., 100\}$ on each quality or resolution scaled subimage.

The implementation of the secured algorithm uses the pseudorandom number generation algorithm from the Crypto++ compilation [35], in preference to the rand function used in the previous version. Furthermore, successive calls to the generation function with the same seed, rather than re-seeding with a location based seed, are used to generate values given different indices, bit positions and sequence positions. This should prevent the appearance of unusually low error rates for certain keys close to the embedding key, which was a problem with the previous algorithm. These changes have been been made to correct the problem of unusually low error rates for certain keys close to the embedding key, which was identified during the key sensitivity analysis of the previous algorithm (section 5.2.1.3, page 164).

If the algorithm is sensitive to changes in the detection key, error rates for all tested (incorrect) detection keys should be high, and there should be no trend towards lower error rates as the detection key nears the embedding key.

Results

The changes to the pseudorandom number generation have eliminated the trend towards lower error rates for some detection keys seen in the previous algorithm, and we obtain high, though less than 50%, error rates for all tested keys.

Error rates are lower than 50% and become lower still as more high-resolution coefficients are included (table 6.3). This appears to be due to an imbalance in the ratio between the numbers of 0 and 1 bits in the candidate and extracted watermarks (see section E.2.1.1, page 526)

Figures 6.2 and 6.3 show, for the first 5 images,[10] the BERs for all scaled subimages, plotted against the detection key used. The improved pseudorandom generation has been effective; unlike the results of the previous algorithm (cf. figures 5.3 and 5.4, page 165), both figures show that there is no bias towards lower BERs when the detection key is close to the embedding key.

[10] The full, 20-image version of this graph is too cluttered to be clear.

Table 6.3: BER for resolution scaled subimages when detecting with an incorrect key.

resolution layers	bit errors	extracted bits	BER
1	1560412	3585953	0.4351
2	3177748	7683491	0.4136
3	6427558	16065461	0.4001
4	11180661	28730221	0.3892
5	16567945	43580103	0.3802
6	19520076	51641587	0.3780

Table 6.4: BER for quality scaled subimages when detecting with an incorrect key.

quality layers	bit errors	extracted bits	BER
1	581396	1487803	0.3907749
2	1221525	3144905	0.3884140
3	1952944	4989187	0.3914353
4	3049892	7920087	0.3850831
5	3900473	10130683	0.3850158
6	19520076	51641587	0.3779914

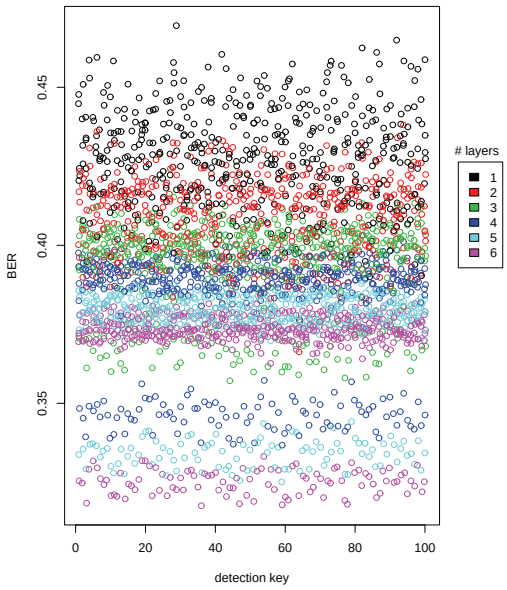


Figure 6.2: Bit error rates for resolution scaled subimages with incorrect detection key – As the detection key nears the embedding key, the BER remains high.

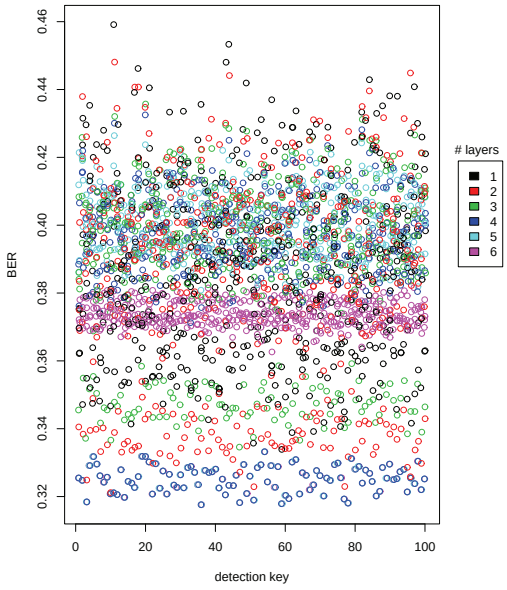


Figure 6.3: Bit error rates for quality scaled subimages with incorrect detection key – As the detection key nears the embedding key, the BER remains high.

6.4.2.3 Recompression

This section shows the errors introduced when the watermarked image is decompressed and recompressed with no additional changes. Any change that involves decompression, (malicious or non-malicious) manipulation of the image, and recompression is likely to produce error rates similar to or worse than those obtained in this section.

Results

Error rates for resolution scaled subimages are consistently high (table 6.5), while there is a trend towards lower error rates at lower quality layers (table 6.6), which remain largely undisturbed by recompression.

The increased image dependence results in higher BERs than with the previous algorithm (section 5.2.1.4, page 166). This can be attributed to the increased dependence of the watermark on the image coefficients. Previously, a change to a single coefficient could affect at most the number of watermark bits embedded in that coefficient. In the secured algorithm, changes to a single coefficient can also affect any watermark bit whose feature sequence contains that coefficient.

Table 6.5: Total bit error rates after recompression for resolution scaled subimages.

resolution layers	bit errors	extracted bits	BER
1	166305	361341	0.4602
2	371866	768815	0.4837
3	773564	1583530	0.4885
4	1345196	2781988	0.4835
5	1979526	4134974	0.4787
6	2271120	4809311	0.4722

These BERs show that the algorithm is fragile to recompression, and will therefore be fragile to attacks involving recompression, in which no specific efforts are made to defeat the watermarking algorithm. This includes non-geometric and geometric processing and spatial-domain tampering.

6.4.3 Scalability

The scalability of the secured algorithm is examined using the detectability and graceful improvement measures established in the previous chapter (section 5.2.2.1, page 186).

Table 6.6: Total bit error rates after recompression for quality scaled subimages.

quality layers	bit errors	extracted bits	BER
1	49141	168265	0.292
2	129465	381169	0.3397
3	233547	618858	0.3774
4	368241	927679	0.3969
5	459566	1127559	0.4076
6	2271120	4809311	0.4722

6.4.3.1 Detectability

To evaluate the scalability of the secured algorithm with respect to the detectability property, the number of correctly extracted watermark bits from the lowest resolution and lowest quality subimages, is measured. A threshold of 30 correctly extracted bits (with 0 errors) is taken as the minimum acceptable level of detectability.[11]

Results

In all tests, no fewer than 900 bits are correctly extracted under resolution scaling to the lowest resolution layer, $\frac{1}{1024}$th the area of the original, and no fewer than 200 bits are correctly extracted under quality scaling to the lowest quality layer, compression rate 0.01.

Figures 6.4 and 6.5 show the detectability results under resolution and quality scaling respectively, for each of the 20 original images and 10 secret keys. As is expected for an image adaptive algorithm, the detectability results vary primarily according to the original image, with the results for different keys being relatively similar.

Based on these results, the fraction of images with detectability below 30 is estimated to be roughly 0.00018 for resolution scaling and roughly 0.0000012 for quality scaling, suggesting that for the vast majority of images, the watermark will be detectable at $\frac{1}{1024}$th the area of the original or a compression rate of 0.01. The details of the estimation process are discussed in section section E.2.2 (page 528).

[11] Recall that a threshold of 30 may or may not be sufficient depending upon the application requirements and bit error probabilities (section 5.2.2.1.1, page 186).

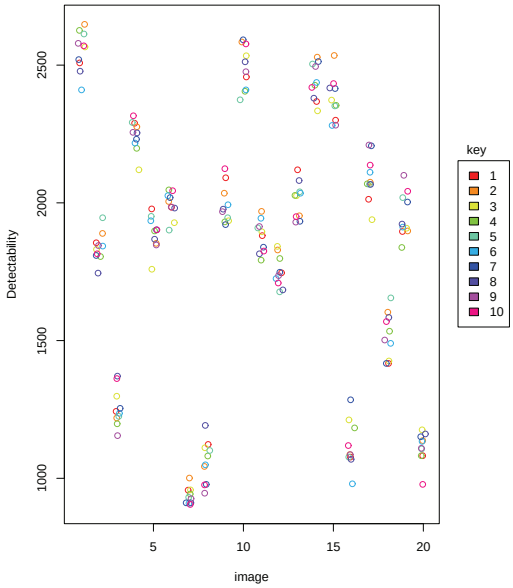


Figure 6.4: Resolution detectability: number of correctly extracted bits in the lowest resolution layer.

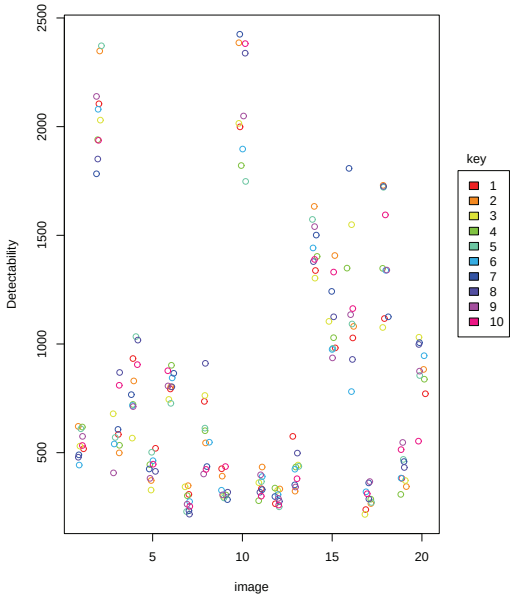


Figure 6.5: Quality detectability: number of correctly extracted bits in the lowest quality layer.

6.4.3.2 Graceful Improvement

To evaluate the scalability of the secured algorithm with respect to the graceful improvement property, the graceful improvement values for each watermarked image are measured. Recall that a watermark that is perfectly allocated, to match the improvement in perceptual quality provided by each layer, produces a graceful improvement value of 1, while the worst possible allocation produces a graceful improvement value of 0.

Results

The graceful improvement values for resolution scaled images (figure 6.6) all exceed 0.9, and the results for quality scaled images (figure 6.7) are in the range 0.8 to 0.9.

This suggests that the algorithm is not distributing the watermark appropriately amongst all quality layers. A more detailed examination of the quality scaled images (section E.2.2, page 528), reveals that too much of the watermark is being assigned to higher quality layers, and too little to the lowest quality layer. Thus, although the algorithm will not be vulnerable to the attacks of section 6.3 (page 239), performance on low quality images could be improved. This could be achieved by improving adaptation to the human visual system.

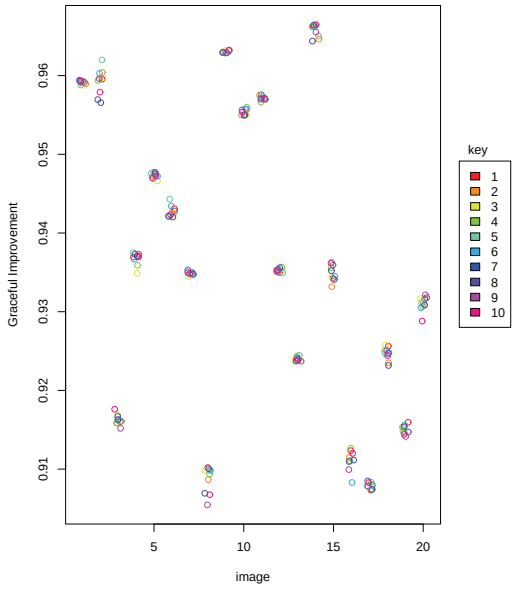


Figure 6.6: Graceful improvement for resolution scaled images.

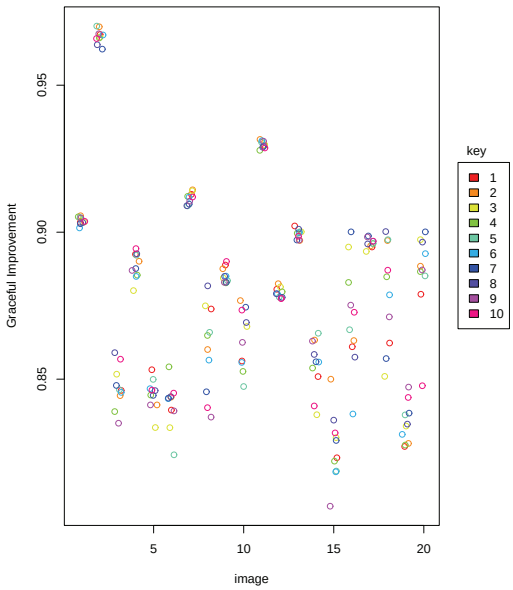


Figure 6.7: Graceful improvement for quality scaled images.

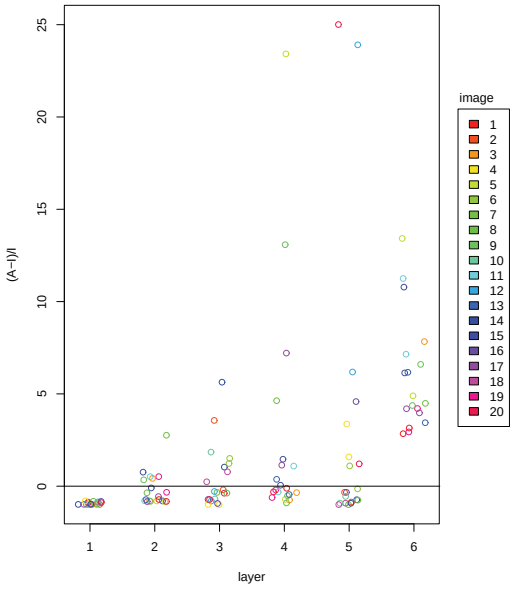


Figure 6.8: Per-layer difference between actual and ideal numbers of extracted bits as a fraction of ideal extracted bits, averaged across keys.

6.4.4 Tamper Detection

The main feature of this watermarking algorithm, over the unsecured algorithm of the previous chapter, is its ability to detect deliberate tampering. Although the previous algorithm detected simple tampering attacks in both the spatial and transform domains, it failed to detect mark transfer attacks and collage attacks.

The security of the new algorithm is tested against mark transfer and collage attacks, in sections 6.4.4.1 and 6.4.4.3 respectively. Wavelet domain tampering is not specifically tested; as both mark transfer and collage attacks are applied in the wavelet domain, so any algorithm that can successfully detect these attacks can also detect more simple forms of wavelet domain tampering. Similarly, spatial tampering is not tested here, as the successful detection of recompression ensures spatial tampering will be detected (see section 6.4.2.3, page 250).

6.4.4.1 Mark Transfer Attack

To test the secured algorithm against mark transfer attacks, all 20 images are watermarked using the same parameters (section 6.4.1, page 243), with a fixed embedding strength of $\alpha = 0.3580455$.

For each image and secret key pair, a mark transfer attack is applied, using the method described in section 5.2.3.7 (page 201), to move the watermark from the valid image to an unwatermarked copy of the next original image in the test set, compressed using the same parameters. This results in 200 attacked images.

The detection algorithm is applied to each attacked image, using the correct secret key and parameters. The number of extracted bits and the number errors are recorded and the BER is calculated for each resolution or quality scaled subimage.

Results

The algorithm is detects the mark transfer attacks for all images. The average BERs for resolution (table 6.7) and quality scaled (table 6.8) subimages all exceed 46%, and the BERs for individual images (section E.2.3.1, page 532) all exceed 39%.

Table 6.7: Total bit error rates after mark transfer attack for resolution scaled subimages.

resolution layers	bit errors	extracted bits	BER
1	182502	368623	0.4951
2	381824	774980	0.4927
3	781247	1593460	0.4903
4	1379290	2840930	0.4855
5	2033673	4215690	0.4824
6	2463053	5130671	0.4801

Table 6.8: Total bit error rates after mark transfer attack for quality scaled subimages.

quality layers	bit errors	extracted bits	BER
1	68802	147516	0.4664
2	154184	329773	0.4675
3	277140	587216	0.472
4	419552	885595	0.4738
5	543412	1142846	0.4755
6	2463053	5130671	0.4801

6.4.4.2 Mark transfer with Quality Scaling

In the test of section 5.2.3.7.1 (page 203), the unsecured algorithm failed to detect the placement of a single pixel spot on the cheek of the Lena image using a mark transfer attack after quality scaling to a compression rate of 0.02. In this section, the same test is applied to the secure algorithm. The 99th percentile embedding induced distortion for the previous test was 2.1923ΔE, we use a strength of $\alpha = 0.11718$ that results in a slightly lower embedding-induced distortion of 2.069ΔE.

Figure 6.9: Tampered image with transferred watermark, preserving MSB only.

Figure 6.9 shows the tampered image, with spot applied using a mark transfer attack, and figure 6.10 shows the quality scaled tampered image at compression rate 0.02. Applying the detection algorithm to the quality scaled tampered image at compression rate 0.02, produces the output False, and the tamper map in figure 6.11. So the mark transfer attack which successfully defeated the previous algorithm is unsuccessful against the secured version.

Note that although the the tampering is correctly identified and the identified region includes the tampered pixel, the tamper location is quite poor. This is expected for small tampered regions, which contain too few coefficients to produce the cluster of errors required for better location.

Figure 6.10: Tampered image with transferred watermark, preserving MSB only, compression rate 0.02.

Figure 6.11: Tamper map with transferred watermark, preserving MSB only, compression rate 0.02.

6.4.4.3 Collage Attack

To test the secured algorithm against collage attacks, all 20 images are watermarked using the same parameters (section 6.4.1, page 243), with a fixed embedding strength of $\alpha = 0.3580455$.

For each image and secret key pair, a collage approximation of the corresponding original image is constructed from the remaining 19 images that were embedded with that secret key, forming a total of 200 collages.

The detection algorithm is applied to each collage image, using the correct secret key and parameters. The number of extracted bits and the number errors are recorded and the bit error rate is calculated for each resolution or quality scaled subimage.

In section 5.2.3.8 (page 206), the previous algorithm was shown to be vulnerable to collage attacks, with an error rate of zero for a full resolution and full quality version of the Greek isles image. To demonstrate security against collage attacks we should have high error rates for all tested resolution and quality scaled subimages.

Results

The collage attacks are detected in all tested images. The average BERs for resolution scaled images (table 6.9) decrease as higher resolution layers are added; however, all levels of scaling have BERs above 35%. The average BERs for quality scaled images (table 6.10) are fairly constant, at 35% all levels of quality scaling.

These error rates are less than 50% because substantially more than half of the watermark bits have the value zero. Assuming that, for a full collage attacked image, the candidate watermark sequence is independent from the extracted watermark sequence and both sequences contain roughly 77% zeroes (see the discussion in section E.2.1.1, page 526), the expected error rate would be around 35%, as seen here. The larger percentage of zero watermark bits in higher resolution layers also explains the trend of decreasing error rate as higher resolution layers are received.

In the test of section 5.2.3.8 (page 206), the unsecured algorithm failed to detect the object removal in the lower section of the Greek isles image using a collage attack. The same test is applied to the secure algorithm. The embedding induced distortion for the previous test was 3.9390ΔE, the strength is set to $\alpha = 0.452024$, which results in a slightly lower distortion of 3.7724ΔE.

Table 6.9: Total bit error rates after collage attack for resolution scaled subimages.

resolution layers	bit errors	extracted bits	BER
1	145467	355040	0.4097
2	287760	739999	0.3889
3	557884	1482373	0.3763
4	896873	2449650	0.3661
5	1178026	3291317	0.3579
6	1231829	3452302	0.3568

Table 6.10: Total bit error rates after collage attack for quality scaled subimages.

quality layers	bit errors	extracted bits	BER
1	63770	180794	0.3527
2	136975	390009	0.3512
3	241486	681328	0.3544
4	338947	952177	0.356
5	448036	1257103	0.3564
6	1231829	3452302	0.3568

Figure 6.12: Watermarked Greek isles image

Figure 6.13: Wavelet tampered image.

Figure 6.14: Tampered image with 21-image collage.

Figure 6.12 shows the watermarked image, figure 6.13 a wavelet tampered version with the satellite removed, and figure 6.14 shows the collage reconstruction of the wavelet tampered image.

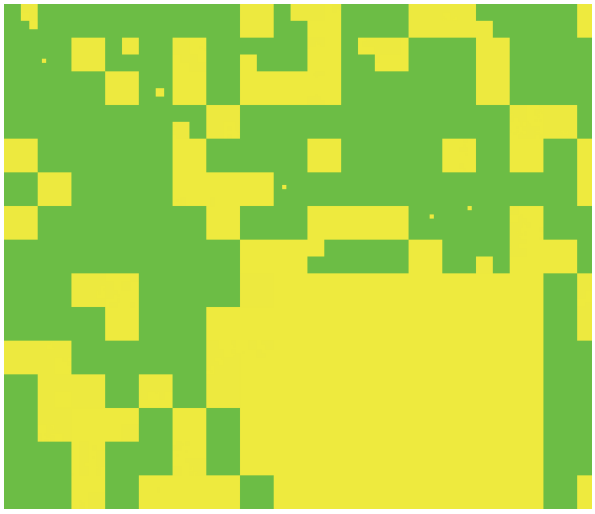


Figure 6.15: Tamper map with 21-image collage.

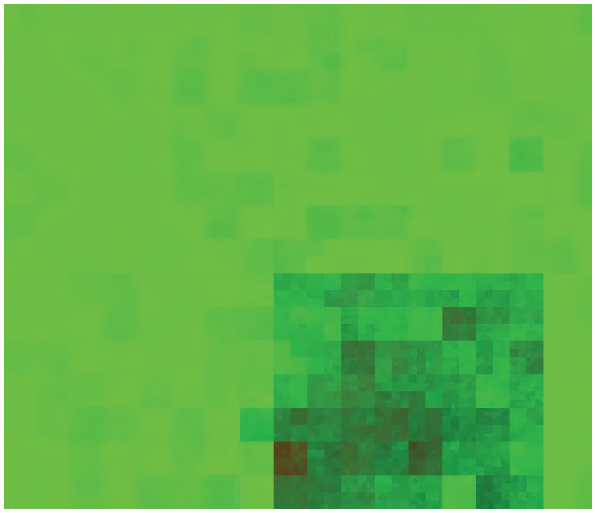


Figure 6.16: Tamper map with 21-image collage, without full saturation. The tampered region is more easily distinguished.

Watermark detection on the collage attacked image produces the output False, with the tamper map in figure 6.15. The collage attack, which successfully defeated the previous algorithm, is correctly detected.

Note that although the tampering is located, this location is not as precise as in successful detections by the previous algorithm, as many coefficients outside the tampered region are also identified as tampered. This problem may be mitigated by the production of more informative tamper maps.

When, for example, rather than setting the red and/or green components of each region to 255 (as described in section 5.11, page 194), we add 2^{κ} to the green component for each bit position κ at which a matched bit occurs, whenever there is an exact match between candidate and extracted watermark elements, and 2^{κ} to the red component for each bit position κ at which a mismatched bit occurs, whenever there is not an exact match, the tampered region is more clearly identified (figure 6.16).

6.5 Conclusion

The design changes made to the watermark generation procedure result in stronger, less predictable dependencies that extend over multiple blocks. Because each watermark bit is formed using features from several image coefficients, the link between the image and the watermark has been strengthened. Pseudorandom selection of these coefficient sequences, using a hash of the secret key and most significant bits of the lowest resolution layer, makes it difficult for an attacker, who does not know the secret key, to identify which watermark bits correspond to a particular tampered region and modify the watermark accordingly. Roughly half of the feature sequences are chosen from anywhere within an entire resolution layer, removing the blockwise independence that caused the vulnerabilities of the previous algorithm (chapter 5). The remaining feature sequences are chosen from within a codeblock, where greater predictability of the effects of quality scaling allows the use of more sensitive features. Testing demonstrates that these design changes defeat both mark transfer and collage attacks (sections 6.4.4.1 and 6.4.4.3 respectively), which were the two successful attacks against the previous version of the algorithm.

The problem of reduced sensitivity to an incorrect detection key, as the detection key approached the embedding key, has been solved by upgrading the pseudorandom generator implementation to use the Crypto++ library. Key sensitivity tests (section 6.4.2.2) on the secured algorithm show consistent results across all tested detection keys. If this solution were applied to the basic algorithm, similar key sensitivity results could be expected.

The other properties of the previous algorithm have been successfully maintained. Tests in section 6.4.2.1 showed that, for untampered watermarked images, the exact match

between candidate and extracted watermarks still holds under resolution scaling to $\frac{1}{1024}$th the original image area and quality scaling to a compression rate of 0.01.

The watermark is still fragile to decompression and recompression, as demonstrated by the tests in section 6.4.2.3. Thus the watermark will also be fragile to all types of processing that involve recompression, including all those tested in sections 5.2.1.5 and 5.2.1.6, and tampering in the spatial domain. Furthermore, the results of the mark transfer and collage attacks (both performed in the wavelet domain) suggest that the watermark will still be fragile to modifications in the wavelet domain.

Scalability testing, in section 6.4.3, suggests that at least 30 bits of the watermark will be detectable in the majority of images under resolution scaling to $\frac{1}{1024}$th the original image area and quality scaling to a compression rate of 0.01, and that the watermark is reasonably well allocated amongst the resolution and quality layers tested. However, both these detectability and graceful improvement results could be enhanced by increasing embedding in lower layers, particularly quality layers, perhaps through the use of improved human visual system models.

A small amount of scalability has been sacrificed for security. The inclusion of the Φ most significant bit planes of the lowest resolution layer in the hash used for feature sequence selection means that using the same secret key to watermark different images will result in different feature sequences. Thus an attacker cannot construct a collage attack using multiple images watermarked with the same secret key. However, this means that all Φ bit planes must all be correct before the candidate watermark will be correctly generated. Thus any quality scaling that is sufficiently severe to result in an incorrect bit amongst the Φ most significant bit planes of the lowest resolution layer will cause the algorithm to fail. This seems a reasonable sacrifice, as the importance of these bit planes to image quality is such that the scaling required would result in an image of unacceptable quality. Finally, because the value of Φ is a parameter of the system, it is possible to adjust the balance between security and scalability by choosing the value of Φ.

The tamper location ability of the algorithm has also been reduced. A mismatch between a pair of candidate and extracted watermark bits could be the result of changes either to the watermarked bit itself, or to any one of the η coefficients in the corresponding feature sequence. Thus a single bit error will typically be shown in $\eta + 1$ locations on the tamper map. Because tampering is usually applied to a set of coefficients in the same spatial region, the concentration of potential errors can be used to distinguish tampered areas from the more randomly spread errors shown at untampered coefficients. More sophisticated methods of tamper map creation may serve to restore some of the lost tamper location accuracy.

Unlike previous image authentication watermarks for scalable compression systems, this algorithm provides not only resolution and quality detectability, but also graceful improvement. While the detectability property allows the authentication of highly scaled images, the graceful improvement property is vital (as was shown in section 6.3) to ensuring the integrity of the full image.

Chapter 7

Concluding Remarks

With widespread demand for both high quality and mobile content, scalable compression offers an appealing solution for digital content delivery, that caters to the widest possible range of users. Through selective discarding of resolution and quality layers, it allows an image of sufficient resolution and quality to satisfy high-end users to be easily and efficiently adapted to suit both changing network conditions and a range of devices with differing limitations.

Yet, given how easily digital content can be stored and retransmitted, the potential for widespread intellectual property rights violation is of significant concern. Digital watermarking continues to play an important role in protecting against such violations, by discouraging or preventing unauthorised copying. It also offers a variety of other applications in the areas of data enrichment, steganography and authentication. With increasing network speeds, display resolutions and user expectations, at the high end, and a combination of legacy networks and increasingly compact devices, at the low end, the need to watermark scalably compressed content can only increase.

Traditionally, watermarking algorithms have not been designed to protect scalably compressed images. Even watermarking algorithms with high robustness were only intended to tolerate processing that did not compromise the acceptability of the image to the user. However, the image quality or resolution most acceptable to a user with a low bandwidth connection or a small screen size is well below what would be considered acceptable for a high resolution, high quality image. As a result, the reductions in resolution or quality that are common in scalable compression can result in watermark detection failure.

Scalable watermarking has emerged as a solution to this problem, with a number of researchers designing watermarking algorithms intended specifically for use with scalably compressed images. However, the requirements for a scalable watermarking algorithm were not particularly well defined, so some researchers developed watermarks that did not protect the full image while others developed watermarks that were not detectable in a highly scaled image. Furthemore, although scalable image compression allows two types

of scalability – resolution and quality – scalable watermarking algorithms focused only on one type of scalability or the other.

This thesis makes four main contributions to the field of scalable watermarking. The first of these is a formal definition for a scalable watermarking algorithm. The remaining three are scalable image watermarking algorithms that support both resolution and quality scalable compression. In particular, the second is a non-blind, resolution and quality scalable watermarking algorithm that uses spread spectrum techniques; the third is a blind, resolution and quality scalable watermarking algorithm, which also allows an exact match between candidate and extracted watermarks in a scaled but otherwise unmodified image; and the fourth is an improved version of the blind algorithm with a specific emphasis on authentication, which retains most of the scalability of the previous algorithm, including the exact match property, but also provides security against sophisticated counterfeiting attacks. Each of these is discussed in more detail in section 7.1.

However, these contributions form only the beginning of a comprehensive study of scalable watermarking and there are a number of open problems that could not be addressed in this thesis. The limitations of this thesis and the many opportunities for future work are addressed in section 7.2

7.1 Main Contributions

7.1.1 Definition of Scalable Watermarking

In chapter 3, the first formal definition of a scalable watermarking algorithm was developed. This definition states that a scalable watermark should possess two properties, detectability and graceful improvement: To satisfy detectability, the watermark must be detectable in any portion of the scaled content which is of acceptable quality, while to satisfy graceful improvement, increased portions of the scaled content must provide reduced error in watermark detection, appropriate to the improved content quality.

Prior to this, definitions of scalable watermarking had been informal descriptions, largely confined to the introductory paragraphs of papers that presented new scalable watermarking algorithms. These descriptions drifted between two main schools of thought. The first believed that a scalable watermark should be detectable as soon as the lowest layer is received, so embedding should occur in the lowest layer. The second took the approach that a scalable watermark should become stronger as more of the image is received, so that all layers are afforded protection.

The definition, in section 3.1.1 (page 52) of this thesis, unifies both of these approaches. The detectability property ensures that the watermark is detectable upon receipt of the

lowest layer, while the graceful improvement property ensures that the higher layers are also protected.

From this definition quantative measures of detectability and graceful improvement can be developed, which allow claims of scalability to be verified or rejected. This thesis also describes the development of such measures (sections 3.1.2 and 5.2.2.1, pages 54 and 186), along with a statistical framework for comparing scalability (section 3.2.4.2, page 77).

7.1.2 Non-Blind Resolution and Quality Scalable Watermarking

In chapter 4, a non-blind scalable watermarking algorithm was developed. A spread spectrum technique (section 2.1.5.4, page 28) was used as the basis for this algorithm because of its high robustness to image processing operations. The algorithm provided both resolution scalability and quality scalability, using a single watermark. This was achieved by first selecting the embedding locations to ensure resolution scalability, and then employing an adaptive embedding strength, based on aspects of the human visual system, to provide quality scalability.

Previous authors of scalable watermarking algorithms had considered only one type of scalability (with the exception of Sun, Chang et al. [181] whose algorithm allowed either type of scalability but required a single type to be chosen at the embedding stage). The difficulty, in achieving both types simultaneously, lay in a tradeoff between resolution and quality scalability that was identified in section 4.2.3 (page 111). This tradeoff occurs because resolution scalability necessarily involves substantial embedding in the lowest resolution layer, whereas quality scalability requires a high embedding strength, but the simultaneous use of low-resolution embedding and a high embedding strength violates watermark imperceptibility constraints.

The success of HVS-based strength adaptation as a mechanism for alleviating this tradeoff was demonstrated in section 4.3.2 (page 122). The resulting watermark was detectable after resolution scaling to $\frac{1}{256}$th the original area and after quality scaling to $\frac{1}{100}$th the original file size, and detection results continued to improve until the full image was received.

7.1.3 Blind Resolution and Quality Scalable Watermarking

In chapter 5, a blind scalable watermarking algorithm was developed, because non-blind watermarking algorithms are unsuitable for some applications. To provide both resolution and quality scalability in a blind watermarking scenario, it was necessary to surmount the additional problem of maintaining watermark detector synchronization without access to

the original image, after potentially severe resolution or quality scaling. This was done by combining threshold based coefficient selection with quantization based embedding (to ensure that all received, selected coefficients would then be reselected at the detector) and then using coefficient indices invariant to resolution scaling and local embedding strengths invariant to quality scaling (to ensure that the original watermark element and embedding strength could be generated for each reselected coefficient). The resulting watermark allowed resolution scaling to $\frac{1}{1024}$th the original area and quality scaling to $\frac{1}{100}$th the original file size (section 5.2.1.2, page 162).

Quantization based techniques (section 2.1.5.3, page 26) are often favoured for blind watermarking because they are not subject to interference from the original image [22]; this allows an exact match to be obtained between candidate and extracted watermarks provided no processing has occurred. An important design goal in chapter 5 was to maintain this exact match between candidate and extracted watermarks at all levels of resolution and quality scaling. This was attained by truncating the candidate watermark according to the pattern of scaling that was identified at the detector, so that the candidate watermark contained only those bits which had not been lost due to scaling. This eliminated scaling errors from the detected watermark yet still allowed embedding in all layers.

The exact match property was demonstrated experimentally, in section 5.2.1.2 (page 162), for resolution scaling to $\frac{1}{1024}$th the original area and quality scaling to $\frac{1}{400}$th the original file size, and theoretically, in appendix D.3, for all levels of resolution and quality scaling. Previous methods of eliminating scaling errors involved restricting the watermark embedding to occur in only those layers that would not be lost during scaling, which provided detectability but no protection of higher layers. Because the method presented in this thesis requires no such restriction, it provides detectability without sacrificing graceful improvement.

7.1.4 Resolution and Quality Scalable Authentication

In chapter 6, a semi-fragile scalable watermarking algorithm was developed specifically for scalable image authentication. Experiments (section 6.4.1, page 243) showed the algorithm to be both resolution and quality scalable for resolution scaling down to $\frac{1}{1024}$th the original area and quality scaling to $\frac{1}{100}$th the original file size, yet fragile to all other tested modifications.

This algorithm, while based on that of chapter 5, is secure against mark transfer and collage attacks (sections 6.4.4.1 and 6.4.4.3, pages 255 and 259), which proved problematic for the basic algorithm (cf. sections 5.2.3.7.1 and 5.2.3.8, pages 203 and 206). This security was obtained by improving the watermark design to be both blockwise-

and image-dependent. Specifically, each watermark bit was generated according to a 'feature sequence' of pseudorandomly selected image coefficients; some feature sequences were allowed to span multiple codeblocks within a resolution layer, to provide blockwise dependence, while others were restricted to a single codeblock, to maximize tamper sensitivity.

The improved security was achieved while preserving the exact match property of the basic algorithm, thereby guaranteeing no false alarms from a scaled but otherwise untampered image. This property was demonstrated experimentally in section 6.4.2.1 (page 245), for resolution scaling down to $\frac{1}{1024}$th the original area and quality scaling to $\frac{1}{100}$th the original file size, and theoretically in appendix E.1, for any level of resolution and quality scaling in which at least the Φ most significant bitplanes of the lowest resolution layer are unchanged.

Prior to the development of this algorithm, scalable watermarking research had focused almost entirely on robust algorithms, with only one semi-fragile algorithm published, by Steinder et al. [174], which itself provided only resolution detectability and not graceful improvement. Semi-fragile watermarking research had produced algorithms designed to tolerate compression, but they did not provide graceful improvement either. As was demonstrated in section 6.3 (page 239), semi-fragile algorithms that provide detectability but not graceful improvement allow tampering with higher layers to pass undetected (regardless of the other security properties of the watermarking algorithm). The semi-fragile algorithm developed in this thesis provides both detectability and graceful improvement, thus it permits resolution and quality scaling down to the base layer without compromising the security of higher layers.

7.2 Limitations and Future Work

A number of the problems encountered in the course of this work could not be fully addressed within the scope of this thesis. These problems remain open, and each is discussed, in the following sections, in terms of desirable future work.

Potential future work arising from this thesis includes, but is not limited to, the development of new and better scalable watermarking algorithms. The problems discussed in sections 7.2.1 through 7.2.3 are quite general in nature and should be of wide benefit to watermarking research. Sections 7.2.4 through 7.2.7 focus on scalable watermarking specifically, and suggest a range of possible extensions to the work in this thesis.

7.2.1 A Standard, Comprehensive Database of Test Images

This thesis used some well known images from the Signal and Image Processing Institute at the University of Southern California (USC-SIPI) [169], as well as a number of images

from Petitcolas' database [141], including copyright photos courtesy of Robert E. Barber, Karel de Gendre and Éric Labouré, and many copyright-free images from the Gimp-Savvy archive [56]. Yet none of these databases provides a large number of high-quality images with a range of subject matter and of sufficient size to be ideal for scalable watermarking research. While several benchmarking tools exist, good image databases are difficult to find. Even the widely used 'classic' images are of dubious quality, and their benefit now lies primarily in their familiarity to researchers.

A standard, comprehensive and widely available database of high resolution and high quality images for watermark testing does not yet exist. Nor is it completely clear what an acceptably comprehensive database would contain, as different watermarking algorithms are sensitive to different image properties and certain application areas will favour particular types of images, for example medical or multi-spectral imagery. The Watermark Evaluation Testbed database, from the Video and Image Processing Laboratory at Purdue University [87] appears to be the best candidate at present, but, beyond the USC-SIPI classics, no image database has been adopted as standard by the community. The development, and more importantly the provision and promotion, of a good image database would be of great benefit to scalable watermarking research, and to watermarking research in general, as would similar databases for other media.

7.2.2 Statistical Methods for Algorithm Evaluation

The experimental and statistical framework developed in section 3.2.4.1 (page 76) allows the determination of an appropriate number of test images to conclusively demonstrate that one algorithm represents a substantial improvement over another; however, an equivalent framework for considering a single algorithm in isolation has yet to be developed. As a result, while the correct number of images could be determined for the comparison of chapter 4, the combination of 20 images with 10 embedding keys, used in chapters 5 and 6, is somewhat arbitrary.

Despite this, 20 images and 10 keys appears to be sufficient to demonstrate the fragility of the watermarks of to a wide range of attacks.[1] However, this number of images and keys may not be sufficient to allow comparison with an algorithm which is fragile to the same attacks. The estimates of expected performance were provided without error bounds, so experiments with a greater number of images could be necessary for certain applications.

The development of a method for determining the number of images required to obtain specific error bounds on watermark performance for a single algorithm is beyond the scope

[1] In every attack except quality recompression on the unsecured algorithm (section D.4.1.3, page 411), the bit error rate for each of 200 tested images was at least 10% (more usually over 30%), yet only a single bit error is required to determine that an image is inauthentic.

of this thesis, yet experimental evaluation remains an important component of watermark research and publication. Statistical methods are well established in other disciplines and a simple experimental and statistical framework, enabling a watermark researcher to determine the number of images necessary to demonstrate the performance of a single algorithm with well-defined levels of accuracy, would be a valuable contribution to the watermarking community.

7.2.3 Watermarking-Specific Human Visual System Models

The texture scoring algorithm, used in the HVS adaptive watermark of chapter 4, was developed (appendix C) using an experimentally determined combination of heuristics, according to their ability to separate texture from edges or smooth regions in three hand-classified images. This was done because no algorithm could be found with the desired properties (of a good separation between texture and edges using a single resolution in the wavelet domain) but a more thorough development of a texture based masking algorithm was not possible within the timeframe of this thesis. Such development would require, at the very least, a larger number of pre-classified images and wider investigation into possible heuristics, based primarily on human vision research. Ideally, the effectiveness of the complete masking model (both contrast sensitivity and texture adaption) would have been evaluated using psychovisual experiments involving a number of participants and viewing conditions. More sophisticated and computationally efficient masking models based on human perception would be beneficial to scalable watermarking, in particular, but also to imperceptible watermarking, more generally.

On a similar note, the HVS model used in chapters 5 and 6 was relatively basic, with the embedding strength adjusted according to the magnitude of the embedded coefficient only. As described in section 6.4.3.2 (page 253), the quality scalability of the algorithm suffered because too little of the watermark was embedded in the lowest quality layer, so both aspects of quality scalability could have been improved though the use of a more sophisticated HVS model. It was not possible to simply use the model developed in chapter 4, because the blind algorithm required that embedding strengths be reproducible at the detector, which does not have access to the original image but only a (potentially highly scaled) version of the watermarked image. For the same reason, otherwise suitable HVS models from other fields, such as compression, computer vision and biology may see limited use in watermarking algorithms unless some effort is placed in adapting these models to a blind watermarking scenario.

7.2.4 Extension to More Advanced Watermarking Tecniques

In order to better study scalability, the watermarking techniques which form the basis of the algorithms presented in this thesis were chosen to be as simple as possible, and robustness to attacks other than JPEG2000 scaling (filtering, rotation, cropping, etc.) was essentially ignored (see the discussion in section 6.3.1, page 242).

Several more sophisticated watermarking techniques have been mentioned in section 2.1.5 (page 26), which can offer superior performance to these basic techniques in terms of imperceptibility, capacity or robustness. Furthermore, there exist a variety of more specific watermarking techniques focused on achieving robustness to various specific attacks. If the methods developed in this thesis can be successfully adapted to enhance the scalability of these other techniques, a range of efficient, practical and scalable watermarking algorithms should result.

7.2.5 Near Perfect Graceful Improvement

Because the graceful improvement results were quite good for all algorithms presented in this thesis, potential techniques for increasing graceful improvement were not explored and the majority of effort was dedicated to achieving detectability. Methods to ensure detectability will, in all likelihood, remain the greater challenge in the development of scalable watermarking algorithms, yet it is worth mentioning that excellent resolution graceful improvement can be achieved using a conceptually trivial method. One simply calculates, for each resolution layer, the ideal proportion of watermark elements that should be embedded in that resolution layer (as shown in sections 3.1.2.2 and 5.2.2.1, pages 57 and 186) and then embeds that proportion elements in the resolution layer.

The same method can be applied to quality graceful improvement, as long as the precise division of the image among the quality layers is available at the time of embedding and no watermark element spans multiple quality layers. Yet it would only be applicable to resolution graceful improvement for the algorithms proposed in this thesis, as embedding occurs after the assignment of coding passes to quality layers, which can thus be influenced by the embedded watermark. It would be interesting to explore whether these algorithms could be adapted to achieve near perfect quality graceful improvement using this method, by either using multiple embedding passes or developing a compressed-domain embedding technique that minimally disturbs the layer assignment process.

7.2.6 Improved Resolution and Quality Detectability

While graceful improvement is relatively easy to achieve, the simultaneous provision of both resolution and quality detectability remains a real chalenge for scalable watermarking

research.

Good resolution scalability was obtained, down to $\frac{1}{256}$th and $\frac{1}{1024}$th of the original image area, for the algorithms proposed in chapter 4 and chapters 5 and 6 respectively, with quality scalability down to $\frac{1}{100}$th the original file size for all three algorithms. However, these are only the first steps towards resolution and quality scalable watermarking, and the estimated detectability at these levels of scaling is not sufficiently high as to be suitable for all applications. Furthermore, it is unlikely that reasonable detectability would be obtained if scaling were to be increased significantly beyond these levels.

Although this thesis does not investigate the theoretical limits of watermark scalability, it seems certain that new scalable watermarking algorithms could be developed that would exceed these detectability results, for both resolution and quality scaling, while still providing graceful improvement. Just how far resolution and quality scalable watermarking can be pushed, given the resolution/quality tradeoff, remains to be seen.

7.2.7 Extension to Other Media

This thesis examined scalable watermarking for images. For a complete picture of scalable watermarking, other scalable media, such as audio and video, and even 3-D models, should also be considered. The addition of a temporal component, with the accompanying possibility of temporal scalability, presents additional problems, yet some progress has already been made in the scalable watermarking of other media [5, 67, 105, 123, 199]. It seems likely that the properties of detectability and graceful improvement will remain important for watermark scalability in general, and the strategies, if not the specific algorithms, presented in this thesis should be applicable at least to video and perhaps to other media.

Glossary

a	Feature robustness parameter	224
$\mathfrak{a}_i$	Performance measurement for algorithm A for image I_i	74
$B(n,p)$	Binomial distribution for n trials each with success probability p	84
b	Codeblock index	48
$\mathfrak{b}_i$	Performance measurement for algorithm B for image I_i	74
C	Number of components in an image	42
c	Component index	41
$\mathsf{Compress}_{\mathrm{X}}()$	Compression algorithm for the system X	29
$\mathrm{CSF}_{\mathrm{c}}(f)$	Contrast sensitivity function for component c, at frequency f	115
$\mathrm{CSF}(c,s)$	Contrast sensitivity for subband s of component c	117
D'	Perceptual distortion measure for watermark induced distortion	16
D^F	Perceptual distortion measure for processing type F	18
d_i	Difference between a pair of measurements $\mathfrak{a}_i$ and $\mathfrak{b}_i$	79
$\mathcal{D}^{\mathcal{F}}$	Detectability measure for scaling type $\mathcal{F}$	57
$\mathsf{Decompress}_{\mathrm{X}}()$	Decompression algorithm for the system X	29
$\mathsf{Detect}_{\mathrm{X}}$	Detection algorithm for the watermarking algorithm X	13
E_s	Quantization exponent for the subband s	153
$\mathsf{Embed}_{\mathrm{X}}$	Embedding algorithm for the watermarking algorithm X	13
F	Processing, e.g. blurring, sharpening, cropping, rotation	17
f	Frequency, in cycles per degree of visual angle (cpd)	115

$\mathsf{f}(i,\,\kappa,x)$	Index of the xth feature coefficient of the sequence $V_{i,\kappa}$, as calculated by the indexing procedure of section 5.1.4.1	220
$f_{\max}$	Maximum displayable frequency	115
f_{peak}	Frequency that maximises CSF_Y	117
$\mathcal{F}$	Scaling of any type	58
FP	False positive rate	93
$\mathsf{G}()$	Watermark generation function	150
$\mathsf{g}()$	Generates pseudorandom numbers in the range $[0, 2^h)$	150
G_c	Number of guard bits for the component c	43
G_i	A pseudorandomly generated value $\mathsf{G}(v_i, i, \Lambda, I)$	217
$\mathcal{G}^{\mathcal{F}}$	Graceful improvement measure for scaling $\mathcal{F}$	60
$\mathsf{Generate}_{\text{X}}()$	Key generation algorithm for watermarking algorithm X	15
h	Exponent defining the maximum value of g	150
H_0	Null hypothesis	78
H_1	Alternative hypothesis	78
HH	Subband orientation: horizontally Highpass, vertically Highpass	43
HL	Subband orientation: horizontally Highpass and vertically Lowpass	43
I	An original (unwatermarked) image	13
i	Index of an image coefficient	149
I'	Watermarked image, $I'_{\text{X},M,sk_e} = \text{Embed}(I, M, sk_e, \Lambda)$	13
I^*	Candidate image, which *may* or *may not* be derived from I'	13
I^C	Compressed image	29
I^D	Decompressed image	29
I^e	Empty image, a mid-grey image	58
I^F	Processed watermaked image $I^F_{K,M,sk_e} = F(I'_{\text{X},M,sk_e})$	66
$I^{\mathcal{F}}$	Scaled image, $I^{\mathcal{F}} = \mathsf{Scale}_X(I^C, \Xi)$	35
$I^{\mathcal{F}}_l$	Scaled image, containing layers 0 through l	56
I^l	$I^{\mathcal{F}_l}$, a scaled image containing layers 0 through l	59
$I^{\mathcal{Q}}$	Quality scaled image	56
$I^{\mathcal{R}}$	Resolution scaled image	56
I^τ	Tampered image	197

$I^{\tau'}$	Tampered image with fake watermark	198
$\mathcal{I}$	A set of images	75
$\mathcal{I}_{\mathrm{u}}$	Set of all images of interest	74
$\mathfrak{I}_{i,\kappa}$	Set of indices from which feature coefficients may be chosen	220
iq^l	Proportional Improvement in image quality resulting from layer l	59
j_i	Number of watermark bits in the coefficient at index i	142
$j^*_{i^*}$	Calculated number of watermark bits j_i using the received image I^*	155
$\mathcal{K}$	Keyspace: a set of key pairs $\{sk_e, sk_d\}$	15
k	Number of watermark elements that have been lost due to scaling	95
L	Number of quality layers in an image	48
l	Quality layer or quality scaled subimage index	49
LH	Subband orientation: horizontally Lowpass, vertically Highpass	43
LL	Subband orientation: horizontally Lowpass, vertically Lowpass	43
M	A message	13
m_i	Number of missing bits from the coefficient at index i after quality scaling	153
$\mathfrak{m}$	Index of the least valuable non-empty layer	60
$m^*_{i^*}$	Calculated number of missing bits m_i using the received image I^*	153
$\mathfrak{M}()$	Magnitude feature: $\mathfrak{M}(v_x, v_t)$ is a bit representing the relative magnitudes of adaptively quantized coefficients v_x and v_t	223
M_s	Number of coefficient magnitude bit planes in subband s	48
m_X	Sample mean performance for the algorithm X, $\sum_{x\in\Omega_X} \frac{x}{\lvert\Omega_X\rvert}$	75

N	Number of elements in the watermark (watermark dimensionality)	59
$N(\mu, \sigma)$	Normal (Gaussian) distribution with mean μ and standard deviation σ	83
n	Sample size (number of images in a test set $\mathcal{I}$)	126
o	Orientation index: for subband orientations LL or LH, $o = 0$; for HL, $o = 1$ and for LH, $o = 2$	44
$P(X \sim d\ c)$	Probability of that random variable X with distribution d satisfies the condition c	80
p	Precinct index	47
p	A p-value	79
P_b	Number passes received for codeblock b	48
P^e	Perceptual quality of the empty image I^e	59
P^l	Perceptual quality of the subimage I^l	58
$\mathcal{P}_c$	Sample precision of component c of an uncompressed image.	41
$Q_\Delta()$	Quantized magnitude, $Q_\Delta(v) = \left\lfloor \frac{\lvert v \rvert}{\Delta} \right\rfloor \Delta$	142
$q_{i,\,\kappa,x}$	The quantization step size exponent for the feature coefficient $v_{\mathfrak{f}}(i,\, \kappa, x)$	224
$\mathcal{Q}$	Quality scaling	58
R	Number of resolution layers in an image	42
r	Resolution layer or resolution scaled subimage index	43
r	Coefficient reconstruction parameter	45
$\mathfrak{r}_{i,\,\kappa,x}$	Pseudorandomly generated number for feature coefficient selection	221
$\mathcal{R}$	Resolution scaling	58
R_{NC}	Normalized correlation detection statistic	93
s	Subband index $s = (r, o)$, represents the subband at resolution r with orientation o.	44
$\mathfrak{S}()$	Sign feature: $\mathfrak{S}(v)$ is a bit representing the sign of the adaptively quantized coefficient v	223
$\mathsf{Scale}_{\mathrm{X}}()$	Scaling algorithm for the scalable compression system X	35

s_x	Sample standard deviation of the random variable x	82
$sign()$	Sign: $sign(x) = \pm 1$, depending on whether x is positive or negative	45
$SIM()$	Similarity measure: $SIM(X,Y) = \frac{X \cdot Y}{\sqrt{Y \cdot Y}}$	28
sk_d	Watermark detection key	13
sk_e	Watermark embedding key	13
T'	Threshold for watermarking induced distortion	16
T	Watermark detection threshold; if $\gamma > T$, the watermark is detected	14
t	Test statistic	78
$\mathfrak{t}_i$	Texture score for the coefficient x_i	122
T_{NC}	Watermark detection threshold for normalized correlation	94
T_θ	Angle based watermark detection threshold	94
T^F	Threshold for distortion induced by processing type F	18
t_ν	Student's t distribution with ν degrees of freedom	80
$t_{x(y),\nu}$	The value such that $P(t \sim t_\nu > t_{x(y),\nu}) = \frac{x}{y}$	81
U	Sequence of integer watermark elements	146
u	Integer watermark element	142
u^*	Candidate integer watermark element	150
U^c	Truncated candidate sequence of integer watermark elements	152
u^c	Truncated candidate integer watermark element	152
U^d	Extracted sequence of integer watermark elements	144
u^d	Extracted integer watermark element	145
$u_{i,\kappa}$	The κth bit of the watermark element u_i	217
u^τ	Fake watermark element	201
V	Selected sequence of quantized wavelet coefficients	141
v	Quantized wavelet coefficient	46
$\bar{v}$	Smallest non-negative integer greater than v	150
$V_{i,\kappa}$	The feature sequence used to generate $u_{i,\kappa}$	219
$v_{\mathfrak{f}(i,\,\kappa,x)}$	The xth feature coefficient in the sequence $V_{i,\kappa}$. Also the coefficient at index $\mathfrak{f}(i,\,\kappa,x)$ in the image I	221
$v^{\mathcal{Q}}$	Quality scaled quantized wavelet coefficient	46

W	Watermark: A sequence of watermark elements	59
w	Watermark element	90
X	Width of an image I	44
x	Unquantized wavelet coefficient	45
x^D	Reconstructed (Dequantized) wavelet coefficient, see also $x^{\mathcal{Q}}$	45
X_{i^*}	Number of significant bits received in reconstructing the coefficient magnitude $\lvert v^*_{i^*}\rvert$	154
$x^{\mathcal{Q}}$	Reconstructed (dequantized) wavelet coefficient, after quality scaling	47
$X[r]$	Width of the wavelet domain region containing resolution layers 0 through r	44
$X[r,o]$	Width of the wavelet domain region containing subband (r,o)	148
x, y	x and y coordinate indices	41
xcb	Codeblock width exponent	47
Y	Height of an image I	44
$Y[r]$	Height of the wavelet domain region containing resolution layers 0 through r	44
$Y[r,o]$	Height of the wavelet domain region containing subband (r,o)	148
ycb	Codeblock height exponent	47
Z	A watermark sequence that is independent of W	90
$\hat{Z}$	Unit vector in the direction of Z	91
Z_b	Number of all-zero most significant bit planes in codeblock b	48
z	A watermark element that is independent of w	90
α	Watermark embedding strength	13
α	Significance level (probability of type-II error)	79
Γ	Compression system parameters, e.g. compression ratio, quantization step size	29
γ	Watermark detection statistic, $\gamma(I^*, I, sk_d)$	14
Δ_s	Quantization step size for subband s	45

Δ	Difference between ideal and extracted numbers of watermark elements, across all layers	60
δ	Minimum 'substantial' difference in the algorithm performance	82
ϵ^l	Number of watermark elements extracted from layer l	60
η	Number of coefficients in each feature sequence $V_{i,\kappa}$	219
θ	Angle between two watermarks (considered as vectors)	94
ι^l	Ideal number of watermark elements in layer l	59
κ	Bit position, $\kappa = 0$ for the least significant bit	37
κ	Bit plane index. For the LSB, $\kappa = 0$	217
Λ	Watermarking parameters	13
λ_l	Distortion-rate threshold for quality layer l	49
μ_X	Population mean performance for the algorithm X, $\sum_{x \in \Omega_{X\mathrm{u}}} \frac{x}{\lvert\Omega_{X\mathrm{u}}\rvert}$	74
μ_x	Mean of the random variable x	91
ν	Degrees of freedom	81
Ξ	Scaling parameters for scalable compression, e.g. image dimensions	35
σ_x	Population standard deviation of the random variable x	81
σ_x^2	Variance of the random variable x	91
Φ	Number of MSB planes from $I_0^{\mathcal{R}}$ used in hash generation	218
$\Psi_{i,\kappa}$	Feature sequence type; 0 for inter-codeblock, 1 for intra-codeblock	220
Ω_X	Set of measurements of the watermarking algorithm X on $\mathcal{I}$	75
$\Omega_{X\mathrm{u}}$	Set of measurements of the watermarking algorithm X on $\mathcal{I}_\mathrm{u}$	74

Acronyms

YC_bC_r	Colour space composed of luminance and blue and red chrominance	33
AC	Any frequency coefficients other than the lowest	28
BER	Bit error rate	159
CSF	Contrast sensitivity function	115
DC	The lowest frequency coefficient(s)	41
DCT	Discrete cosine transform	33
DWT	Discrete wavelet transform	42
EBCOT	Embedded Block Coding with Optimized Truncation compression	39
EZW	Embedded Zerotree Wavelet compression	39
FFT	Fast Fourier transform	33
HVS	Human visual system	71
ICT	Irreversible component transform	42
JBIG	Compression system by the Joint Bi-level Image Experts Group	48
JPEG2000	Compression system by the Joint Photographic Experts Group, designed to be both resolution and quality scalable	40

KLT	Karhunen-Loeve transform	33
LABR	Least authenticable bit rate; similar to the lowest quality layer	236
LRCP	Layer-resolution-component-precinct; a progression in JPEG2000	50
LSB	Least significant bit	27
LZ	Lempel-Ziv coding	32
LZW	Lempel-Ziv-Welch coding	32
MSE	Mean Squared Error	34
PSNR	Peak signal to noise ratio	71
QIM	Quantization Index Modulation	27
RCT	Reversible component transform	42
RGB	Colour space composed of red, green and blue components	32
S-CIELAB	A spatial extension to CIELAB colour difference measurement	71
SPIHT	Set Partitioning in Hierarchical Trees compression	39
TCQ	Trellis-coded quantization	35
TSVQ	Tree-structured quantization	35
VQ	Vector quantization	34
WHT	Walsh-Hadamard transform	33
YIQ	Colour space composed of luminance and orange and purple chrominance	33
YUV	Colour space composed of luminance and blue and red chrominance	33

Bibliography

[1] M. D. Adams and R. K. Ward, "JasPer: a portable flexible open-source software tool kit for image coding/processing," in *Proc. 2004 IEEE Intl. Conf. Acoustics, Speech, and Signal Processing, (ICASSP'04), Montreal, QC, Canada, May 17–21*, vol. 5, pp. 241–244. Cited on pages 40, 46, 78, 161, and 243.

[2] A. Adelsbach, S. Katzenbeisser, and A.-R. Sadeghi, "A computational model for watermark robustness," in *Information Hiding, 8th Intl. Workshop, IH 2006. Revised Selected Papers, Alexandria, VA, USA, Jul. 10–12, 2006*, ser. Lecture Notes in Computer Science, J. Camenisch, C. S. Collberg, N. F. Johnson, and P. Sallee, Eds., vol. 4437. Springer Berlin / Heidelberg, 2007, pp. 145–160. Cited on page 67.

[3] A. Adelsbach, B. Pfitzmann, and A.-R. Sadeghi, "Proving ownership of digital content," in *Proc. Information Hiding, 3rd Intl. Workshop, IH'99, Dresden, Germany, Sept. 29 – Oct. 1, 1999*, ser. Lecture Notes in Computer Science, A. Pfitzmann, Ed., vol. 1768. Springer Berlin/Heidelberg, 2000, pp. 117–133. Cited on page 6.

[4] N. Ahmed and K. R. Rao, *Orthogonal transforms for digital signal processing*. New York: Springer-Verlag, 1975. Cited on page 33.

[5] A. M. Alattar, K. L. Levy, R. R. Stager, B. G. Rhoads, and E. E. Ellingson, "Digital Watermarking and Fingerprinting Including Synchronization, Layering, Version Control, and Compressed Embedding," *U.S. Patent no. 7020304*, March 28 2006. [Online]. Available: http://www.patentstorm.us/patents/7020304.html Cited on page 275.

[6] H. O. Altun, A. Orsdemir, G. Sharma, and M. F. Bocko, "Optimal spread spectrum watermark embedding via a multistep feasibility formulation," *IEEE Trans. Image Processing*, vol. 18, no. 2, pp. 371–387, Feb. 2009. Cited on page 28.

[7] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Image Processing*, vol. 1, no. 2, pp. 205–220, Apr. 1992. Cited on page 42.

[8] S. Armeni, D. Christodoulakis, I. Kostopoulos, Y. Stamatiou, and M. Xenos, "A transparent watermarking method for color images [online]," in *Proc. 1st IEEE Balkan Conf. Signal Processing, Communications, Circuits, and Systems, Istanbul, Turkey, June 2–3*, June 2000. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.18.2019 Cited on page 75.

[9] M. Barni, F. Bartolini, V. Cappellini, A. Lippi, and A. Piva, "A DWT-based technique for spatio-frequency masking of digital signatures," in *Proc. Security and Watermarking of Multimedia Contents*, ser. Proceedings of SPIE, P. W. Wong and E. J. Delp, Eds., vol. 3657, 1999, pp. 31–39. Cited on pages 118 and 121.

[10] P. S. L. M. Barreto, H. Y. Kim, and V. Rijmen, "Toward a secure public-key blockwise fragile authentication watermarking," in *Proc. 2001 Intl. Conf. Image Processing, (ICIP 2001)*, vol. 2. IEEE, 2001, pp. 494–497. Cited on pages 25, 232, and 234.

[11] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," *IBM Systems Journal*, vol. 35, no. 3-4, pp. 313–336, 1996. Cited on page 26.

[12] M. Bertran, J.-F. Delaigle, and B. Macq, "Some improvements to HVS models for fingerprinting in perceptual decompressors," in *Proc. 2001 Intl. Conf. on Image Processing (ICIP 2001) Thessaloniki, Greece, Oct. 7–10*, vol. 3. IEEE, pp. 1039–1042. Cited on pages 118 and 121.

[13] P. J. Bickel and K. A. Doksum, *Mathematical Statistics: Basic Ideas and Selected Topics*. Oakland, CA. USA: Holden-Day, Inc., 1977, ch. 4. Cited on page 78.

[14] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data," *IEEE Trans. Information Theory*, vol. 44, no. 5, pp. 1897–1905, Sept. 1998. Cited on page 6.

[15] ——, "Collusion-secure fingerprinting for digital data," *IEEE Trans. Information Theory*, vol. 44, no. 5, pp. 1897–1905, sep 1998. Cited on page 24.

[16] J. Brassil, S. Low, N. F. Maxemchuk, and L. O'Gorman, "Hiding information in document images," in *Proc. 29th Annu. Conf. Information Sciences and Systems (CISS 1995), Baltimore, MD, USA, Mar. 17-19*, 1995, pp. 482–489. Cited on page 6.

[17] C. M. Briquet, *Les Filigranes. Dictionnaire historique des marques du papier, dès leur apparition vers 1282 jusqu'en 1600, avec 39 figures dans le texte et 16 112 facsimilés de filigranes.*, 2nd ed. Liepzig: Hiersemann, 1923, see also [134]. Cited on pages 2 and 3.

[18] P. Burt and E. Adelson, "The laplacian pyramid as a compact image code," *IEEE Trans. Communications*, vol. COM-31, no. 4, pp. 532–540, Apr. 1983. Cited on page 37.

[19] C. S. Chan and C. C. Chang, "An efficient image authentication method based on hamming code," *Pattern Recognition*, vol. 40, no. 2, pp. 681–690, Feb. 2007. Cited on page 27.

[20] T. Chang and C.-C. Kuo, "Texture analysis and classification with tree-structured wavelet transform," *IEEE Trans. Image Processing*, vol. 2, no. 4, pp. 429–441, Oct 1993. Cited on page 118.

[21] Y.-K. Chee, "Survey of progressive image transmission methods," *Intl. Journal of Imaging Systems and Technology*, vol. 10, pp. 3–19, Jan. 1999. [Online]. Available: http://www3.interscience.wiley.com/journal/30000920/abstract Cited on page 38.

[22] B. Chen and G. W. Wornell, "An Information-theoretic Approach to the Design of Robust Digital Watermarking Systems," in *Proc. 1999 IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing, (ICASSP '99), Phoenix, AZ, USA, Mar. 15-19*, vol. 4, pp. 2061–2064. Cited on pages 27, 139, and 270.

[23] ——, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," *IEEE Trans. Information Theory*, vol. 47, no. 4, pp. 1423–1443, 2001. Cited on pages 27, 28, and 67.

[24] T. P.-C. Chen and T. Chen, "A framework for optimal blind watermark detection," in *Proc. ACM Multimedia 2001 Workshop Multimedia and Security (MM&Sec 2001), Ottawa, ON, Canada, Sept. 30 – Oct. 5*, pp. 11–14. Cited on page 75.

[25] ——, "Progressive image watermarking," in *Proc. 2000 IEEE Intl. Conf. Multimedia and Expo (ICME 2000). July 30 – Aug 2*, pp. 1025–1028. Cited on pages 9, 10, 28, 61, and 232.

[26] J. C. Chou, S. S. Pradhan, L. El Ghaoui, and K. Ramchandran, "Robust optimization solution to the data hiding problem using distributed source coding principles," in *Image and Video Communications and Processing 2000*, ser. Proceedings of SPIE, B. Vasudev, T. R. Hsing, A. G. Tescher, and R. L. Stevenson, Eds., vol. 3974, Apr. 2000, pp. 270–279. Cited on page 27.

[27] J. W. Cooley and J. W.Tukey, "An algorithm for the machine calculation of complex fourier series," *Mathematics of Computation*, vol. 19, pp. 297–301, 1965. Cited on page 33.

[28] D. Coppersmith, F. C. Mintzer, C. P. Tresser, C. W. Wu, and M. M. Yeung, "Fragile imperceptible digital watermark with privacy control," in *Proc. Security and Watermarking of Multimedia Contents, San Jose, CA, USA, Jan 25–27*, ser. Proceedings of SPIE, P. Wong and E. Delp, Eds., vol. 3657, 1999, pp. 79–84. Cited on pages 6 and 234.

[29] M. Costa, "Writing on dirty paper," *IEEE Trans. Information Theory*, vol. 29, no. 3, pp. 439–441, May 1983. Cited on page 27.

[30] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Processing*, vol. 6, no. 12, pp. 1673–1687, Dec. 1997. Cited on pages 6, 26, 28, 87, 89, 91, 100, 102, 122, 123, and 138.

[31] I. J. Cox and J.-P. M. G. Linnartz, "Public watermarks and resistance to tampering," in *Proc. Intl. Conf. Image Processing, 1997. (ICIP) Washington, DC, USA, Oct. 26–29*, vol. 3, 1997, pp. 3–6. Cited on pages 6 and 24.

[32] I. J. Cox, M. L. Miller, and J. A. Bloom, *Digital Watermarking: principles and practice*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001. Cited on pages 5 and 71.

[33] I. J. Cox, M. L. Miller, and A. L. McKellips, "Watermarking as Communications with Side Information," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1127–1141, 1999. Cited on page 27.

[34] S. Craver, N. Memon, B.-L. Yeo, and M. M. Yeung, "Resolving rightful ownerships with invisible watermarking techniques: Limitations, attacks, and implications," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, pp. 573–586, May 1998. Cited on pages 6, 20, and 25.

[35] W. Dai, "Crypto++: a C++ class library of cryptographic schemes. version 5.4 [Online]," http://www.cryptopp.com/, Last access: 22 May 2008. Cited on page 247.

[36] A. D'Angelo, M. Barni, and N. Merhav, "Stochastic image warping for improved watermark desynchronization [online]," *EURASIP Journal on Information Security*, vol. 2008, article ID 345184, 14 pages. Cited on page 24.

[37] H. Danyali and M. D. Amiri, "A multiresolution robust watermarking approach for scalable wavelet image compression," in *Proc. Advanced Concepts for Intelligent Vision Systems (ACIVS 2008), Juan-les-Pins, France, Oct. 20-24*, ser. Lecture Notes in Computer Science, vol. 5259, 2008, pp. 57–66. Cited on pages 9, 10, 64, and 138.

[38] H. Danyali and A. Mertins, "Highly scalable image compression based on spiht for network applications," in *Proc. 2002 Intl. Conf. on Image Processing. (ICIP 2002), Rochester, NY, USA, Sept. 22–25*, vol. 1, pp. 217–220. Cited on page 8.

[39] I. Daubechies, *Ten Lectures on Wavelets*, ser. CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics, 1992, vol. 61. Cited on page 33.

[40] J. F. Delaigle, C. Devleeschouwer, B. Macq, and L. Langendijk, "Human visual system features enabling watermarking," in *Proc. IEEE Intl. Conf. Multimedia and Expo. ICME '02. Lausanne, Switzerland, August 26-29*, vol. 2, 2002, pp. 489–492. Cited on page 121.

[41] J. Dittmann, D. Megías, A. Lang, and J. Herrera-Joancomartí, "Theoretical framework for a practical evaluation and comparison of audio watermarking schemes in the triangle of robustness, transparency and capacity," *Trans. Data Hiding and Multimedia Security I*, vol. 4300, pp. 1–40, 2006. Cited on pages 68 and 75.

[42] R. Dugad, K. Ratakonda, and N. Ahuja, "A new wavelet-based scheme for watermarking images," in *Proc. 1998 Intl. Conf. on Image Processing. ICIP'98. Chicago, IL, USA, Oct. 4-7.* IEEE, pp. 419–423. Cited on pages 87 and 101.

[43] J. J. Eggers and B. Girod, "Blind watermarking applied to image authentication," in *Proc. 2001 IEEE Intl. Conf. Acoustics, Speech, and Signal Processing (ICASSP'01). Salt Lake City, UT, USA, May 7-11*, vol. 3, pp. 1977–1980. Cited on page 233.

[44] J. J. Eggers, J. Su, and B. Girod, "A blind watermarking scheme based on structured codebooks [Online]," *IEE Seminar on Secure Images and Image Authentication (Ref. No. 2000/039)*, pp. 4/1–4/21, 2000. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=848144 Cited on pages 27 and 28.

[45] Y.-C. Fan and H.-W. Tsao, "A dual pyramid watermarking for jpeg-2000," *Intl. J. High Performance Computing and Networking*, vol. 5, no. 1, pp. 84–96, Nov. 2007. Cited on page 237.

[46] R. M. Fano, "The transmission of information," *Tech. Rep. no. 65, The Research Laboratory of Electronics*, March 1949, quoted in [165]. Cited on page 31.

[47] N. Fatemi-Ghomi, "Performance measures for wavelet-based segmentation algorithms," Ph.D. dissertation, University of Surrey, U.K., 1997. Cited on page 118.

[48] C. Fei, D. Kundur, and R. H. Kwong, "Analysis and design of secure watermark-based authentication systems," *IEEE Trans. Information Forensics and Security*, vol. 1, no. 1, pp. 43–55, Mar. 2006. Cited on page 242.

[49] ——, "A hypothesis testing approach to semi-fragile watermark-based authentication," *IEEE Trans. Information Forensics and Security*, vol. 4, no. 2, pp. 179–192, June 2009. Cited on page 242.

[50] D. J. Fleet and D. J. Heeger, "Embedding invisible information in color images," in *Proc. Intl. Conf. on Image Processing, 1997.(ICIP 1997), Santa Barbara, CA, USA, Oct. 26–29*, vol. 1, pp. 532–535. Cited on page 101.

[51] E. Franz, A. Jerichow, S. Möller, A. Pfitzmann, and I. Stierand, "Computer based steganography: How it works and why therefore any restrictions on cryptography are nonsense, at best," in *Proc. 1st Intl. Workshop on Information Hiding. (IH 1996). Cambridge, U.K., May 30 – June 1*, ser. Lecture Notes in Computer Science, vol. 1174, 1996, pp. 7–22. Cited on pages 6 and 27.

[52] J. Fridrich, "Security of fragile authentication watermarks with localization," in *Proc. Security and Watermarking of Multimedia Contents IV*, ser. Proceedings of SPIE, E. J. Delp and P. W. Wong, Eds., vol. 4675, Apr. 2002, pp. 691–700. Cited on pages 6 and 235.

[53] J. Fridrich and M. Goljan, "Comparing robustness of watermarking techniques," in *Proc. Security and Watermarking of Multimedia Contents*, ser. Proceedings of SPIE, P. W. Wong and E. J. Delp, Eds., vol. 3657, 1999, pp. 214–225. Cited on pages 67 and 75.

[54] J. Fridrich, M. Goljan, and N. Memon, "Cryptanalysis of the Yeung-Mintzer fragile watermarking technique," *Journal of Electronic Imaging*, vol. 11, no. 2, pp. 262–274, April 2002. Cited on pages 25 and 206.

[55] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Boston, MA, USA: Kluwer Academic Publishers, 1992. Cited on page 35.

[56] Gimp-Savvy.com, "Community-indexed photo archive," Database of images sourced from the National Aeronautics and Space Administration (NASA), the National Oceanic and Atmospheric Administration (NOAA) and the U.S. Fish and Wildlife Service (FWS). Last access: 05 Jun. 2008. [Online]. Available: http://gimp-savvy.com/PHOTO-ARCHIVE/index.html Cited on pages 272 and 317.

[57] B. Girod, "The information theoretical significance of spatial and temporal masking in video signals," in *Human Vision, Visual Processing and Digital Display*, ser. Proceedings of SPIE, B. E. Rogowitz, Ed., vol. 1077, Aug. 1989, pp. 178–187. Cited on page 121.

[58] R. Gold, "Optimal binary sequences for spread spectrum multiplexing," *IEEE Trans. Information Theory*, vol. IT-13, no. 4, pp. 619–621, Oct. 1967. Cited on page 28.

[59] R. C. Gonzalez and R. E.Woods, *Digital Image Processing*, 3rd ed. Prentice Hall, 2007. Cited on page 33.

[60] A. S. Greenberg, "The ancient lineage of trade-marks," *Journal of the Patent Office Society*, vol. 33, pp. 876–887, 1951. Cited on page 2.

[61] H. Guo and N. Georganas, "Multiresolution image watermarking scheme in the spectrum domain," in *Proc. Cananadian Conf. Electrical and Computer Engineering, 2002. IEEE CCECE 2002.*, vol. 2, Winnipeg, MB, Canada, May 2002, pp. 873–878. Cited on pages 10, 63, and 87.

[62] C. Halope, A. Barthez, and J. Menez, "Watermarked plastic support," *U.S. Patent no. 5275870*, January 4 1994. [Online]. Available: http://www.patentstorm.us/patents/5275870/fulltext.html Cited on page 3.

[63] B. T. Hannigan, A. M. Reed, and B. A. Bradley, "Digital watermarking using improved human visual system model," in *Security and Watermarking of Multimedia Contents III*, ser. Proceedings of SPIE, P. Wong and E. Delp, Eds., vol. 4314, Aug. 2001, pp. 468–474. Cited on page 121.

[64] P. Hao and Q. Shi, "Comparative study of color transforms for image coding and derivation of integer reversible color transform," in *Proc. 15th Intl. Conf. Pattern Recognition, 2000. (ICPR'00), Barcelona, Spain, Sept. 3–8*, vol. 3, pp. 224–227. Cited on page 33.

[65] R. M. Haralick, "Statistical and structural approaches to texture," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 786–804, May 1979. Cited on page 118.

[66] F. Hartung and B. Girod, "Watermarking of uncompressed and compressed video," *Signal Processing*, vol. 66, pp. 283–301, May 1998. Cited on pages 26, 62, and 87.

[67] K. Hayat, W. Puech, and G. Gesquiere, "Scalable data hiding for online textured 3d terrain visualization," in *Proc. 2008 IEEE Intl. Conf. Multimedia and Expo (ICME 2008), Hannover, Germany, June 23–26*, 2008, pp. 217–220. Cited on page 275.

[68] E. F. Hembrooke, "Identification of sound and like signals," *U.S. Patent no. 3004104, October 10*, 1961. [Online]. Available: http://www.google.com/patents/about?id=PdVTAAAAEBAJ Cited on pages 4 and 5.

[69] A. Herrigel, S. V. Voloshynovskiy, and Y. B. Rystar, "Watermark template attack," in *Proceedings of SPIE, Electronic Imaging, Security and Watermarking of Multimedia Contents III*, P. W. Wong and E. J. Delp, Eds., vol. 4314, San Jose, CA, USA, Jan. 2001. Cited on page 24.

[70] C. K. Ho and C.-T. Li, "Semi-fragile watermarking scheme for authentication of jpeg images," in *Proc. Intl. Conf. Information Technology: Coding and Computing, ITCC 2004. Las Vegas, NV, USA, Apr. 5–7*, vol. 1, 2004, pp. 7–11. Cited on pages 10 and 236.

[71] M. Hollander and D. A. Wolfe, *Nonparametric Statistical Methods*. New York: John Wiley & Sons, Inc, 1973. Cited on page 83.

[72] M. Holliman and N. Memon, "Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes," *IEEE Trans. Image Processing*, vol. 9, no. 3, pp. 432–441, Mar. 2000. Cited on pages 25, 159, 198, and 199.

[73] H.-C. Huang, J.-S. Pan, Y.-H. Huang, F.-H. Wang, and K.-C. Huang, "Progressive watermarking techniques using genetic algorithms," *Circuits, Systems and Signal Processing*, vol. 26, no. 5, pp. 671–687, Oct. 2007. Cited on page 64.

[74] D. A. Huffman, "A method for the construction of minimum-redundancy codes," in *Proc. Institute of Radio Engineers (IRE)*, vol. 40, Sept. 1952, pp. 1098–1101. Cited on page 32.

[75] D. Hunter, *Papermaking: The History and Technique of an Ancient Craft*, 2nd ed. London, U.K.: Cresset Press, 1957. Cited on pages 2, 3, and 4.

[76] ISO TC 42, *Photography – Psychophysical experimental methods for estimating image quality*, ISO Std. 20 462, Nov. 2005. [Online]. Available: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=38330 Cited on page 70.

[77] ISO/IEC JTC 1/SC 29/WG 1, "ISO/IEC 10918-1 — ITU-T Rec. T.81: Information technology — Digital compression and coding of continuous-tone still images — requirements and guidelines," 1993. [Online]. Available: http://www.w3.org/Graphics/JPEG/itu-t81.pdf Cited on page 37.

[78] ISO/IEC JTC 1/SC 29/WG 1, "ISO/IEC FCD 15444-1 — ITU-T Rec. T.800 : Information technology – JPEG2000 image coding system: Core coding system [WG 1 N1646]," March 2000. [Online]. Available: http://www.jpeg.org/jpeg2000 Cited on pages 40, 45, 47, and 479.

[79] ISO/IEC JTC 1/SC 29/WG 1, "JPEG2000 requirements and profiles version 6.3[WG 1 N1803] [online]," July 2000. [Online]. Available: http://www.jpeg.org/public/wg1n1803.pdf Cited on page 40.

[80] ——, "ISO/IEC 14492-1 — ITU-T Rec. T.88: Information technology – Lossy/lossless coding of bi-level images," 2001. Cited on pages 32 and 48.

[81] ——, "ISO/IEC 15444-1 — ITU-T Rec. T.800 : Information technology – JPEG2000 image coding system: Core coding system," 2004. Cited on pages 8 and 40.

[82] K. Jack, *Video Demystified: A Handbook for the Digital Engineer*, 4th ed. Oxford, U.K.: Newnes, 2005. Cited on page 33.

[83] A. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations," *IEEE Trans. Image Processing*, vol. 1, no. 1, pp. 18–30, Jan. 1992. Cited on page 33.

[84] S. Joo, Y. Suh, J. Shin, H. Kikuchi, and S.-J. Cho, "A new robust watermark embedding into wavelet DC components," *ETRI Journal*, vol. 24, pp. 401–404, Oct. 2002. Cited on page 28.

[85] A. Kent, H. Lancour, and J. E. Daly, Eds., *Encyclopedia of Library and Information Science*. New York: Marcel Decker Inc., 1977, vol. 21, pp. 355–359. Cited on page 2.

[86] H. C. Kim and E. J. Delp, "A reliability engineering approach to digital watermark evaluation," in *Proc. Security, Steganography, and Watermarking of Multimedia Contents VIII*, ser. Proceedings of SPIE, P. W. Wong and E. J. Delp, Eds., Jan. 2006, pp. 635–646. Cited on pages 68, 73, and 75.

[87] H. C. Kim, H. Ogunley, O. Guitart, and E. J. Delp, "The watermark evaluation testbed," in *Proc. Security and Watermarking of Multimedia Contents*, ser. Proceedings of SPIE, P. W. Wong and E. J. Delp, Eds., Jan. 2004, pp. 236–247. Cited on pages 68, 75, and 272.

[88] J. R. Kim and Y. S. Moon, "A robust wavelet-based digital watermarking using level-adaptive thresholding," in *Proc. 1999 Intl. Conf. Image Processing (ICIP 99). Kobe, Japan, Oct. 24–28*, vol. 2. IEEE, pp. 226–230. Cited on page 101.

[89] K. C. Knowlton, "Progressive Image Transmission," *U.S. Patent no. 4222076*, September 9 1980. [Online]. Available: http://www.freepatentsonline.com/4222076.html Cited on pages 7 and 37.

[90] E. Koch and J. Zhao, "Towards robust and hidden image copyright labeling," in *Proc. 1995 IEEE Workshop on Nonlinear Signal and Image Processing. Halkidiki, Greece, Jun. 20–22*, June 1995, pp. 452–455. Cited on pages 6 and 26.

[91] N. Komatsu and H. Tominaga, "Authentication system using concealed image in telematics," *Memoirs of the School of Science and Engineering, Waseda University*, vol. 52, pp. 45–60, 1988. Cited on pages 5 and 6.

[92] O. Koval, S. Voloshynovkiy, F. Perez-Gonzales, F. Deguillaume, and T. Pun, "Spread spectrum watermarking for real images: is it everything so hopeless?" in *Proc. 12th European Signal Processing Conference. Vienna, Austria, Sept. 6–10*, 2004. Cited on page 28.

[93] D. Kundur and D. Hatzinakos, "Digital watermarking using multiresolution wavelet decomposition," in *Proc. 1998 IEEE Intl. Conf. Acoustic, Speech and Signal Processing (ICASP 1998). Seattle, WA, USA, May 12-15*, vol. 5, pp. 2969–2972. Cited on page 26.

[94] C. Kurak and J. McHugh, "A cautionary note on image downgrading," in *Proc. 8th Annu. Computer Security Applications Conference. San Antonio, TX, USA, Nov. 30 – Dec. 9*, 1992, pp. 153–159. Cited on pages 6, 26, and 27.

[95] M. Kutter and F. A. P. Petitcolas, "A fair benchmark for image watermarking systems," in *Proc. Security and Watermarking of Multimedia Contents*, ser. Proceedings of SPIE, P. W. Wong and E. J. Delp, Eds., vol. 3657, Jan. 1999, pp. 226–239. Cited on pages 24, 67, and 75.

[96] M. Kutter, S. V. Voloshynovskiy, and A. Herrigel, "Watermark copy attack," in *Proc. SPIE Security and Watermarking of Multimedia Contents II*, P. W. Wong and E. J. Delp, Eds., vol. 3971, Jan. 2000, pp. 371–380. Cited on page 25.

[97] M. Kutter, S. K. Bhattacharjee, and T. Ebrahimi, "Towards second generation watermarking schemes," in *Proc. 1999 Intl. Conf. Image Processing (ICIP 99). Kobe, Japan, Oct. 24–28*, vol. 1. IEEE, pp. 320–323. Cited on page 101.

[98] E. Landais, "The marking of livestock in traditional pastoral societies," *Traceability of animals and animal products, Scientific and Technical Review*, vol. 20, pp. 463–479, 2001. [Online]. Available: http://www.oie.int/boutique/extrait/landaisa.pdf Cited on pages 1 and 303.

[99] D. Le Gall and A. Tabatabai, "Sub-band coding of digital images using symmetric short kernel filters and arithmetic coding techniques," in *Proc. 1988 Intl. Conf. on Acoustics, Speech, and Signal Processing, (ICASSP-88). New York, NY, USA, Apr. 11-14*, vol. 2, pp. 761–764. Cited on page 42.

[100] W. Lee, *Experimental Design and Analysis*. San Fransisco, CA, USA: W. H. Freeman and Company, 1975. Cited on page 83.

[101] G. E. Legge and J. M. Foley, "Contrast masking in human vision," *J. Opt. Soc. of Am. A*, vol. 70, no. 12, pp. 1458–1471, 1980. Cited on pages 114 and 121.

[102] A. S. Lewis and G. Knowles, "Image compression using the 2-D wavelet transform," *IEEE Trans. Image Processing*, vol. 1, no. 2, pp. 244–250, Apr 1992. Cited on page 118.

[103] C.-T. Li and H. Si, "Wavelet-based fragile watermarking scheme for image authentication," *J. Electronic Imaging*, vol. 16, no. 1, pp. 013 009–1 – 013 009–9, Mar. 2007. Cited on page 235.

[104] X. Li, G. Harbottle, J. Zhang, and C. Wang, "The earliest writing? sign use in the seventh millennium BC at Jiahu, Henan Province, China," *Antiquity*, vol. 77, pp. 31–44, March 2003. Cited on page 1.

[105] Z. Li, Q. B. Sun, Y. Lian, and R. S. Yu, "A scalable watermarking scheme for the scalable audio coder," in *Proc. 2005 IEEE Intl. Conf. Communications, ICC2005. Seoul, Korea, May 16–20*, vol. 2, pp. 1341–1346. Cited on pages 9, 64, and 275.

[106] C.-H. Lin and W.-S. Hsieh, *Semi-fragile Authentication Method for Robust to JPEG, JPEG2000 Compressed and Scaled Images*, ser. Studies in Computational Intelligence. Springer Berlin/Heidelberg, 2009, vol. 227/2009. Cited on page 242.

[107] C.-Y. Lin and S.-F. Chang, "Semifragile watermarking for authenticating JPEG visual content," in *Proc. SPIE Security and Watermarking of Multimedia Contents II*, P. W. Wong and E. J. Delp, Eds., vol. 3971, Jan. 2000, pp. 140–151. Cited on pages 10 and 237.

[108] ——, "SARI: Self-authentication-and-recovery image watermarking system," *ACM Multimedia*, vol. 4518, Oct. 2001. Cited on page 242.

[109] E. T. Lin and E. J. Delp, "A review of fragile image watermarks," in *Proc. Multimedia and Security Workshop at ACM Multimedia'99. Orlando, FL, USA, Oct. 30-31)*, J. Dittmann, K. Nahrstedt, and P. Wohmacher, Eds., 1999, pp. 25–29. Cited on page 159.

[110] E. T. Lin, C. Podilchuk, T. Kalker, and E. J. Delp, "Streaming video and rate scalable compression: What are the challenges for watermarking?" in *Proc. Security and Watermarking of Multimedia Contents III*, ser. Proceedings of SPIE, P. W. Wong and E. J. Delp, Eds., vol. 4314, San Jose, CA, USA, Aug. 2001, pp. 116–127. Cited on pages 9, 53, and 63.

[111] E. T. Lin, C. I. Podilchuk, and E. J. Delp, "Detection of image alterations using semi-fragile watermarks," in *Security and Watermarking of Multimedia Contents II*, ser. Proceedings of SPIE, vol. 3971, May 2000, pp. 152–163. Cited on page 233.

[112] W. S. Lin, H. V. Zhao, and K. J. R. Liu, "Scalable multimedia fingerprinting forensics with side information," in *Proc. 2006 IEEE Intl. Conf. Image Processing, (ICIP 2006), Atlanta, GA, USA, Oct. 8–11*, Oct. 2006, pp. 2293–2296. Cited on page 64.

[113] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Information Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982. Cited on page 34.

[114] W. Lu, R. Safavi-Naini, T. Uehara, and W. Li, "A scalable and oblivious digital watermarking for images," in *Proc. 2004 7th. Intl. Conf. on Signal Processing, ICSP'04, Bejing, China, Aug. 31 – Sept. 4*, vol. 3, pp. 2338–2341. Cited on pages 10, 63, 138, and 232.

[115] B. Macq, J. Dittmann, and E. J. Delp, "Benchmarking of image watermarking algorithms for digital rights management," *Proceedings of the IEEE*, vol. 92, no. 6, pp. 971–984, June 2004. Cited on page 68.

[116] K. Maeno, Q. Sun, S.-F. Chang, and M. Suto, "New semi-fragile image authentication watermarking techniques using random bias and nonuniform quantization," *IEEE Trans. Multimedia*, vol. 8, no. 1, pp. 32–45, Feb. 2006. Cited on pages 9, 10, and 237.

[117] B. Mak and E. Barnard., "Phone clustering using bhattacharyya distance," in *Proc. 4th Intl. Conf. Spoken Language, ICLSP 96. Philadelphia, PA, USA, Oct. 3–6*, vol. 4. IEEE, 1996, pp. 2005–2008. Cited on page 359.

[118] S. G. Mallat, "Multifrequency channel decompositions of images and wavelet models," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 37, no. 12, pp. 2091–2110, Dec. 1989. Cited on page 87.

[119] ——, "A theory for multiresolution signal decomposition : The wavelet representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, July 1989. Cited on page 33.

[120] M. W. Marcellin and T. R. Fischer, "Trellis coded quantization of memoryless and gauss-markov sources," *IEEE Trans. Communications*, vol. 38, no. 1, pp. 82–93, Jan. 1990. Cited on page 35.

[121] B. Mathon, P. Bas, and F. Cayre, "Practical performance analysis of secure modulations for WOA spread-spectrum based image watermarking," in *Proc. 9th Workshop on Multimedia & Security (MM& Sec '07), Dallas, TX, USA, Sept. 20-21.* New York, NY, USA: ACM Press, 2007, pp. 237–244. Cited on page 28.

[122] J. Max, "Quantizing for minimum distortion," *IRE Trans. Information Theory*, vol. 6, no. 1, pp. 7–12, Mar. 1960. Cited on page 34.

[123] P. Meerwald and A. Uhl, "Toward robust watermarking of scalable video," in *Proc. Security, Forensics, Steganography, and Watermarking of Multimedia Contents X, San Jose, C.A., USA, Jan. 26–31*, ser. Proceedings of SPIE, E. J. Delp, P. W. Wong, J. Dittmann, and N. D. Memon, Eds., vol. 6819, Jan. 2008. Cited on pages 10, 28, 64, 138, 233, and 275.

[124] ——, "Scalability evaluation of blind spread-spectrum image watermarking," in *Proc. 7th Intl. Workshop on Digital Watermarking (IWDW 2008), Busan, Korea, Nov. 10-12 2008*, ser. Lecture Notes in Computer Science, vol. 5450. Springer Berlin/Heidelberg, 2009, pp. 61–75. Cited on page 64.

[125] Y. Meyer, *Ondelettes et Opérateurs I*, ser. Actualities Math. Hermann, Paris, 1990, vol. 37, translated by D.H. Salinger, Wavelets and Operators, Cambridge Stud. in Adv. Math., Cambridge University Press, Cambridge, U.K., 1992. Cited on page 33.

[126] M. L. Miller, G. J. Dorr, and I. J. Cox, "Dirty-paper trellis codes for watermarking," in *Proc. 2002 Intl. Conf. Image Processing, (ICIP 2002), Rochester, NY, USA, Sept. 22–25*, vol. 2, pp. 129–132. Cited on page 29.

[127] M. L. Miller and J. A. Bloom, "Computing the probability of false watermark detection," in *Proc. Information Hiding, 3rd Intl. Workshop, IH'99, Dresden, Germany,*

Sept. 29 – Oct. 1, 1999, ser. Lecture Notes in Computer Science, A. Pfitzmann, Ed., vol. 1768. Springer Berlin/Heidelberg, 2000, pp. 146–158. Cited on pages 93 and 137.

[128] J. Morlet, G. Arens, E. Fourgeau, and D. Glard, "Wave propagation and sampling theory — Part 1: Complex signal and scattering in multilayered media," *Geophysics*, vol. 47, Feb. 1982. Cited on page 33.

[129] J. Morovic and P.-L. Sun, "Visual differences in colour reproduction and their colorimetric correlates," in *Proc. 10th Color Imaging Conference: Color Science and Engineering Systems, Technologies, Applications, (CIC 2002), Scottsdale, AZ, USA, Nov. 12*. IS&T/SID - The Society for Imaging Science and Technology, 2002, pp. 292–297. Cited on pages 130 and 161.

[130] D. P. Mukherjee, S. Maitra, and S. T. Acton, "Spatial domain digital watermarking of multimedia objects for buyer authentication," *IEEE Trans. Multimedia*, vol. 6, pp. 1–15, Feb. 2004. Cited on page 75.

[131] M. J. Nadenau, "Integration of human colour vision models into high quality image compression," Ph.D. dissertation, École Polytechnique fédérale de Lausanne, Lausanne, Switzerland, 2000. Cited on pages 89, 115, and 137.

[132] I. Ohzawa, "Make your own Campbell-Robson contrast sensitivity chart," Last access: April 2009. [Online]. Available: http://neurovision.berkeley.edu/Demonstrations/VSOC/izumi/CSF/A_JG_RobsonCSFchart.html Cited on page 114.

[133] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 23–50, Nov. 1998. Cited on page 31.

[134] Österreichischen Akademie der Wissenschaften, Kommission für Schrift- und Buchwesen des Mittelalters (Wien) and Laboratoire de Médiévistique Occidentale de Paris, "Briquet online (v.1 — 2009-07-14)," Last access: 25 August 2009. [Online]. Available: http://www.ksbm.oeaw.ac.at/_scripts/php/BR.php?lang=fr Cited on pages 3 and 288.

[135] B. G. Paster, "Trademarks — their early history," *The Trademark Reporter*, vol. 59, pp. 551–572, Jan. 1969. Cited on page 1.

[136] E. Peli, L. Arend, and A. T. Labianca, "Contrast perception across changes in luminance and spatial frequency," *J. Opt. Soc. Am. A*, vol. 13, pp. 1953–1959, 1996. Cited on page 114.

[137] C. Peng, R. H. Deng, Y. Wu, and W. Shao, "A flexible and scalable authentication scheme for jpeg2000 image codestreams," in *Proc. 11th ACM Intl. Conf. Multimedia, (MM'03), Berkeley, CA, USA, Nov. 2-8.* New York, NY, USA: ACM Press, 2003. Cited on page 232.

[138] S. Pereira, J. Ruanaidh, F. Deguillaume, G. Csurka, and T. Pun, "Template based recovery of fourier-based watermarks using log-polar and log-log maps," in *IEEE International Conference on Multimedia Computing and Systems, 1999*, vol. 1, July 1999, pp. 870–874. Cited on page 24.

[139] S. Pereira, S. Voloshynovskiy, M. Madueo, S. Marchand-Maillet, and T. Pun, "Second generation benchmarking and application oriented evaluation," in *Proc. Information Hiding: 4th Intl. Workshop, IH 2001, Pittsburgh, PA, USA, Apr. 25–27*, ser. Lecture Notes in Computer Science, vol. 2137. Springer Berlin/Heidelberg, 2001, pp. 340–353. Cited on pages 67, 71, 160, 162, 167, and 178.

[140] B. Perry, B. MacIntosh, and D. Cushman, "Digimarc MediaBridge: The birth of a consumer product from concept to commercial application," in *Proc. Security and Watermarking of Multimedia Contents IV, San Jose, CA, USA, Jan. 21-24*, ser. Proceedings of SPIE, E. J. Delp and P. W. Wong, Eds., vol. 4675, 2002, pp. 118–123. Cited on page 6.

[141] F. A. P. Petitcolas, "Image database," Photos from this database used in this thesis were courtesy of Robert E. Barber, Karel de Gendre and Éric Labouré. Last access: 20 Feb. 2009. [Online]. Available: http://www.petitcolas.net/fabien/watermarking/image_database Cited on pages 272, 321, 322, 323, and 368.

[142] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Attacks on copyright marking systems," in *Proc. Information Hiding: 2nd Intl. Workshop, IH 1998, Portland, OR, USA, Apr. 14–17.* London, UK: Springer-Verlag, 1998, pp. 218–238. Cited on page 24.

[143] R. Pickholtz, D. Schilling, and L. Milstein, "Theory of spread-spectrum communications–a tutorial [online]," *IEEE Trans. Communications [legacy, pre - 1988]*, vol. 30, no. 5, pp. 855–884, May 1982. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1095533&tag=1 Cited on page 28.

[144] J. R. Pierce, *An introduction to information theory : symbols, signals & noise*, 2nd ed. New York, NY, USA: Dover Publications, Inc., 1980. Cited on page 32.

[145] A. Piva, M. Barni, F. Bartolini, and V. Cappellini, "DCT-based watermark recovering without resorting to the uncorrupted original image," in *Proc. Intl. Conf. Image Processing, (ICIP 1997), Santa Barbara, CA, USA, Oct. 26–29*, vol. 1, 1997, pp. 520–523. Cited on pages 87 and 101.

[146] J. S. Pointer and R. F. Hess, "The contrast sensitivity gradient across the human visual field: with emphasis on the low spatial frequency range." *Vision research*, vol. 29, no. 9, pp. 1133–1151, 1989. Cited on page 114.

[147] D. Potts, "The potter's marks of Tepe Yahya [online]," *Paléorient*, vol. 7, no. 1, pp. 107–122, 1981. [Online]. Available: http://www.persee.fr/web/revues/home/prescript/article/paleo_0153-9345_1981_num_7_1_4290 Cited on page 1.

[148] C. Poynton, *A technical introduction to digital video.* New York, NY, USA: John Wiley & Sons, Jan. 1996. Cited on page 33.

[149] L. Qiao and I. J. Cox, "Using perceptual models to improve fidelity and provide invariance to valumetric scaling for quantization index modulation watermarking," in *Proc. IEEE Intl. Conf. Acoustics, Speech, and Signal Processing, (ICASSP '05). Philadelphia, PA, USA, Mar. 18–23*, vol. 2, 2005, pp. 1–4. Cited on page 75.

[150] T. Randen and J. H. Husøy, "Filtering for texture classification: a comparative study," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 291–310, Apr 1999. Cited on page 118.

[151] J. J. Rissanen, "Generalized kraft inequality and arithmetic coding," *IBM Journal of Research*, vol. 20, pp. 198–203, may 1976. Cited on page 32.

[152] E. S. Rogers, "Some historical matter concerning trademarks," *Michigan Law Review*, vol. 9, pp. 29–43, 1910. Cited on pages 2 and 4.

[153] G. Ruston, "On the origin of trademarks," *The Trademark Reporter*, vol. 45, pp. 127–144, Feb. 1955. Cited on page 1.

[154] S. Rytsar, Y. Voloshynovskiy, F. Ehrler, and P. Thierry, "Interactive segmentation with hidden object based annotations: towards smart media," in *Proc. Storage and Retrieval Methods and Applications for Multimedia, San Jose, CA, USA, Jan. 20–22*, ser. Proceedings of SPIE, M. M. Y. R. W. L. C.-S. Li, Ed., vol. 5307, 2004. Cited on page 6.

[155] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243–250, June 1996. Cited on pages 39 and 233.

[156] A. Sanz, C. Munoz, and N. Garcia, "Approximation quality improvement techniques in progressive image transmission," *IEEE Journal on Selected Areas in Communications*, vol. 2, no. 2, pp. 359–373, Mar. 1984. Cited on pages 7 and 37.

[157] D. V. Sarwate and M. B. Pursley, "Crosscorrelation properties of pseudorandom and related sequences," *Proceedings of the IEEE*, vol. 68, no. 5, pp. 593–619, May 1980. Cited on page 28.

[158] E. Sayrol, J. Vidal, S. Cabanillas, and S. Santamaria, "Optimum watermark detection in color images," in *Proc. 1999 Intl. Conf. Image Processing (ICIP 99). Kobe, Japan, Oct. 24–28*, vol. 2. IEEE, 1999, pp. 231–235. Cited on page 101.

[159] B. Schiele and J. L. Crowley, "Object recognition using multidimensional receptive field histograms," in *Computer Vision – ECCV'96, 4th European Conf. Computer Vision, Cambridge, U.K., Apr. 15–18*, ser. Lecture Notes in Computer Science, vol. 1064. Springer, 1996, pp. 610 – 619. Cited on page 359.

[160] M. Schlauweg and E. Müller, "Content-adaptive semi-fragile image authentication based on jpeg2000 compression," in *16th Intl. Conf. Digital Signal Processing*, July 2009, pp. 1–8. Cited on page 238.

[161] M. Schlauweg, D. Pröfrock, and E. Müller, "Jpeg2000-based secure image authentication," in *Proc. 8th Workshop on Multimedia & Security (MM& Sec '06), Geneva, Switzerland, Sept. 26–27*. New York, NY, USA: ACM Press, 2006, pp. 62–67. Cited on pages 238 and 239.

[162] P. Segond, "Les représentations des bovidés dans les fresques preéhistoriques du tassili-n-ajjer. Quoted in [98]," Ph.D. dissertation, École nationale vétérinaire dAlfort, Maisons-Alfort, 1974. Cited on page 1.

[163] J.-H. Seo and H.-B. Park, "Data protection of multimedia contents using scalable digital watermarking," in *Proc. 4th Annu. ACIS Intl. Conf. Computer and Information Science, (ICIS '05), Jeju Island, South Korea, Jul. 14–16*, Y. K. Yang and K. Akingbehin, Eds. IEEE Computer Society, 2005, pp. 376–380. Cited on page 232.

[164] J. P. Shaffer, "Multiple hypothesis testing," *Annual Review of Psychology*, vol. 46, pp. 561–576, 1995. Cited on page 80.

[165] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423,623–656, Jul. and Oct. 1948. [Online]. Available: http://cm.bell-labs.com/cm/ms/what/shannonday/paper.html Cited on pages 31 and 291.

[166] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3445–3462, Dec. 1993. Cited on page 39.

[167] G. Sharma, W. Wu, and E. N. Dalal, "The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations," *Color Research and Application*, vol. 30, no. 1, pp. 21–30, Feb. 2005. [Online]. Available: http://www.ece.rochester.edu/~gsharma/ciede2000/ Cited on pages 130 and 161.

[168] Y. Q. Shi and H. Sun, *Image and Video Compression for Multimedia Engineering : fundamentals, algorithms and standards*. Boca Raton, FL, USA: CRC Press, Inc., 2000. Cited on page 33.

[169] Signal and Image Processing Institute, University of Southern California, "The USC-SIPI image database," Last access: 05 Jun. 2008. [Online]. Available: http://sipi.usc.edu/database/ Cited on pages 271, 317, and 318.

[170] I. Skiljan, "IrfanView," Last access: 05 September 2007. [Online]. Available: http://www.irfanview.com/ Cited on page 197.

[171] K. R. Sloan, Jr. and S. L. Tanimoto, "Progressive refinement of raster images," *IEEE Trans. Computers*, vol. C-28, no. 11, pp. 871–874, Nov. 1979. Cited on pages 7 and 37.

[172] V. Solachidis, A. Tefas, N. Nikolaidis, S. Tsekeridou, A. Nikolaidis, and I. Pitas, "A benchmarking protocol for watermarking methods," in *Proc. 2001 Intl. Conf. Image Processing, (ICIP 2001), Thessaloniki, Greece, Oct. 7–10*, vol. 3. IEEE, pp. 1023–1026. Cited on page 67.

[173] H. Sonnet, T. Isenberg, J. Dittmann, and T. Strothotte, "Illustration watermarks for vector graphics," in *Proc. 11th Pacific Conf. Computer Graphics and Applications (PG'03), Canmore, AB, Canada, Oct. 8–10*. IEEE, 2003, pp. 73–82. Cited on page 6.

[174] M. Steinder, S. Iren, and P. D. Amer, "Progressively authenticated transmission," in *Proc. IEEE Military Communications Conference, MILCOM 1999, Atlantic City,*

NJ, USA, Oct. 31 – Nov. 3, vol. 1, 1999, pp. 641–645. Cited on pages 9, 10, 62, 232, 233, and 271.

[175] J. K. Su, J. J. Eggers, and B. Girod, "Analysis of digital watermarks subjected to optimum linear filtering and additive noise," *Signal Processing*, vol. 81, no. 6, pp. 1141–1175, June 2001. Cited on page 67.

[176] P.-C. Su and C.-C. J. Kuo, "Steganography in JPEG2000 compressed images," *IEEE Trans. Consumer Electronics*, vol. 49, pp. 824–832, Nov. 2003. Cited on pages 6 and 27.

[177] P.-C. Su, H.-J. M. Wang, and C.-C. J. Kuo, "Digital watermarking on EBCOT compressed images," in *Applications of Digital Image Processing XXII. Denver, CO, USA, Jul. 20–23*, ser. Proc. of SPIE, A. G. Tescher, Ed., vol. 3808, Oct. 1999, pp. 313–324. Cited on pages 9, 10, and 61.

[178] ——, "An integrated approach to image watermarking and jpeg2000 compression," *J. of VLSI Signal Processing*, vol. 27, pp. 35–53, Feb. 2001. Cited on pages 9, 10, 61, and 232.

[179] M. A. Suhail and M. S. Obaidat, "On the digital watermarking in jpeg 2000," in *Proc. 8th IEEE Intl. Conf. Electronics, Circuits and Systems, 2001. ICECS 2001. Malta, Sept. 2–5*, vol. 2, pp. 871–874. Cited on page 26.

[180] Q. Sun and S.-F. Chang, "A secure and robust digital signature scheme for JPEG2000 image authentication," *IEEE Trans. Multimedia*, vol. 7, no. 3, pp. 480–494, June 2005. Cited on page 236.

[181] Q. Sun, S.-F. Chang, K. Maeno, and M. Suto, "A quantitative semi-fragile jpeg2000 image authentication system," in *Proc. 2002 Intl. Conf. Image Processing, (ICIP 2002), Rochester, NY, USA, Sept. 22–25*, vol. 2. IEEE, pp. 921–924. Cited on pages 9, 10, 236, and 269.

[182] D. J. Swift and R. A. Smith, "Spatial frequency masking and weber's law," *Vision Research*, vol. 23, no. 5, pp. 495–505, 1983. Cited on page 114.

[183] W. Szepanski, "A signal theoretic method for creating forgery-proof documents for automatic verification," in *Proc. Carnahan Conf. Crime Countermeasures. Lexington, KY, USA, May. 16–18*, J. Jackson, Ed. Ores Publications, May 1979, pp. 101–109. Cited on page 5.

[184] K. Tanaka, Y. Nakamura, and K. Matsui, "Embedding secret information into a dithered multi-level image," in *Conf. Record, 1990 IEEE Military Communications Conference. MILCOM '90. Monterey, CA, USA, Sept. 30 – Oct. 3*, vol. 1, pp. 216–220. Cited on page 6.

[185] S. L. Tanimoto and T. Pavlidis, "A hierarchical data structure for picture processing," *Computer Graphics and Image Processing*, vol. 4, pp. 104–119, June 1975. Cited on page 7.

[186] B. Tao and B. Dickinson, "Adaptive watermarking in the DCT domain," in *Proc. 1997 IEEE Intl. Conf. Acoustics, Speech, and Signal Processing. ICASSP-97. Munich, Germany, Apr. 21–24*, vol. 4, pp. 2985–2988. Cited on page 121.

[187] D. S. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. Image Processing*, vol. 9, no. 7, pp. 1158–1170, 2000. Cited on pages 8, 39, and 61.

[188] D. S. Taubman and M. W. Marcellin, *JPEG2000 : image compression fundamentals, standards, and practice*. Boston, MA, USA: Kluwer Academic Publishers, 2002. Cited on page 40.

[189] A. Tefas and I. Pitas, "Robust spatial image watermarking using progressive detection," in *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP '01). Salt Lake City, UT, USA, May 7–11*, vol. 3, 2001, pp. 1973–1976. Cited on page 63.

[190] A. Z. Tirkel, G. A. Rankin, R. M. van Schyndel, W. J. Ho, N. R. A. Mee, and C. F. Osborne, "Electronic Water Mark," in *Proc. Digital Image Computing, Technology and Applications (DICTA-93). Sydney, Australia, Dec. 8–10*. Australian Pattern Recognition Society, 1993, pp. 666–673. Cited on pages 26 and 28.

[191] L. Turner, "Digital Data Security System," *Patent no. WO/1989/008915*, September 21 1989. [Online]. Available: http://www.wipo.int/pctdb/en/wo.jsp?IA=GB1989000293&WO=1989PLAY=STATUS Cited on pages 5 and 26.

[192] T. Uehara, R. Safavi-Naini, and P. Ogunbona, "Recovering DC coefficients in block-based DCT," *IEEE Trans. Image Processing*, vol. 15, no. 11, pp. 3592–3596, Nov. 2006. Cited on page 52.

[193] M. Utku Celik, G. Sharma, E. Saber, and A. Murat Tekalp, "Hierarchical watermarking for secure image authentication with localization," *IEEE Trans. Image Processing*, vol. 11, no. 6, pp. 585–595, June 2002. Cited on page 234.

[194] G. Van de Wouwer, P. Scheunders, and D. Van Dyck, "Statistical texture characterization from discrete wavelet representations," *IEEE Trans. Image Processing*, vol. 8, no. 4, pp. 592–598, Apr. 1999. Cited on page 118.

[195] V. Vaurio and E. Frans, "Paper product with watermark and process therefor," *U.S. Patent no. 3085898*, April 16 1963. [Online]. Available: http://www.freepatentsonline.com/3085898.html Cited on page 3.

[196] S. Voloshynovskiy, A. Herrigel, N. Baumgaertner, and T. Pun, "A stochastic approach to content adaptive digital image watermarking," in *Proc. Information Hiding, 3rd Intl. Workshop, IH'99, Dresden, Germany, Sept. 29 – Oct. 1, 1999*, ser. Lecture Notes in Computer Science, A. Pfitzmann, Ed., vol. 1768. Springer Berlin/Heidelberg, 2000, pp. 211–236. Cited on pages 71 and 121.

[197] S. Voloshynovskiy, S. Pereira, A. Herrigel, N. Baumgartner, and T. Pun, "Generalized watermarking attack based on watermark estimation and perceptual remodulation," in *Proc. SPIE Security and Watermarking of Multimedia Contents II*, P. W. Wong and E. J. Delp, Eds., vol. 3971, Jan. 2000, pp. 358–370. Cited on page 24.

[198] S. Walton, "Information authentication for a slippery new age," *Dr. Dobbs Journal*, vol. 20, April 1995. [Online]. Available: http://www.ddj.com/184409531?pgno=18 Cited on pages 6, 26, and 27.

[199] C.-C. Wang and P.-Y. Chen, "Compression-watermarking scheme for mpeg-4 fine granularity scalability video recompression and authentication," *Optical Engineering*, vol. 47, no. 3, p. 037010, 2008. [Online]. Available: http://link.aip.org/link/?JOE/47/037010/1 Cited on page 275.

[200] H.-J. M. Wang and C.-C. J. Kuo, "High fidelity image compression with multithreshold wavelet coding (MTWC)," in *Proc. Applications of Digital Image Procession XX, San Diego, CA, USA, Jul. 30–Aug. 1*, ser. Proceedings of SPIE, A. G. Tescher, Ed., vol. 3164, Oct. 1997, pp. 383–392. Cited on page 61.

[201] ——, "An integrated progressive image coding and watermarking system," in *Proc. 1998 IEEE Conf. Acoustics, Speech and Signal Processing. Seattle, WA, USA, May 12–15*, vol. 6, 1998, pp. 3721– 3724. Cited on pages 9, 10, 61, and 232.

[202] M.-S. Wang and W.-C. Chen, "A majority-voting based watermarking scheme for color image tamper detection and recovery," *Computer Standards & Interfaces*, vol. 29, no. 5, pp. 561–570, 2007. Cited on page 235.

[203] A. B. Watson, R. Borthwick, and M. Taylor, "Image quality and entropy masking," in *Proc. SPIE Conf. Human Vision and Electronic Imaging II*, B. E. Rogowitz and T. N. Pappas, Eds., vol. 3016, June 1997, pp. 2–12. Cited on pages 114, 117, and 121.

[204] A. B. Watson, G. Y. Yang, J. A. Solomon, and J. Villasenor, "Visibility of wavelet quantization noise," *IEEE Trans. Image Processing*, vol. 6, no. 8, pp. 1164–1175, Aug. 1997. Cited on page 71.

[205] T. A. Welch, "A technique for high-performance data compression," *Computer*, vol. 17, no. 6, pp. 8–19, June 1984. Cited on page 32.

[206] R. B. Wolfgang, C. I. Podilchuk, and E. J. Delp, "The effect of matching watermark and compression transforms in compressed color images," in *Proc. 1998 Intl. Conf. Image Processing (ICIP'98). Chicago, IL, USA, Oct. 4–7*, vol. 1, pp. 440–444. Cited on pages 26, 62, and 87.

[207] P. W. Wong, "A public key watermark for image verification and authentication." in *Proc. 1998 Intl. Conf. on Image Processing. ICIP'98. Chicago, IL, USA, Oct. 4-7*, vol. 1. IEEE, pp. 455–459. Cited on pages 6, 27, and 200.

[208] P. W. Wong and N. Memon, "Secret and public key image watermarking schemes for image authentication and ownership verification," *IEEE Trans. Image Processing*, vol. 10, no. 10, pp. 1593–1601, Oct. 2001. Cited on page 235.

[209] H.-C. Wu, N.-I. Wu, C.-S. Tsai, and M.-S. Hwang, "Image steganographic scheme based on pixel-value differencing and lsb replacement methods," *IEE Proceedings - Vision, Image and Signal Processing*, vol. 152, no. 5, pp. 611–615, Oct. 2005. Cited on pages 6, 27, and 238.

[210] X.-G. Xia, C. G. Boncelet, and G. R. Arce, "Wavelet transform based watermark for digital images," *Optics Express*, vol. 3, pp. 497–511, Dec. 1998. [Online]. Available: http://www.opticsinfobase.org/oe/abstract.cfm?uri=oe-3-12-497 Cited on pages 89, 122, and 123.

[211] L. Xie and G. R. Arce, "Joint wavelet compression and authentication watermarking," in *Proc. 1998 Intl. Conf. on Image Processing. ICIP'98. Chicago, IL, USA, Oct. 4–7*, vol. 2. IEEE, pp. 427–431. Cited on page 26.

[212] M. M. Yeung and F. Mintzer, "An invisible watermarking technique for image verification," in *Proc. Intl. Conf. Image Processing, 1997. (ICIP '97). Santa Barbara, CA, USA, Oct. 26–29*, vol. 2, 1997, pp. 680–683. Cited on pages 6 and 200.

[213] X. Zhang and B. Wandell, "A spatial extension of CIELAB for digital color image reproduction," in *SID Symposium Digest of Technical Papers. San Diego, CA, USA, May 12-17*, vol. 27, 1996, pp. 731–734. Cited on pages 71, 130, and 161.

[214] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Trans. Information Theory*, vol. 23, no. 3, pp. 337–343, May 1977. Cited on page 32.

Index

acceptable quality, *see* image quality
attack
  counterfeiting, 15, 25, 192
    collage, 193, 206–207, 221, 259–263
    Holliman-Memon, 198, 221
    mark transfer, 185, 193, 201–203, 223, 255–257
    solutions in literature, 233–239
    tampering, 193, 196, 240, 255
    transplantation, 232, 234, 236–237
  desynchronization, 24
  processing, 15, 17, 74
    geometric, 178–185
    nongeometric, 167–177
    recompression, 166, 250
  protocol, 25
  removal, 23–24
  strength adjustment, 72–74
attacking, 66
  blind, 161, 195
  improved, 244
  spread spectrum, 104, 124, 131
authentication, 22, 213–214
  condfidence, 163
  graceful improvement, 239
  in literature, 232
  scenario, 158–159
  vulnerabilities, 210–211

base layer, 36, 52, 55–56
benchmarking, *see* evaluation
bit error rate, 159
bit plane, 37
  all zero, 48, 154, 226
  coefficient magnitude, 48, 153
  most significant, 48
bit rate, 30
bits
  guard, 43, 153
  missing, 153, 225
blind watermarking, 16–17, 139, 213
broadcast monitoring, 20–21

candidate image, 13–14
candidate watermark, 14, 146
  generation
    blind, 146
    improved, 226
  truncation, 152
capacity, *see* payload
codeblock, 47
coding, 29
  context-adaptive arithmetic, 32, 48
  decorellation, 32
  entropy, 31–32
coefficient, 45
  completely lost, 49, 141, 144, 147

high-magnitude, 101, 141
index, 148–149, 216
partially lost, 49, 144, 147
reconstructed, 45
coefficient selection, 100–112, 137
blind, 141–143, 216
improved, 215
colourspace, 33, 42, 101, 115
`comp`, 103
comparison, 74–85, 122
component, 42
dimensions, 44
compression, 29–35
scalable, *see* scalable compression
techniques, 31–35
compression rate, 30
compression ratio, 30
confidence, 75, 76, 79, 82
contrast sensitivity, 114–117
copy control, 21
copyright protection, 5, 19–20
Cox, 123

data enrichment, 22, 64
decoding, *see* coding
decompression, *see* compression
decorrelation, 32–34
detectability
definition, 52
measurement of, 55–57, 104, 125, 186–188
results
blind, 188
improved, 251
spread spectrum, 107, 133–135
detection, 66
blind, 162, 195
improved, 228, 244
spread spectrum, 104, 131
detection algorithm, 13–15
blind, 157
improved, 230
spread spectrum, 88
detection key, *see* key
detection statistic, 14, 57, 78, 123
device control, 21
distortion measure
embedding, 16, 69–72
objective, 71–72
processing, 18, 72–74
scaling, 55
subjective, 69–70
distortion scalability, *see* quality scalability

EBCOT, 39–40
embedded coding, 38–40
embedding, 66
blind, 161, 194
improved, 243
spread spectrum, 103, 130
embedding algorithm, 13
blind, 142–143, 157
improved, 229
spread spectrum, 88
embedding key, *see* key
embedding parameters, 13
embedding strength, 13, 150
adjustment, 69–72, 103, 105, 130
entropy coding, 31–32
error
hypothesis testing, *see* type-I error; type-II error

watermark detection, *see* false positive; false negative
error rate, 159
evaluation
framework, 66
literature, 66
extracted watermark, 14
extraction
blind, 144–146
improved, 227
spread spectrum, 88
EZW, 38–39

false alarm, *see* false positive
false negative
model, 68
watermark detection, 14–15, 134
false positive
hypothesis testing, *see* type-I error
model, 14, 89–95, 137
effects of scaling on, 95–100
watermark detection, 14–15
feature
magnitude, 223
sign, 223
feature coefficient, 217
feature robustness, 224
feature sequence, 219–225
fine granularity scalability, 38
fingerprinting, 20
fragile, 18
full image, 35

Gaussian model, 91–93
generation
blind, 146–152
improved, 217
spread spectrum, 88
graceful improvement, 125, 239
definition, 52
ideal watermark allocation, 59–60
in imperceptibility, 64
measurement of, 57, 59–60, 105, 188
normalization, 347
perfect, 274
results
blind, 190
improved, 253
spread spectrum, 109, 135–137
granularity, 38

HH, *see* subband, orientation
HL, *see* subband, orientation
human visual system, *see also* masking
adaptation, 113, 122, 123, 137
image quality measurement, 71–72
model, 16
visually lossless, 34
`hvs`, 124
hypersphere model, 93
hypothesis testing, 76, 78, 126
assumptions, 82, 131

image
candidate, *see* candidate image
empty, 58
original, *see* original image
scaled, *see* subimage
image quality, 67–68
acceptable, 18, 52, 55–56, 192, 206
as compression target, 30
improvement in, 58–59
objective measurement, 71–72
subjective measurement, 69–70
images
list of, 317

number of, *see* sample size
imperceptibility, 15–16
coefficient selection, 101
irreversible component transform, 42

JPEG2000, 40–50, 154
effects of scaling, 143

key, 18, 126
detection, 13, 14
embedding, 13, 104, 130
generation, 15
image dependent, 218
sensitivity, 160
blind, 164
improved, 247
keyspace, 15

layer
higher, *see* refinement layer
lowest, *see* base layer
layered coding, 36
least significant bit, 45, 49
watermarking, 27
LH, *see* subband, orientation
LL, *see* subband, orientation
lossless, 29–30, 45, 87
lossy, 29–30, 45, 48
`lum`, 102
`lumnl`, 102

magnitude feature, 223
masking
contrast sensitivity, 114–117
edge, 121
texture, 117–122, 351
Weber's Law, 150
match
exact, 139, 158
blind, 162
improved, 245
message, 13
size, *see* payload
missed detection, *see* false negative

`nohvs`, 124
`nolow`, 102

orientation, *see* subband, orientation
original image, 13, 16

p-value, 79, 80
packet, 50
paired difference, 79, 83, 127, 131
paired t-test, 79, 83, 131, 133
passes, 39, 48, 154
number of, 48
to quality layers, 49
payload, 19
population mean, 74
power, 80, 82, 127
power analysis, 76, 80
precinct, 47
precision, 34, 41
processing, *see* attack, processing
progression, 50
proof of ownership, 19–20
proofs
blind, 377
improved, 487
PSNR, 58, 68, 71

quality, *see* image quality
quality layer, 49
quality scalability, *see also* graceful improvement
compression, 36, 37, 39
effect on false positives, 98

in JPEG2000, 49, 143
watermarking, 54
quantization
compression, 34–35
in JPEG2000, 45–47
watermarking, 139, 142
quantization step size, 34
in JPEG2000, 45
of feature, 224, 236
watermarking, 142, 152, 218
quantization watermarking, 26–28
LSB embedding, 27
quantize-and-replace, 26
quantized coefficient, 45

rate control, 30–31, 49
rate distortion, 30
rate scalability, *see* quality scalability
refinement layer, 36, 53
`res`, 102
resolution layer, 43
resolution layers, number of, 42
resolution scalability
compression, 36–38
effect on false positives, 95
in JPEG2000, 43, 143
watermarking, 54
reversible component transform, 42
robustness, 17–18

S-CIELAB, 71, 130
sample mean, 75
sample size, 75–82, 126
in watermarking literature, 75
scalable compression, *see also* quality scalability, resolution scalability
definition, 35
techniques, 36–40
scalable watermarking, *see also* quality scalability, resolution scalability
definition, 52
literature, 9–10, 60–65
measurement, 54–60
notation, 141
types of, 54
scaling, 35
notation, 56
parameters, 35, 55, 78
quality, 153–155
security, *see* attack
semi-fragile, 18, 158
in literature, 233
sign feature, 223
sign function, 45
sign test, 83, 133
significance
bit plane, 37, 48, 112
in tier-1 coding, 48
perceptual, 101, 106, 107
practical, *see* substantial difference
statistical, *see* type-I error
SNR scalability, *see* quality scalability
spatial domain tampering, *see* attack
spatial domain watermarking, 26
spatial scalability, *see* resolution scalability
SPIHT, 39
spread spectrum, 28–29, 87
steganography, 22–23
subband, 42, 44
orientation, 43, 44
subimage, 35, 55, 58
subimage, resolution, 43
substantial difference, 81, 82, 85, 127
symmetric watermarking, 18–19

t-test
paired, 79, 83, 131, 133
two sample, 349
tamper location, 257, 263
tamper map, 192, 195, 263
tampering, *see* attack
texture, *see* masking
texture score, 118, 122, 351
$T_{\mathcal{F}}$, *see* threshold, distortion, scaling
`thresh`, 103
threshold
coefficient selection, 103
distortion
embedding, 16, 18
processing, 18
scaling, 55
watermark detection, 14, 90, 137
tier-1 coding, 47–50
tier-2 coding, 48
`top`, 102
tradeoff
resolution/quality, 111–114, 135, 137
robustness/imperceptibility, 16, 68
transform
colourspace, 33
decorrelating, 32
wavelet, 26, 33, 42–44
transform domain watermarking, 26
true negative
hypothesis testing, *see* power
watermark detection, 14
true positive
hypothesis testing, *see* confidence
rate, 56
watermark detection, 14
truncation
candidate, 152–153
codestream, 50
two sample t-test, 349
type-I error
correction for multiple comparisons, 80
hypothesis testing, 80, 84, 127
watermarking, *see* false positive
type-II error
hypothesis testing, 80, 127
watermarking, *see* false negative

watermark detection
algorithm, *see* detection algorithm
watermark element
generation
blind, 150
improved, 217
non-existent, 142, 145
watermark elements
ideal number of, 59, 104
number of extracted, 60, 105
watermarking
algorithm, *see* embedding algorithm; detection detection algorithm
applications, 19–23
definition, 13
early digital, 4–6
paper, 2–4
techniques, 26–29
wavelet domain, 87, 198
wavelet transform, 26, 33, 42–44
Weber's Law, 150

Xia, 123

all-zero bit planes, 154
all-zero most significant bit planes, 48
zero-bit, 14, 19

Appendix A

Images used in this Thesis

The following figures contain the images used in the body of thesis. Each caption contains the name or number of the image as used in this thesis, with the name or number of the image as used in the source database in parentheses. This is followed by the image size (in pixels). Note that although the aspect ratio has been maintained, the figures shown here have been scaled to a fixed width for consistency of presentation. The vast majority were obtained from the Gimp-Savvy photo archive [56]; for other remaining images, the source is listed in the caption of the figure.

Figure A.1: Lena, 512×512. From the Signal and Image Processing Institute at the University of Southern California (USC-SIPI) [169]

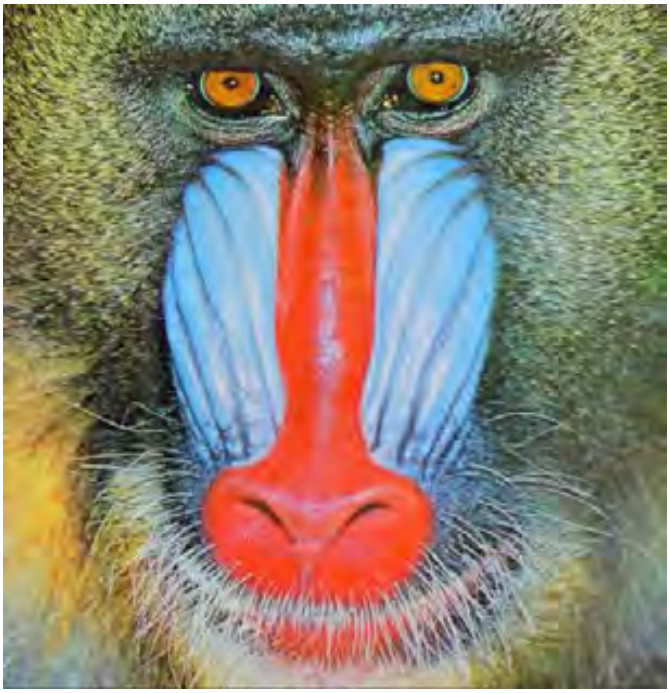


Figure A.2: Mandrill, 512 × 512. From the Signal and Image Processing Institute at the University of Southern California (USC-SIPI) [169]

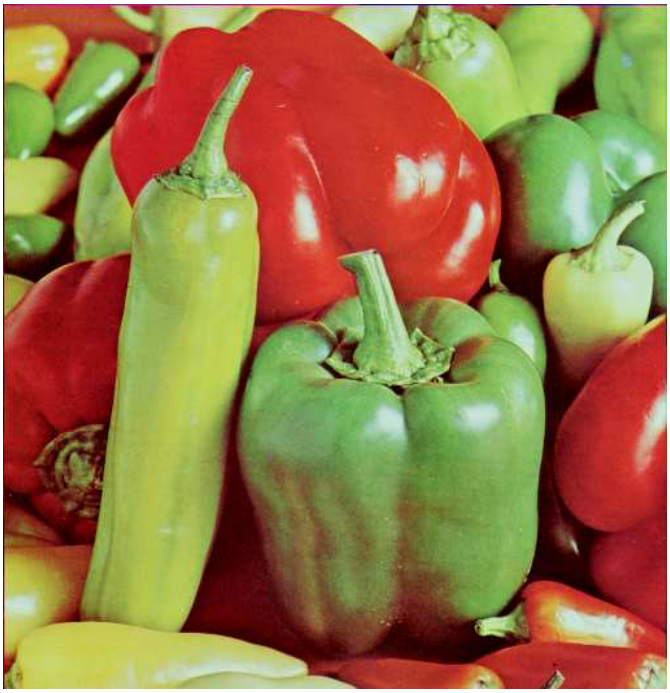


Figure A.3: Peppers 512 × 512. From the Signal and Image Processing Institute at the University of Southern California (USC-SIPI) [169]

Figure A.4: Image 1 (AC94-0273-35_a), 768 × 512.

Figure A.5: Image 2 (AC85-0037-67_a), 768 × 512.

Figure A.6: Image 3 (AC98-0145-11_a), 768 × 512.

Figure A.7: Image 4 (AC96-0232-52_a), 768 × 512.

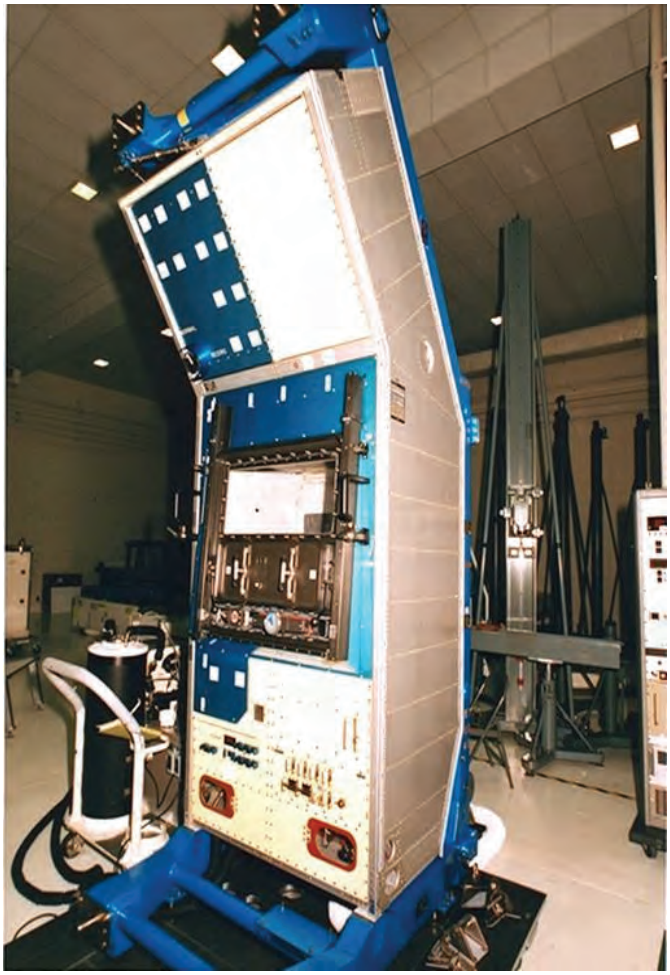


Figure A.8: Image 5 (AC91-0329-6_a), 512 × 768.

Figure A.9: Image 6 (Arctic Hare), 594 × 400. Copyright photo courtesy of Robert E. Barber [141]

Figure A.10: Image 7 (Black Bear), 394×600. Copyright photo courtesy of Robert E. Barber [141]

Figure A.11: Image 8 (Kid), 487 × 703. Copyright photo courtesy of Karel de Gendre [141]

Figure A.12: Image 9 (nerr0787), 700 × 457.

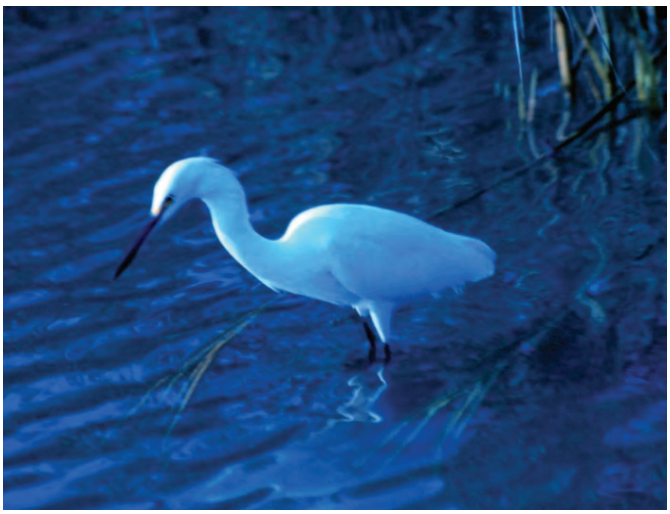


Figure A.13: Image 10 (line1789), 700 × 515.

Figure A.14: Image 11 (line1199), 700 × 457.

Figure A.15: Image 12 (geod0788), 700 × 389.

Figure A.16: Image 13 (mvey0656), 700 × 457.

Figure A.17: Image 14 (fly00013), 700 × 364.

Figure A.18: Image 15 (theb1436), 700 × 458.

Figure A.19: Image 16 (nssl0180), 700 × 463.

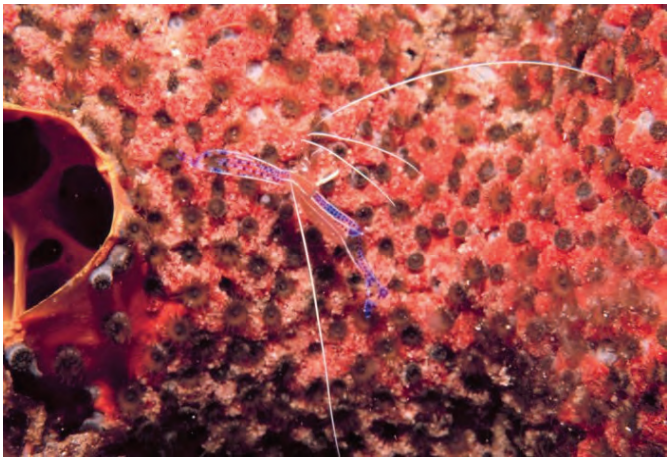


Figure A.20: Image 17 (reef1024), 700 × 462.

Figure A.21: Image 18 (corp1225), 700 × 453.

Figure A.22: Image 19 (nur08509), 700 × 458.

Figure A.23: Image 20 (img04), 537 × 360.

Figure A.24: Image 21 (img 14) 537 × 360.

Figure A.25: Image 22 (img30), 537 × 360.

Figure A.26: Image 23 (000000ef), 658 × 437.

Figure A.27: Image 24 (crn), 576 × 384.

Figure A.28: Image 25 (img28), 537 × 360.

Figure A.29: Image 26 (img36), 360 × 537.

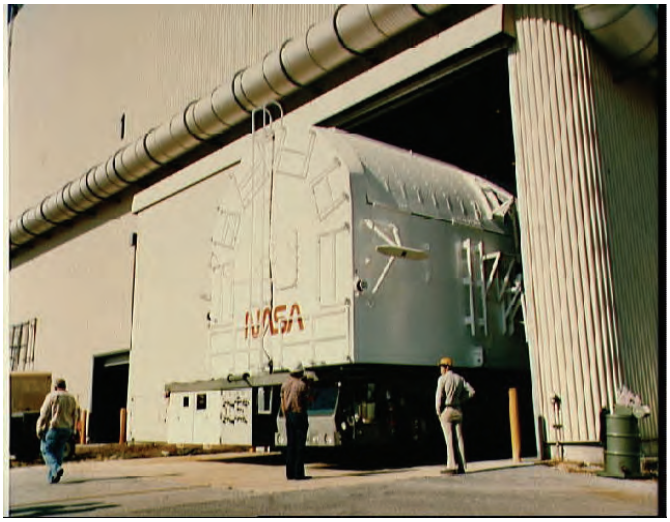


Figure A.30: Image 27 (10061607), 640 × 480.

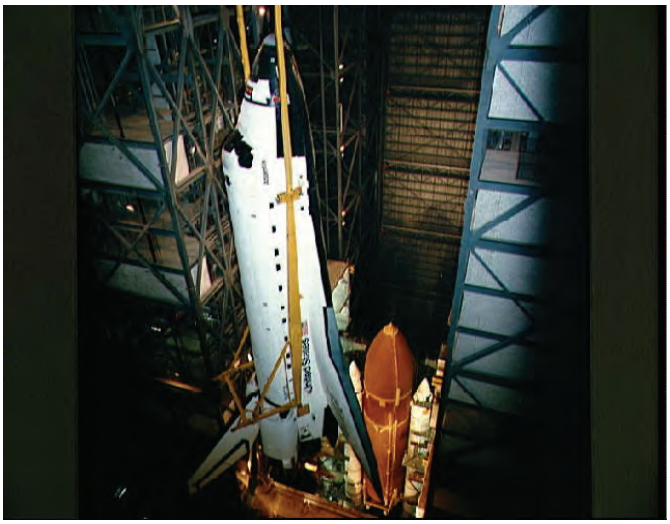


Figure A.31: Image 28 (10063055), 640 × 480.

Figure A.32: Image 29 (10074747), 640 × 480.

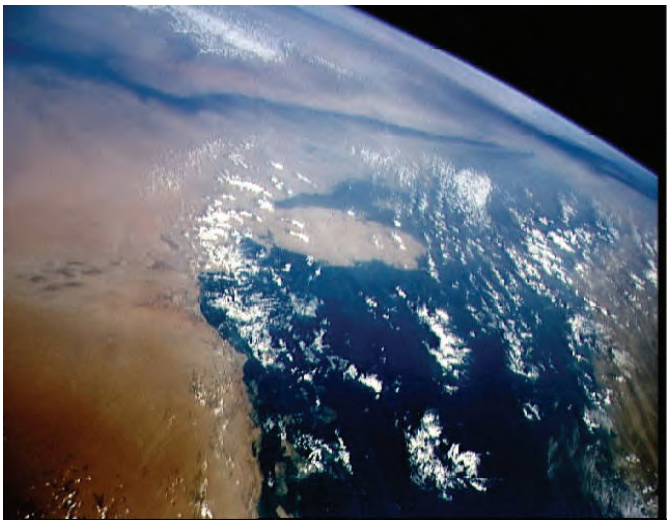


Figure A.33: Image 30 (10064587), 640 × 480.

Figure A.34: Image 31 (10065101), 640 × 480.

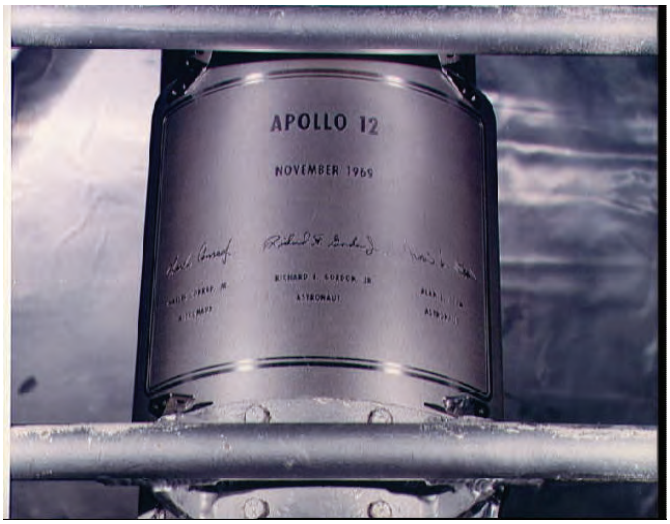


Figure A.35: Image 32 (10075364), 640 × 480.

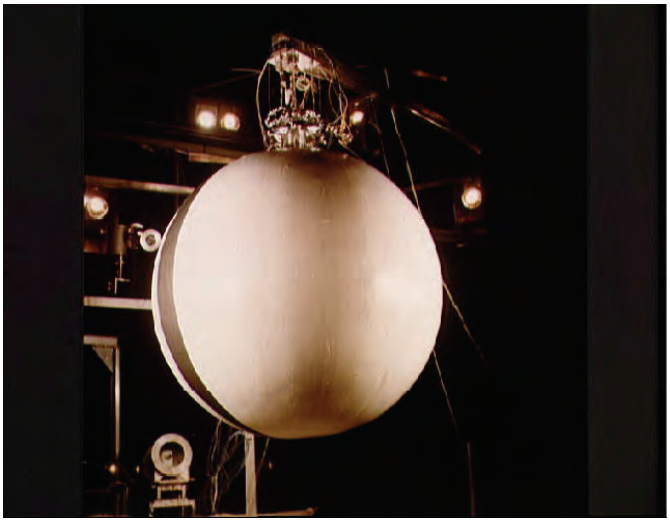


Figure A.36: Image 33 (10061731), 640 × 480.

Figure A.37: Image 34; Greek Isles (10073226), 640 × 480.

Figure A.38: Image 35 (10075505), 640 × 460.

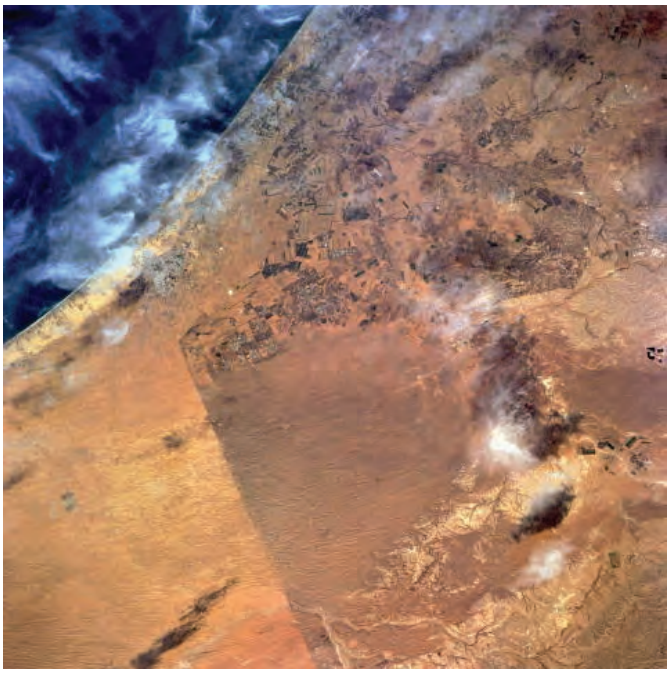


Figure A.39: Image 36 (73708089), 480 × 465.

Figure A.40: Image 37 (wea00597), 700 × 458.

Figure A.41: Image 38 (mvey0159), 700 × 472.

Figure A.42: Image 39 (reef0107), 700 × 466.

Figure A.43: Image 40 (corp1149), 700 × 462.

Figure A.44: Image 41 (000000a5), 459 × 760.

Figure A.45: Image 42 (img40), 537 × 360.

Figure A.46: Image 43 (AC72-2143_a), 512 × 768.

Figure A.47: Image 44 (AC95-0054-7_a), 768 × 512.

Figure A.48: Image 45 (AC96-0232-106_a), 768 × 512.

Figure A.49: Image 46 (nerr0799), 700 × 592.

Figure A.50: Image 47 (sanc0316), 700 × 472.

Figure A.51: Image 48 (geod0098), 700 × 458.

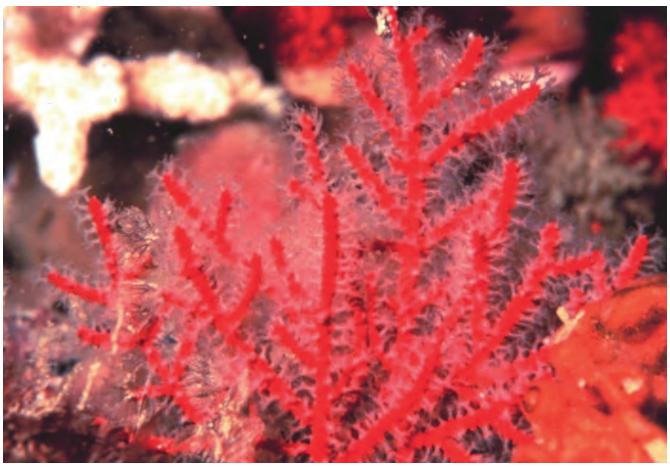


Figure A.52: Image 49 (sanc0516), 700 × 467.

Figure A.53: Image 50 (mvey0518), 700 × 457.

Figure A.54: Image 51 (reef1008), 700 × 470.

Figure A.55: Image 52 (reef0436), 700 × 470.

Figure A.56: Image 53 (corp1565), 700 × 450.

Figure A.57: Image 54 (fly00385), 700 × 475.

Figure A.58: Image 55 (wea02408), 700 × 470.

Figure A.59: Image 56 (line0206), 700 × 462.

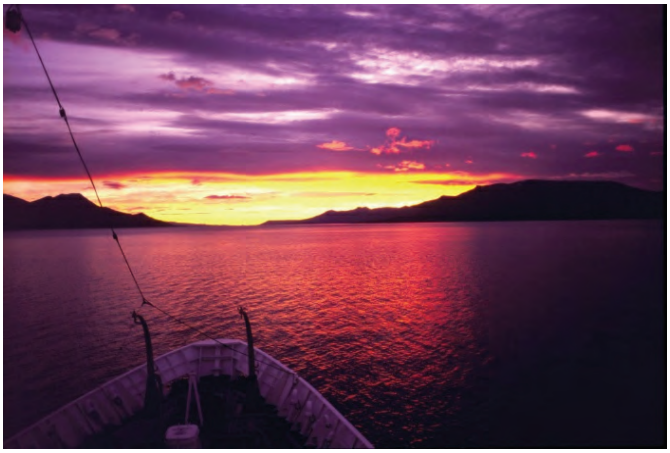


Figure A.60: Image 57 (corp2945), 700 × 454.

Figure A.61: Image 58 (nerr0363), 700 × 463.

Figure A.62: Image 59 (nerr0215), 700 × 460.

Figure A.63: Image 60 (line0301), 700 × 440.

Figure A.64: Image 61 (fish0693), 700 × 311.

Figure A.65: Image 62 (mvey0584), 700 × 458.

Figure A.66: Image 63 (corp1881), 700 × 455.

Figure A.67: Image 64 (line0404), 700 × 461.

Figure A.68: Image 65 (line0157), 700 × 452.

Appendix B

Definition and Evaluation of Scalable Watermarking

B.1 Normalization of Graceful Improvement

In section 3.1.2.2 (page 57), the graceful improvement measure $\mathcal{G}$ was defined using the depature from the ideal Δ, normalized to be between 0 and 1, with $\mathcal{G} = 1$ representing an exact match to the ideal and $\mathcal{G} = 0$ the maximum possible departure from the ideal.

Normalizing graceful improvement requires that we determine the numbers of extracted watermark elements ϵ^l, for each layer $l \in 0, ..., L-1$, that lead to the maximum and minimum values of

$$\Delta = \sum_{l=0}^{L-1} \frac{(\epsilon^l - \iota^l)^2}{\iota^l}, \tag{3.13}$$

given that both the sum of the ideal values across all layers and the sum of the extracted values across all layers equal the total number of watermark elements N:

$$\sum_{l=0}^{L-1} \iota^l = N \tag{B.1}$$

$$\sum_{l=0}^{L-1} \epsilon^l = N. \tag{B.2}$$

Rearranging equations B.1 and B.2 to make ι^{L-1} and ϵ^{L-1}, respectively, the subjects, and substituting into equation 3.13 we obtain:

$$\Delta = \left(\sum_{l=0}^{L-2} \frac{(\epsilon^l - \iota^l)^2}{\iota^l} + \frac{(\sum_{l=0}^{L-2} \iota^l - \sum_{l=0}^{L-2} \epsilon^l)^2}{N - \sum_{l=0}^{L-2} \iota^l} \right) \tag{B.3}$$

We wish to find values of ϵ^l which minimise and maximise Δ. Setting the partial derivative of Δ to 0 at each layer $l \in \{0, ..., K-2\}$ we obtain the stationary points.

$$\frac{\partial \Delta}{\partial \epsilon^l} = 0 \tag{B.4a}$$

$$\frac{\partial}{\partial \epsilon^l}\left(\sum_{l=0}^{L-2} \frac{(\epsilon^l - \iota^l)^2}{\iota^l} + \frac{(\sum_{l=0}^{L-2} \iota^l - \sum_{l=0}^{L-2} \epsilon^l)^2}{N - \sum_{l=0}^{L-2} \iota^l}\right) = 0 \tag{B.4b}$$

$$\left(\frac{2(\epsilon^l - \iota^l)}{\iota^l} - \frac{2(\sum_{k=1}^{K-1} \iota^l - \sum_{k=1}^{K-1} \epsilon^l)}{N - \sum_{k=1}^{K-1} \iota^l}\right) = 0 \tag{B.4c}$$

This gives us L-1 simultaneous equations which can be simplified to

$$\epsilon^l - \frac{\iota^l}{\iota^{L-2}} \epsilon^{L-2} = 0 \qquad l \in \{0, ..., L-3\} \tag{B.4d}$$

and

$$\epsilon^{L-2} - \iota^{L-2} = 0 \qquad l = L-2 \tag{B.4e}$$

and solved to give

$$\epsilon^l = \iota^l \qquad \forall l \in \{0, ..., L-2\}, \tag{B.5}$$

which, using equations B.1 and B.2, gives

$$\begin{aligned} \epsilon^{L-1} &= N - \sum_{l=0}^{L-2} \epsilon^l \\ &= N - \sum_{l=0}^{L-2} \iota^l \\ &= \iota^{L-1}. \end{aligned} \tag{B.6}$$

So our only stationary point occurs when

$$\epsilon^l = \iota^l \qquad \forall l \in \{0, ..., L-1\}.$$

This is clearly the minimum departure from the ideal, as it results in $\Delta = 0$, therefore the maximum departure must occur on the domain boundaries, where ϵ^l is either maximal or minimal for all layers. The minimum number of elements that may be extracted from any layer is 0, and the maximum is N, but, to satisfy equation B.2, ϵ^l may only be maximal for a single layer, call it $\mathfrak{m}$. Additionally, $\mathfrak{m}$ cannot be an empty layer, as it is not possible to extract any elements from an empty layer, thus $\iota^{\mathfrak{m}} > 0$:

$$\epsilon^l = 0 \qquad l \neq \mathfrak{m} \tag{B.7a}$$

$$\epsilon^l = N \qquad l = \mathfrak{m} \tag{B.7b}$$

The maximum departure for an image with layers $0 \leq l < L$ must therefore be

$$\begin{aligned}
\Delta &= \sum_l \frac{(\epsilon^l - \iota^l)^2}{\iota^l} \\
&= \left(\sum_{l \neq \mathfrak{m}} \frac{(0 - \iota^l)^2}{\iota^l} + \frac{(N - \iota^{\mathfrak{m}})^2}{\iota^{\mathfrak{m}}} \right) \\
&= \left(\sum_{l \neq \mathfrak{m}} \iota^l + \frac{N^2}{\iota^{\mathfrak{m}}} - 2N + \iota^{\mathfrak{m}} \right) \\
&= \left(\sum_l \iota^l + \frac{N^2}{\iota^{\mathfrak{m}}} - 2N \right) \\
&= \left(\frac{N^2}{\iota^{\mathfrak{m}}} - N \right) \qquad \text{(eqn. B.1)} \\
&= N \left(\frac{N}{\iota^{\mathfrak{m}}} - 1 \right)
\end{aligned} \tag{B.8}$$

for some $\mathfrak{m}$.

This will be greatest when $\mathfrak{m}$ corresponds to the layer in which $\iota^{\mathfrak{m}}$ is smallest, but $\mathfrak{m}$ must also be non-empty, so the maximum value of Δ occurs when $\mathfrak{m}$ is the smallest non-empty layer

$$0 < \iota^{\mathfrak{m}} \leq \iota^l \qquad \iota^l > 0. \tag{B.9}$$

This also makes intuitive sense, since if our aim is to distribute the watermark evenly amongst the layers according to their contribution to perceptual quality, then the worst possible distribution is to place it entirely in the layer that contributes least to the image.

Since the minimum value of Δ is already 0, normalization is simply a matter of dividing by the maximum value $N\left(\frac{N}{\iota^{\mathfrak{m}}} - 1\right)$, giving

$$\mathcal{G} = 1 - \frac{\Delta}{N(\frac{N}{\iota^{\mathfrak{m}} - 1})}, \tag{3.14}$$

where $\mathfrak{m}$ is the layer such that

$$0 < \iota^{\mathfrak{m}} \leq \iota^l \qquad \iota^l > 0. \tag{B.9}$$

B.2 Two Sample t-test

A two sample t-test is a standard way of comparing the means of two groups of independent samples $\Omega_A = \{\mathfrak{a}_i | 1 \leq i \leq n_A\}$ and $\Omega_B = \{\mathfrak{b}_j | 1 \leq j \leq n_B\}$. Both groups of samples are assumed to be drawn from normally distributed populations with the same standard deviation σ. The t-test tests against $H_0 : \mu_A = \mu_B$. The test statistic is the difference in

sample means divided by the standard error for the estimate of the mean:

$$\begin{aligned} t_2 &= \frac{m_A - m_B}{SE_{m_A - m_B}} \\ &= \frac{m_A - m_B}{\sqrt{\frac{(n_A-1)s_A^2+(n_b-1)s_A^2}{n_A+n_A-2}\left(\frac{1}{n_A}+\frac{1}{n_B}\right)}} \end{aligned} \tag{B.10}$$

where s_A and s_B are the sample standard deviations from groups Ω_A and Ω_B respectively. If H_0 is true this statistic will have a student's t distribution with $\nu = n_A + n_B - 2$ degrees of freedom, with an expected value of zero.

The p-value is the probability p that a sample t drawn from a student's t distribution with ν degrees of freedom is further from zero than t_2.

$$\begin{aligned} p &= P(t \sim t_\nu > |t_2|) + P(t \sim t_\nu < -|t_2|). \\ &= 2 \times P(t \sim t_\nu > |t_2|). \end{aligned} \tag{B.11}$$

If $p < \alpha$, we reject H_0 with confidence $1 - \alpha$ and conclude that there is a significant difference between the underlying populations Ω_{A_u} and Ω_{B_u}.

Comparison between watermarking algorithms can be performed using measurements on the same set of images. In this case, the groups of measurements are no longer independent, and instead consist of matched pairs, $\mathfrak{a}_i$ and $\mathfrak{b}_i$. This allows the use of the more powerful *paired* t-test (section 3.2.4.2.3, page 79).

Appendix C

Development of the Texture Scoring Algorithm

For the texture scoring algorithm of section 4.3 a few desirable properties were identified (section 4.3.1.2, page 117). The texture scoring algorithm should:

1. operate in the wavelet domain, as this allows texture scores to be calculated at the time of embedding without requiring transformation into another domain.

2. only use coefficients from the same resolution as the selected coefficient, to increase computational efficiency and because the major masking effect occurs when the target and the masker have the same frequency,

3. separate textured regions from edged regions as well as smooth regions, because the human visual system is sensitive to changes around edges as well as in smooth regions.

In this section, a texture scoring algorithm is developed in an attempt to satisfy these properties.

C.1 Characterizing Texture in the Wavelet Domain

The initial stage of algorithm development is to consider what characterizes textured regions, as distinct from smooth and edged regions, in terms of wavelet-domain features.

C.1.1 Medium to High Frequency

Texture is essentially a medium to high frequency phemomenon. This can be seen by comparing the original lena image with a scaled version in which only the low frequency data is retained (figure C.1); while many textured regions appear in the original image, its low-frequency counterpart consists only of smooth regions separated by edges.

As a result, the coefficients which occur in the low resolution layers are excluded from the texture scoring process. Texture scoring will only be applied to coefficients x_i in the medium and high resolution layers:

$$r_i \geq \lfloor \frac{R}{3} \rfloor. \tag{C.1}$$

(a) The Lena image

(b) The reconstructed Lena image with resolution layers 2 through 5 removed

Figure C.1: The majority of texture information is contained in the medium to high frequency resolution layers.

C.1.2 High Energy

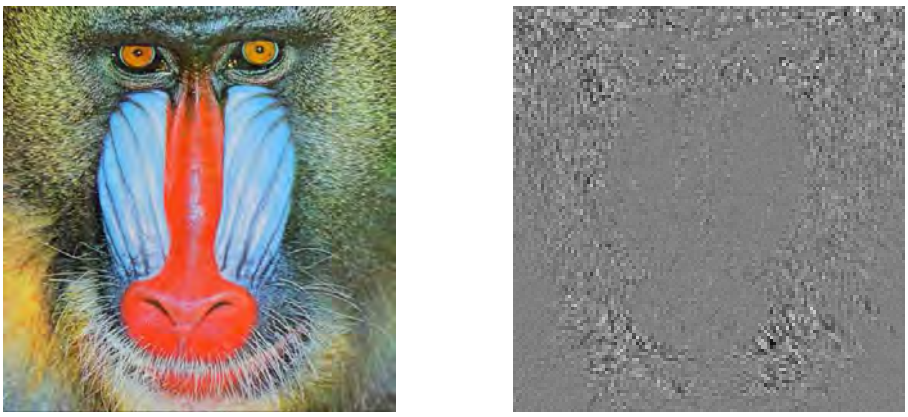
For a smooth region we expect all the pixels which comprise the region to have similar values, whereas for a textured region we expect the pixels to have widely differing values. In the pixel domain, whether a given coefficient is part of a smooth region or a textured region is indicated by the energy, or variance, of the pixels in a local neighbourhood about the coefficient's location in the untransformed image. Pixels for which the energy is high are assumed to belong to textured regions.

In the wavelet domain, similar information is conveyed by the magnitudes of the coefficients corresponding to the same local neighbourhood. The coefficients which comprise a smooth region tend to have low magnitudes, whereas for a textured region the coefficients tend to have large magnitudes (see figure C.2). Thus, in the DWT domain, the energy may be represented by the average magnitude of the neighbourhood of coefficients. For a block of coefficients B_i containing the selected coefficient x_i, in the subband s_i the energy $e(B_i)$ of the block will therefore be defined as

$$e(B_i) = \frac{sum_{x \in B_i} |x|}{|B_i|}, \tag{C.2}$$

where $|B_i|$ represents the number of coefficients in the block B_i. If the energy in a block is high then it is likely to belong to a textured region, whereas if the energy is low then it is likely to belong to a smooth region.

Unfortunately, both edge regions and textured regions tend to have high energy. As a result, an energy measure may not provide sufficient ability to distingush between textured and edge regions.



(a) The mandrill image

(b) A wavelet subband of the mandrill image

Figure C.2: Textured regions tend to contain high magnitude coefficients (black or white). Smooth regions tend to contain low magnitude coefficients (mid-grey).

C.1.3 High Count

While an edge, like a textured region, has high energy, the spatial extent of the high energy area is limited other than in the direction of the edge. A texture is rarely limited in this manner. Thus, while both textured and edge regions have high-magnitude coefficients, the count of high magnitude coefficients in a given neighbourhood tends to be smaller for edge regions than it is for textured regions (figure C.3).

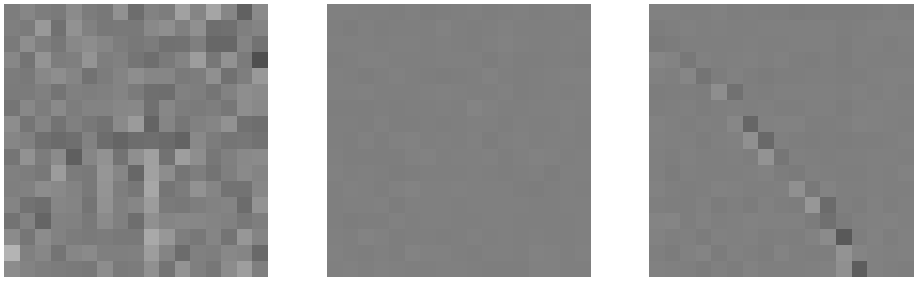
Thus an alternative method of detecting texture would be to calculate the count of high-magnitude coefficients in the neighbourhood. Smooth regions will predominantly contain low-magnitude coefficients, and should thus have a low count, edge regions should also have a reasonably low count, and textured regions should have a high count.

To use this method, it is necessary to determine the threshold at which a coefficient is considered high-magnitude. If this threshold is too high, coefficients in textued regions with moderately high magnitudes will not be included, bringing the count closer to that of an edge. However, too low a threshold will cause the inclusion of coefficients from smooth regions, causing the counts for both edged and smooth regions to increase towards those of textured regions. Using the subband energy $e(s_i)$ to form the threshold will

select coefficients that are high-magnitude relative to their subband, while a constant multiplier q can be used to set which magnitudes, relative to the average, are deemed "high". The ideal value for the constant multiplier will be image dependent, and will even vary between different resolution levels of the same image. That said, a fixed value of q should still provide reasonable results for the majority of images.

Finally, the count $c(B_i)$ includes division by the number of coefficients in the block to allow for variation in block size.

$$c(B_i) = \sum_{x \in B_i \ \wedge \ |x| > qe(s_i)} \frac{1}{|B_i|} \tag{C.3}$$



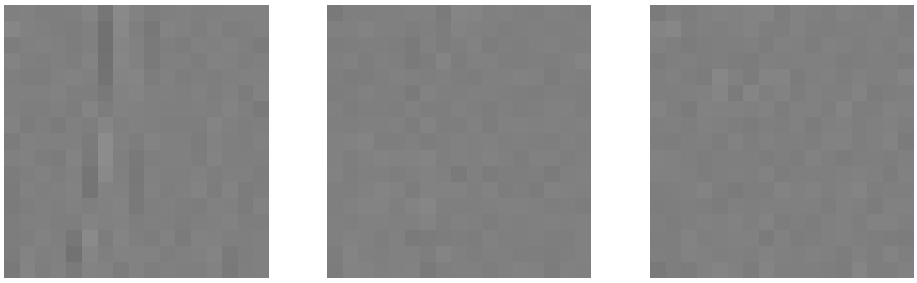
(a) Textured (b) Smooth (c) Edged

Figure C.3: Textured regions contain a greater number of high magnitude wavelet coefficients than smooth or edged regions.

C.1.4 Multiple Subbands

Although a high value for $e(B_i)$ or $c(B_i)$ suggests the presence of texture, there remains a reasonable chance that the high value is the result of a strong edge. However, unlike a great many textures, the high contrast area that occurs at an edge is strongly directional. This means that an edge is more likely to have high magnitude coefficients in only one of the vertically, horizontally or diagonally oriented subbands (figure C.4). Whereas a textured region is likely to have high magnitude coefficients at two or more of these orientations (figure C.5).

If we let $B_{i,0}$, $B_{i,1}$ and $B_{i,2}$ denote blocks of coefficients in resolution r_i, centred at the spatial location of coefficient x_i, but with subband orientations of 0, 1 and 2 respectively, then the chance of erroneously assigning a high texture score to an edged region can be reduced by ensuring that the energy or count values for at least two of the three blocks are high.

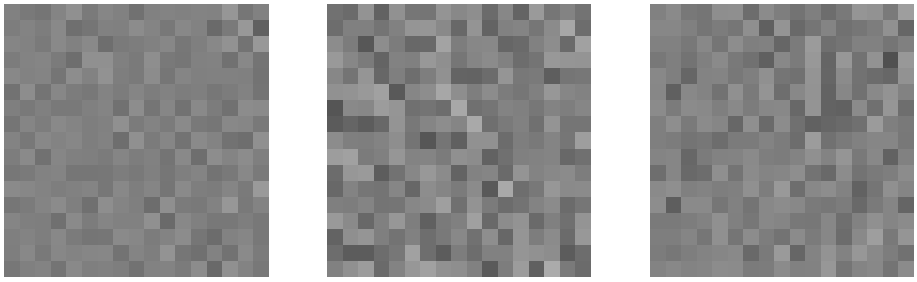


(a) An edge block in the HL subband, $o_i = 0$

(b) The same block in the LH subband, $o_i = 1$

(c) The same block in the HH subband, $o_i = 2$

Figure C.4: Edged regions typically have high magnitude coefficients at one orientation.



(a) A textured block in the HL subband, $o_i = 0$.

(b) The same block in the LH subband, $o_i = 1$

(c) The same block in the HH subband, $o_i = 2$

Figure C.5: Textured regions typically have high magnitude coefficients at more than one orientation.

Thus new *energy* and *count* features can be defined by using the maximum pairwise multiple of the values for the three different orientations, and restoring the range by dividing by their average

$$E_i = \frac{\max(\ e(B_{i,0})e(B_{i,1}),\ e(B_{i,0})e(B_{i,2}),\ e(B_{i,1})e(B_{i,2})\ )}{\frac{e(B_{i,0})+e(B_{i,2})+e(B_{i,1})}{3}} \tag{C.4}$$

$$C_i = \frac{\max(\ c(B_{i,0})c(B_{i,1}),\ c(B_{i,0})c(B_{i,2}),\ c(B_{i,1})c(B_{i,2})\ )}{\frac{c(B_{i,0})+c(B_{i,2})+c(B_{i,1})}{3}} \tag{C.5}$$

These features are calculated using only the wavelet coefficients in the same resolution r_i as the selected coefficient x_i, thus first two desriable properties for the texture scoring algorithm are satisfied.

C.2 Improving the Separation of Textured Regions

The second stage of algorithm development is to determine how best to refine and use the features developed in section C.1 to obtain a good separation of textured regions from smooth and edged regions, thereby satisfying the final desirable property of the texture scoring algorithm.

For each coefficient a texture scoring algorithm is applied to, a score is obtained which estimates the 'texturedness' of the region in which the coefficient lies. By applying such an algorithm to a large set of coefficients, a frequency histogram of the scores may be produced. An ideal method would be one for which the histogram composed using coefficients belonging to textured regions would be completely disjoint from the histograms composed using coefficients belonging to smooth or edged regions. Thus any measure of the separability of the texture histogram from the edge and smooth histograms will provide an indication of the performance of the algorithm.

A series of experiments are conducted, in which potential modifications to the features are tested to see which provides the best separation between the texture-score histograms of textured regions and those of smooth and edged regions. The best modifications in each experiment are used as the basis for further improvements in following experiments, until an algorithm is obtained in which the scores for textured regions are reasonably well separated from those of smooth and edged regions.

C.2.1 Experimental Method

To test which of the potential texture scoring algorithms provides the best separation, they are evaluated using experiments on three images (Lena, Mandrill and Peppers). Each experiment can be considered as four steps (although the outcome of the first two steps remains identical for all experiments):

Coefficient Selection

A set of significant coefficients X is selected from each image using the same coefficient selection scheme as in section 4.3 (page 113) that is used to select points for watermarking. To provide a good sized sample, 3000 coefficients are selected from each image. Six resolution layers are used in the wavelet decomposition; however, texture scores will only be calculated for medium to high frequency coefficients.

Classification

Each selected coefficient $x_i \in X$ is classified as smooth, edged or textured, according to the region to which it belongs. Because there is no way to objectively classify

coefficients based on texture, these classifications are performed based on a hand constructed 'texture map' of the image from which the coefficients are drawn, in which the pixels have been coloured red, green or blue depending on whether they appear to belong to smooth, edged or textured regions respectively (figure C.6).

Texture Scoring

For each selected coefficient $x_i \in X$ whose resolution layer r_i satisfies $r_i \geq 2$ (see section C.1.1), a texture score t_i is calculated. This process is repeated using each of the potential texture scoring algorithms being considered in the experiment.

Separability Evaluation

The separation between the texture-score histograms of textured and edged coefficients, and textured and smooth coefficients, is calculated. Because the numbers of coefficients belonging to each class are unequal, the histograms are first normalised by dividing by the number of samples in the corresponding class before computing the separability. Each calculation uses the aggregate of a variety of separability measures, which are described in section C.2.2. A single separability score $\mathcal{S}$ is assigned to each algorithm by averaging the aggregate separabiltiy results for textured vs. edged and textured vs. smooth[1]

$$\mathcal{S} = \frac{\mathrm{HS}(Textured, Edged) + \mathrm{HS}(Textured, Smooth)}{2} \tag{C.6}$$

C.2.2 Separability Measures

For easier comparison, an aggregate measure of histogram separability is calculated by combining four individual measures of separability. Each individual measure is adjusted to range from zero to one with higher values indicating a better method. The aggregate measure, $\mathrm{HS}(A, B)$ is composed of the sum of these four measures:

$$\mathrm{HS}(A, B) = (1 - \mathrm{HI}(A, B)) + \mathrm{HX}^2(A, B) + tp_{tex95}(A, B) + (1 - \mathrm{BE}(A, B)). \tag{C.7}$$

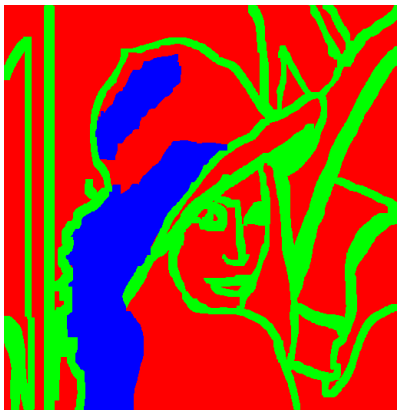
Each of the individual measures is described below.

C.2.2.1 Histogram Intersection $HI(A, B)$

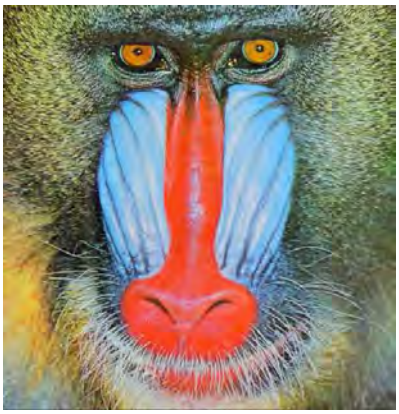
Histogram intersection is the percentage error arising from classifying each coefficient by the the class which has the greatest probability of occurance at the corresponding texture score. The greater the histogram intersection, the poorer the separation between the two

[1] The separation between the texture-score histograms of edged and smooth coefficients is not considered in the separability score. This is because the human visual system is assumed to be equally sensitive to changes in both types of regions, so only the ability to separate textured regions from both smooth and edged regions is important for watermarking purposes.

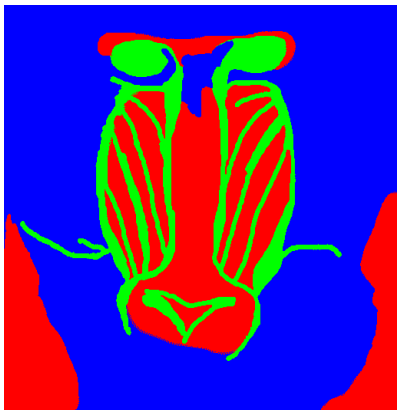
(a) Lena



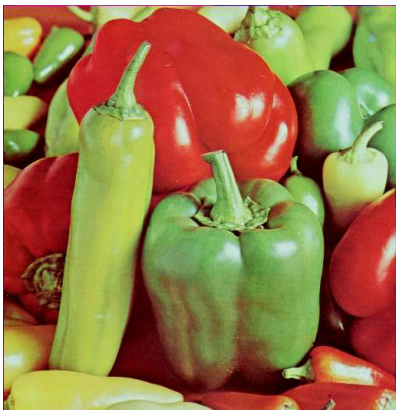
(b) Map of Lena



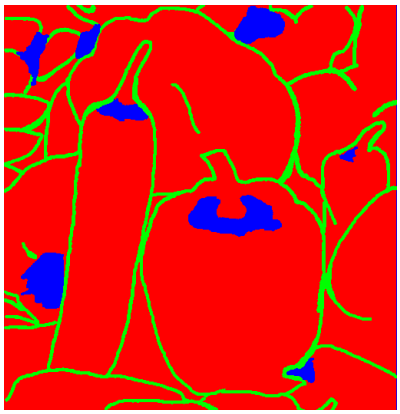
(c) Mandrill



(d) Map of Mandrill



(e) Peppers



(f) Map of Peppers

Figure C.6: Hand-constructed texture maps for the three images showing smooth (red), edged (green) and textured (blue) regions.

histograms. If $h_A(t)$ is the value of the histogram for class A at score t, then the histogram intersection is

$$\mathrm{HI}(A,B) = \sum_i min(h_A(t), h_B(t)). \tag{C.8}$$

Using $1 - \mathrm{HI}(A,B)$ gives a separability measure which ranges between 0 and 1.

C.2.2.2 Histogram Chi Squared

The chi-squared test is a standard statistical test for calculating histogram similarity. The chi squared metric for the separability of the two histograms for classes A and B is [159]:

$$\mathrm{HX}^2(A,B) = \sum_t \frac{(h_A(t) - h_B(t))^2}{h_A(t) + h_B(t)}. \tag{C.9}$$

C.2.2.3 True Positives

As the desired outcome of employing these methods is to be able to select which points in an image belong to textured regions, while keeping selected points from edged or smooth regions to a minimum, a measure of their success would be the percentage of textured points with scores exceeding a threshold. In order to make this comparable across all methods the threshold should be established to ensure that the majority (in this case 95%) of smooth or edge points are excluded from selection.

Assuming that positive result (class A) is expected to produce a larger score than a negative result (class B), the true positive rate at the 95% true negative threshold is

$$\mathrm{TP}_{\mathrm{tex95}}(A,B) = \sum_{t > T_{\mathrm{tex95}}} h_A(t) \tag{C.10}$$

where T_{tex95} is the score satisfying

$$\sum_{t < T_{\mathrm{tex95}}} h_B(t) \tag{C.11}$$

C.2.2.4 Bhattacharyya Error

Perhaps the most commonly used distance measure for normally distributed scores is the Bhattacharyya Distance

$$\mathrm{BD}(A,B) = 0.125(\mu_A - \mu_B)^2 \times \frac{2}{\sigma_A^2 + \sigma_B^2} + \frac{1}{2}\ln\frac{\sigma_A^2 + \sigma_B^2}{2\sigma_A\sigma_B} \tag{C.12}$$

The battacharyya error,

$$\mathrm{BE}(A,B) = \frac{1}{2}e^{-\mathrm{BD(A,B)}} \tag{C.13}$$

provides an upper bound on the error arising from the use of an optimal bayesian classifier[117]. Thus, using $1 - \mathrm{BE}(A,B)$ gives a separability measure which ranges between 0 and 1.

C.3 Experiments

C.3.1 Experiment 1 – Blocksize

The first experiment examines the effect of the number of coefficients $|B_i|$ used to define each block on calculation of both energy-based $t_i = E_i$ and count-based $t_i = C_i$ texture scores. Note that only odd blocksizes are used, so that the coefficient x_i remains in the centre of the block. For this experiment the constant multiplier q used in the calculation of $c(B_i)$ is set to 0.375. In all cases, an increase in blocksize $|B_i|$ increases the separability $\mathcal{S}$

Table C.1: Separation obtained using count and energy based scores at blocksizes ranging from 5×5 to 11×11.

Algorithm no.	$\lvert B_i \rvert$	Technique	$\mathcal{S}$
1	5×5	energy	1.25
2	7×7	energy	1.44
3	9×9	energy	1.57
4	11×11	energy	1.67
5	5×5	count	1.23
6	7×7	count	1.38
7	9×9	count	1.56
8	11×11	count	1.71

of the texture histograms from the edge and smooth histograms, although it will increase the expense of score calculation also.

C.3.2 Experiment 2 – Constant Multiplier and Blocksize

The second experiment examines the effect of doubling constant multiplier q on calculation of count based scores. As increased blocksizes offered increased separability, this experiment will also increase the blocksize further.

A multiplier of 0.75 shows a subtantial increase in separability compared to a multiplier of 0.375. The increase in separation achieved by increasing blocksize becomes limited for blocks of size 15×15 and above.

Table C.2: Separation obtained using count based scores with constant multipliers 0.375 and 0.75 and blocksizes ranging from 13×13 to 19×19.

Algorithm no.	$\lvert B_i \rvert$	Technique	q	$\mathcal{S}$
9	13×13	count	0.375	1.71
10	15×15	count	0.375	1.71
11	17×17	count	0.375	1.74
12	19×19	count	0.375	1.77
13	11×11	count	0.75	1.93
14	13×13	count	0.75	2.08
15	15×15	count	0.75	2.11
16	17×17	count	0.75	2.11
17	19×19	count	0.75	2.08

C.3.3 Experiment 3 – Spatial Blocksize and Constant Multiplier

For the third experiment the type of block used is altered so that instead of all blocks having a fixed size in the DWT domain, they all have a fixed size in the spatial domain. Because the blocksize doubles with each increase of resolution level, only a few blocksizes are reasonable. At the highest resolution level $r_i = 5$, the blocksizes are 16×16, 32×32 and 64×64 coefficients but are only 2×2, 4×4 and 8×8 at the lowest resolution level $r_i = 2$.

Due to the increase in separation obtained by doubling the constant multiplier for the count based scores in experiment 2, these count based scores are calculated using the better performing multiplier of experiment 2 ($q = 0.75$) and twice that value($q = 1.5$).

In all cases, the best of the spatially constant blocksizes is $2^{r_i} \times 2^{r_i}$. However, comparing the results of the count based algorithm with a constant multiplier of 0.75 and that blocksize (algorithm no. 22) to the corresponding algorithms with fixed blocksizes in the DWT domain (algorithm no.s 13 to 19) suggests that a fixed blocksize in the DWT domain is superior to a fixed blocksize in the spatial domain. As with the previous experiment, increasing the constant multiplier has increased performance.

Table C.3: Separation obtained using energy and count based scores with spatially fixed blocksizes, with constant multipliers of 0.75 and 1.5 for the count based scores.)

Algorithm no.	$\lvert B_i \rvert$	Technique	q	S
18	$2^{r_i-1} \times 2^{r_i-1}$	energy	-	1.62
19	$2^{r_i} \times 2^{r_i}$	energy	-	1.79
20	$2^{r_i+1} \times 2^{r_i+1}$	energy	-	1.49
21	$2^{r_i-1} \times 2^{r_i-1}$	count	0.75	1.68
22	$2^{r_i} \times 2^{r_i}$	count	0.75	1.96
23	$2^{r_i+1} \times 2^{r_i+1}$	count	0.75	1.65
24	$2^{r_i-1} \times 2^{r_i-1}$	count	1.5	1.75
25	$2^{r_i} \times 2^{r_i}$	count	1.5	2.15
26	$2^{r_i+1} \times 2^{r_i+1}$	count	1.5	1.71

C.3.4 Experiment 4 – Blocksize and Exclude Highly Directional

Although increased blocksizes were considered for the count based scores, in experiment 2, they have not been considered for the energy based scores. In this experiment the energy based scores are calculated using blocksizes ranging from 11×11 to 17×17.

Another technique to exclude highly-directional features from obtaining high texture scores is also considered. If a coefficient has energy scores of $e(B_{i,o_0})$ $e(B_{i,o_1})$ and $e(B_{i,o_2})$ at different orientations o_0, o_1 and o_2 (in any order) and $e(B_{i,o_0}) > 1.5(e(B_{i,o_1}) + e(B_{i,o_2}))$ then the texture score for that coefficient is set to zero.

Table C.4: Separation obtained using energy based scores with various blocksizes, with and without exclusion of highly directional features.

Algorithm no.	$\lvert B_i \rvert$	Technique	Exclude	$\mathcal{S}$
27	11×11	energy	no	1.67
28	13×13	energy	no	1.78
29	15×15	energy	no	1.87
30	17×17	energy	no	1.92
31	11×11	energy	yes	1.39
32	13×13	energy	yes	1.47
33	15×15	energy	yes	1.56
34	17×17	energy	yes	1.60

In the case of energy based techniques, further increases in block size continue to show improvements in separability. However, the exclusion criterion given is at least as likely to select points from textured regions as it is points from smooth or edged regions, and thus reduces the separability rather than improving it.

C.3.5 Experiment 5 – Constant Multiplier and Blocksize

Increases in the contstant multiplier q, which increase the value which the coefficient magnitudes must exceed before being included in the count, have improved the separability of the count based algorithms, in experiments 2 and 3. This experiment shows considers constant multipliers of 0.75 and 1.5, as well as a further increase to 2.5, for the two best performing DWT-based blocksizes and the best spatially-based blocksize.

Table C.5: Separation obtained using count based scores with constant multipliers 0.75, 1.5 and 2.5 and blocksizes 15×15, 17×17 and $2^{r_i} \times 2^{r_i}$.

Algorithm no.	$B_i\|$	Technique	q	$\mathcal{S}$
35	15×15	count	0.75	2.11
36	17×17	count	0.75	2.11
37	$2^{r_i} \times 2^{r_i}$	count	0.75	1.96
38	15×15	count	1.5	2.31
39	17×17	count	1.5	2.32
40	$2^{r_i} \times 2^{r_i}$	count	1.5	2.141
41	15×15	count	2.5	2.17
42	17×17	count	2.5	2.23
43	$2^{r_i} \times 2^{r_i}$	count	2.5	1.87

The multiplier 1.5 shows superior separability compared to either the smaller multiplier 0.75 or the larger multiplier 2.5. Assuming the coefficients have a laplacian distribution, the proportion of coefficients in the subband that deemed high magnitude (i.e. $|x_i| \geq qe(s_i)$) is expected to be e^{-q}. The DWT-based blocksizes are consistently superior to the spatially-based blocksize, with a blocksize of 17×17 showing the best performance by a small margin.

C.3.6 Experiment 6 – Exclude Highly Directional

Having established an acceptable value for the constant multiplier, the exclusion of highly directional features is tested on the count based scores. As with experiment 4, the texture score for a coefficient is set to zero if $c(B_{i,o_0}) > 1.5(c(B_{i,o_1}) + c(B_{i,o_2}))$. This may include

too many coefficients, so the same approach is tried using a stricter criterion $c(B_{i,o_0}) > 3(c(B_{i,o_1}) + c(B_{i,o_2}))$. Finally, because the HH subband covers a wide range of orientations coefficients with high scores in this subband are prevented from exclusion; that is, the texture score is set to zero when $c(B_{i,o_0}) > 1.5(c(B_{i,o_1}) + c(B_{i,o_2}))$ provided o_0 is not the HH orientation.

The blocksizes chosen are for the two best performing DWT-based blocks, 15×15 and 17×17.

Table C.6: Separation obtained using count based scores for various methods of excluding highly directional features, with blocksizes 15×15 and 17×17.

Experiment no.	$\lvert B_i \rvert$	q	Exclude	$\mathcal{S}$
44	15×15	1.5	none	2.31
45	17×17	1.5	none	2.32
46	15×15	1.5	$c(B_{i,o_0}) > 3(c(B_{i,o_1}) + c(B_{i,o_2}))$	2.30
47	17×17	1.5	$c(B_{i,o_0}) > 3(c(B_{i,o_1}) + c(B_{i,o_2}))$	2.31
48	15×15	1.5	$c(B_{i,o_0}) > 1.5(c(B_{i,o_1}) + c(B_{i,o_2}))$	2.27
49	17×17	1.5	$c(B_{i,o_0}) > 1.5(c(B_{i,o_1}) + c(B_{i,o_2}))$	2.31
50	15×15	1.5	$c(B_{i,o_0 \neq \mathrm{HH}}) > 1.5(c(B_{i,o_1}) + c(B_{i,o_2}))$	2.27
51	17×17	1.5	$c(B_{i,o_0 \neq \mathrm{HH}}) > 1.5(c(B_{i,o_1}) + c(B_{i,o_2}))$	2.31

As was the case with the energy scores in experiment 4, exclusion (of all the types considered) creates a slight decrease in the separability. Also, no coefficients had sufficiently strong features in the HH orientation to be excluded, thus algorithms 58 and 49 produced precisely the same results as algorithms 50 and 51.

C.3.7 Experiment 7 – Combining Techniques and Blocksize

While the energy based technique results in poorer separability than does the count based technique, both tend to assign generally high scores to all textured coefficients but have problems with edged or smooth coefficients also recieving high scores. It is possible that the inherent differences between the two approaches will mean that some coefficients that incorrectly recieve high scores with one technique, will recieve quite low scores with the other technique, while textured coefficients will recieve relatively high scores on both measures. Thus averaging the scores from the two techniques may result in better separability than the use of either technique alone.

This experiment examines the separability obtained using energy and count based scores individually and in combination. Because the count based scores are substantially

lower than the energy based scores, they are multiplied by 10 before combining: $t_i = \frac{E_i+10C_i}{2}$. An increased blocksize of 19×19 is considered in addition to the blocksizes 15×15, and 17×17. The constant multiplier remains at $q = 1.5$.

Table C.7: Separation obtained using energy and count based scores individually and in combination, for blocksizes 15×15 to 19×19.

Algorithm no.	$\lvert B_i \rvert$	Technique	q	$\mathcal{S}$
52	15×15	E_i	-	1.87
53	17×17	E_i	-	1.93
54	19×19	E_i	-	1.93
55	15×15	C_i	1.5	2.31
56	17×17	C_i	1.5	2.32
57	19×19	C_i	1.5	2.31
58	15×15	$\frac{E_i+10C_i}{2}$	1.5	2.34
59	17×17	$\frac{E_i+10C_i}{2}$	1.5	2.40
60	19×19	$\frac{E_i+10C_i}{2}$	1.5	2.40

Calculating texture scores as a combination $t_i = \frac{E_i+10C_i}{2}$ of energy and count results achieves superior separability across all blocksizes compared to using either energy $t_i = E_i$ or count $t_i = C_i$ alone. Increasing the blocksize to 19×19 does not positively affect separability, which suggests 17×17 to be the best blocksize.

C.3.8 Experiment 8 – Contribution Ratio

The mean texture score obtained using an energy based calculation μ_{E_i} is roughly fifteen times the mean texture score obtained using a count based calculation μ_{C_i}.

Thus, the combined score calculated in the previous experiment $t_i = \frac{E_i+10C_i}{2}$ effectively weights the contribution of energy as of more importance than count at a ratio of 3 : 2 $(1\mu_{E_i} : 10\mu_{C_i})$. This experiment shows the separation obtained for other ratios.

As could be expected from the superior performance of count based scores relative to energy based scores in earlier experiments, changing the bias away from energy and towards count increases separability. This continues until the point where the contribution from the count scores starts to overwhelm the contribution from energy scores and begins to erode the gains obtained by combining.

Table C.8: Separation obtained by combining energy and count based scores at various ratios based on their respective means, for a blocksize of 17×17 and a constant multiplier of 1.5.

Experiment no.	$\lvert B_i \rvert$	ratio	q	$\mathcal{S}$
61	17×17	2:1	1.5	2.36
62	17×17	3:2	1.5	2.40
63	17×17	4:3	1.5	2.45
64	17×17	5:4	1.5	2.40
65	17×17	1:1	1.5	2.45
66	17×17	4:5	1.5	2.46
67	17×17	3:4	1.5	2.48
68	17×17	2:3	1.5	2.47
69	17×17	1:2	1.5	2.47

C.3.9 Experiment 9 – Readjustment and Contribution Ratio

The count and energy values E_i and C_i have, until this point, been readjusted to have a similar range to their single-subband derived counterparts $e(B_i)$ and $c(B_i)$ by dividing by $\frac{e(B_{i,0})+e(B_{i,2})+e(B_{i,1})}{3}$ (see section C.1.4). However, this procedure unreasonably penalizes those coefficients with high values at all three subband orientations compared to those with high values at only two subband orientations. A better way to perform the same adjustment is to take the square root of max($e(B_{i,0})e(B_{i,1})$, $e(B_{i,0})e(B_{i,2})$, $e(B_{i,1})e(B_{i,2})$) instead, as this makes no reference to the third subband and so avoids any penalty.

This experiment is a repetition of experiment 8, in which the same combination ratios are considered but the E_i and C_i values are now calculated according to the formulae

$$E_i = \sqrt{\max(\ e(B_{i,0})e(B_{i,1}),\ e(B_{i,0})e(B_{i,2}),\ e(B_{i,1})e(B_{i,2})\ )} \quad \text{and}$$
$$C_i = \sqrt{\max(\ e(B_{i,0})e(B_{i,1}),\ e(B_{i,0})e(B_{i,2}),\ e(B_{i,1})e(B_{i,2})\ )}.$$

The improved readjustment strategy has resulted in an increase of separability in all cases, relative to the results of experiment 8. The best separation occurs at algorithm no. 75 for which

$$t_i = 0.4E_i + 7.7\dot{2}C_i$$

Table C.9: Separation obtained by combining the readjusted energy and count based scores at various ratios based on their respective means, for a blocksize of 17×17 and a constant multiplier of 1.5.

Experiment no.	$\lvert B_i \rvert$	ratio	q	$\mathcal{S}$
70	17×17	2:1	1.5	2.40
71	17×17	3:2	1.5	2.45
72	17×17	4:3	1.5	2.48
73	17×17	5:4	1.5	2.48
74	17×17	1:1	1.5	2.53
75	17×17	4:5	1.5	2.56
76	17×17	3:4	1.5	2.54
77	17×17	2:3	1.5	2.54
78	17×17	1:2	1.5	2.55

giving a contribution ratio of

$$\begin{aligned} 0.4\mu_{E_i} : 7.7\dot{2}\mu_{C_i} &= 0.4 \times 15.\dot{4} : 7.7\dot{2} \times 1 \\ &= 6.1\dot{7} : 7.7\dot{2} \\ &= 4 : 5 \end{aligned}$$

and texture scores which usually range between 0 and 10. Algorithm no. 75 is the algorithm described in section 4.3.1.2.3, page 122 and is used to generate the texture scores for the HVS algorithm that chapter.

C.4 Evaluation

Because the texture map for each image is constructed by hand in the spatial domain and is used for all three components, the class assigned to each coefficient in the classification stage is not always correct. So even an ideal texture scoring algorithm would not obtain a perfect separability result of 4 based on these assigned classes. For the purposes of section 4.3 (page 113), the texture scoring algorithm need not be ideal; however, it is at least necessary to check that adequate separation between textured and edged and smooth regions can be obtained on images other than those used for testing.

To determine this, a texture map is generated for a number of images by comparing the texture scores of selected coefficients to a threshold of $T_{tex} = 4.5$. The generated texture map is then visually compared to the original image, to see whether this classification appears reasonable.

The texture map is constructed as follows:

- the 10000 highest magnitdue coefficients X of the luminance component[2] are selected
- a texture score t_i is generated for each selected coefficient $x_i \in X$ using algorithm 75
- a greyscale multi-resolution texture map is constructed by colouring the pixel corresponding to each coefficent
  - 128 (mid-grey) if the coefficient was not scored, $x_i \notin X$ or $r_i < 2$
  - $255 - 20t_i$ (light grey) if the coefficient is smooth or edged, $t_i \leq 4.5$
  - $128 - 13t_i$ (dark grey) if the coefficient is texture, $t_i > 4.5$

Note that a fixed threshold $T_{tex} = 4.5$ is used for all images. Although performance might be improved by selecting a threshold specifically to match the image, automatic threshold selection would likely require texture scores to be calculated for substantially more coefficients than those selected for watermarking, increasing computation costs.

The images used in this evaluation are: fontaine des terreaux, kid, paper machine, pills, black bear and wildflowers from Petitcolas' database [141].

In the first four of these images, the generated texture maps correspond well with the perceived image texture. In the fontaine des terreaux image (figure C.7), the textured regions within the fountain are classified as textured, while the edges of the fountain and the surrounding pool, as well as the light in the upper left corner are not.

In the kid image (figure C.8), the texture in the hair is picked up correctly, as is the fine texture on the sunlit serviette and sweater at the kid's left shouler. However, the coefficients along edge of the serviette at the kid's left shoulder are assigned scores which reflect the surrounding texture rather than the edge itself.

In the paper machine image (figure C.9) the texture on the upper metal grating is identified (the lower metal grating does not contain sufficiently high-magnitude coefficients to be selected). The interaction between the two closely positioned edges of reflected light along the axis of the roller and the blemish across the axis of the roller, the upper right corner of the image, are sufficiently multidirectional to cause a small texture response where no texture is visible.

The pills image (figure C.10) has almost no coefficients classified as textured; it consists almost exclusively of smooth areas and edges. The only exceptions to this occur in the highest resolution, where two pills close to each other have a tiny segment of the dark

[2] Although texture and its associated masking effects can occur in colour components, features which are perceived as texture more usually occur in the luminance component, so this component is the obvious choice when generating a texture map for visual evaluation.

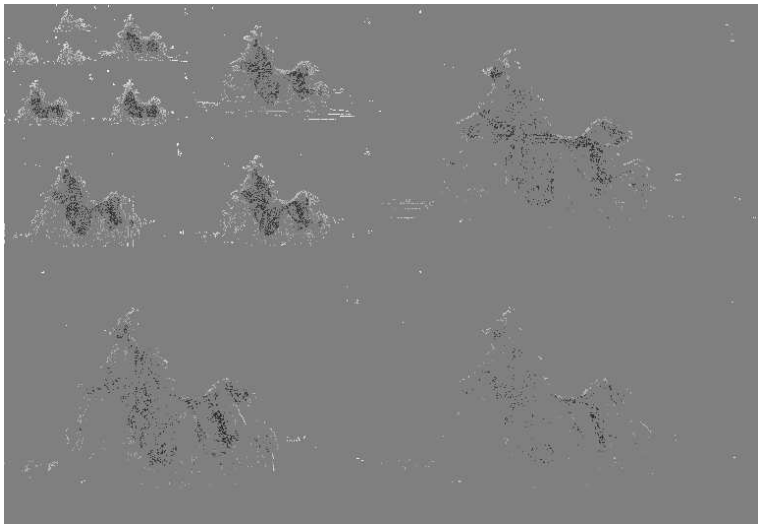
background between them. As with the paper machine image, this combination of high contrast edges at different orientations results in a classification of textured where none is expected.

The texture maps for the two remaining images may be somewhat surprising. Both the wildflowers (figure C.11) and black bear (figure C.12) images appear highly textured, yet the texture maps do not reflect this. This occurs because the threshold used in the count calculation, which controls whether a coefficient is sufficiently high in magnitude to be counted, is set according to the subband energy $e(s_i)$ (see section C.1.3). In images that consist of uniform texture, such as the wildflowers image, no area is particularly textured relative to other areas of the image, so no area is classified as textured.

While this is clearly not a desirable trait for a texture identification algorithm in general, it is ideal for inclusion in the watermarking algorithm of section 4.3, in which the global watermark strength is already controlled via the parameter α and the aim is to identify local regions for which the strength can be increased relative to the image as a whole. This is exactly what occurs for the black bear image (figure C.12), where the texture of the bear's fur is ignored, while the texture of the ground is picked up. Looking at a patch of the ground and one of the fur (figure C.13), it is clear that the ground is a higher contrast and more multidirectional texture, and is thus a better candidate for increases in embedding strength.

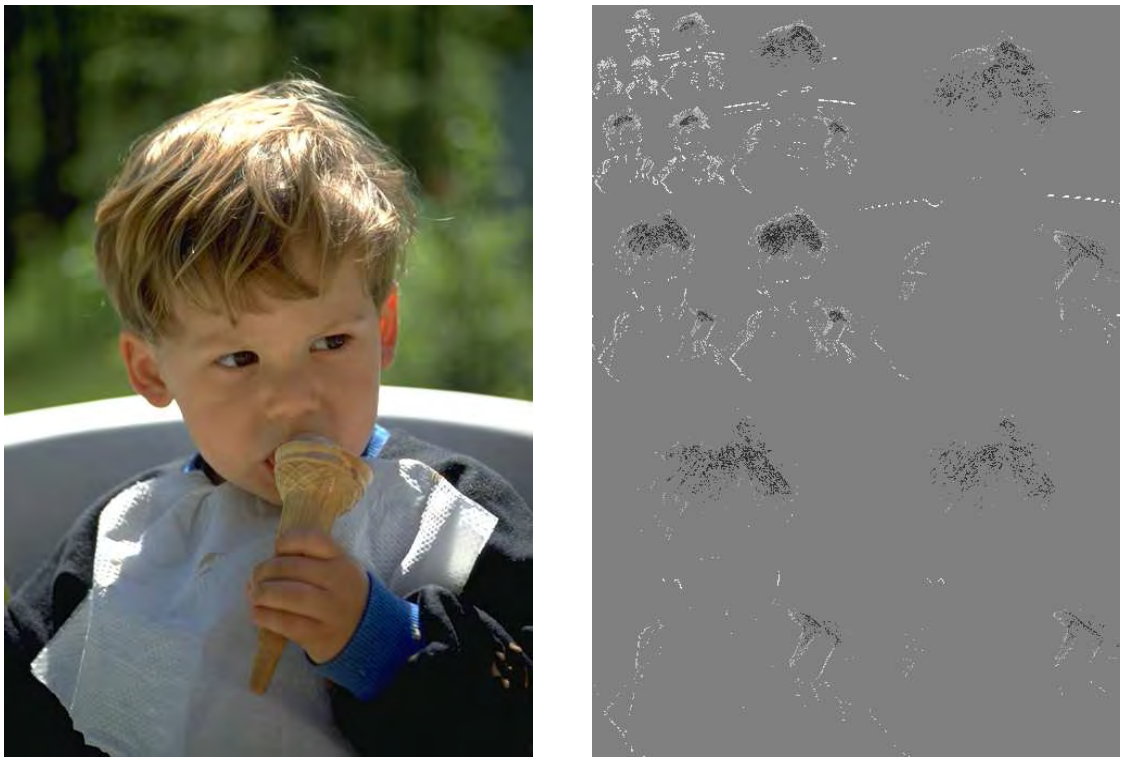
Overall, the performance of algorithm 75 appears quite acceptable. Although minor problems with incorrectly classified edges do exist, and thorough psychovisual experiments would doubtless yield substantial improvements, the algorithm provides single-resolution, wavelet domain texture scoring that excludes edges sufficiently to be suitable for use in section 4.3.

(a) Fontaine de terreaux



(b) Generated texture map

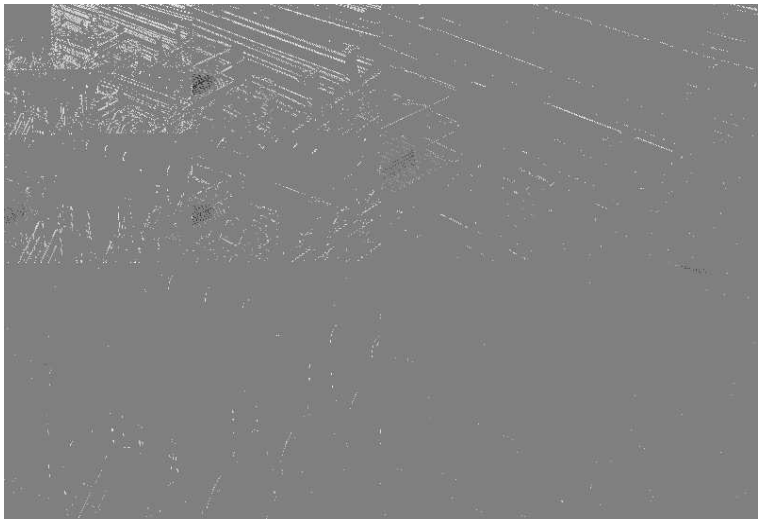
Figure C.7: Fontaine de terreaux and the corresponding texture map generated by algorithm 75, showing textured (dark), non-textured (light) and unselected (mid-grey) coefficients.



(a) Kid (b) Generated texture map

Figure C.8: Kid and the corresponding texture map generated by algorithm 75, showing textured (dark), non-textured (light) and unselected (mid-grey) coefficients.

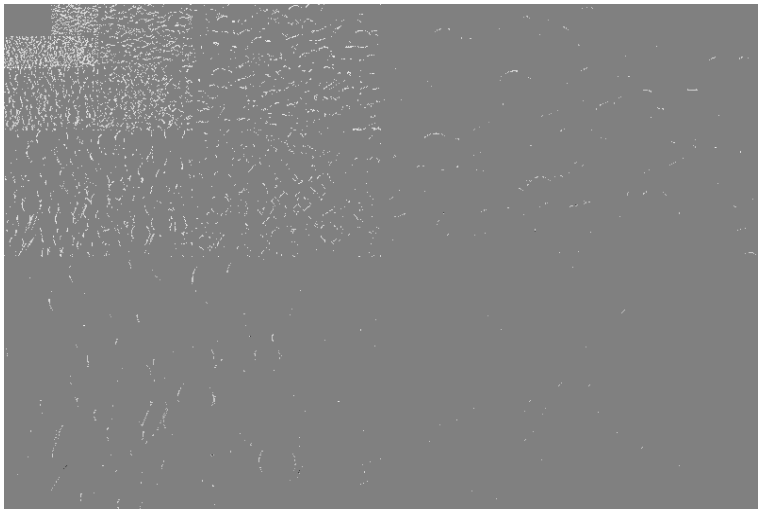
(a) Paper machine



(b) Generated texture map

Figure C.9: Paper machine and the corresponding texture map generated by algorithm 75, showing textured (dark), non-textured (light) and unselected (mid-grey) coefficients.

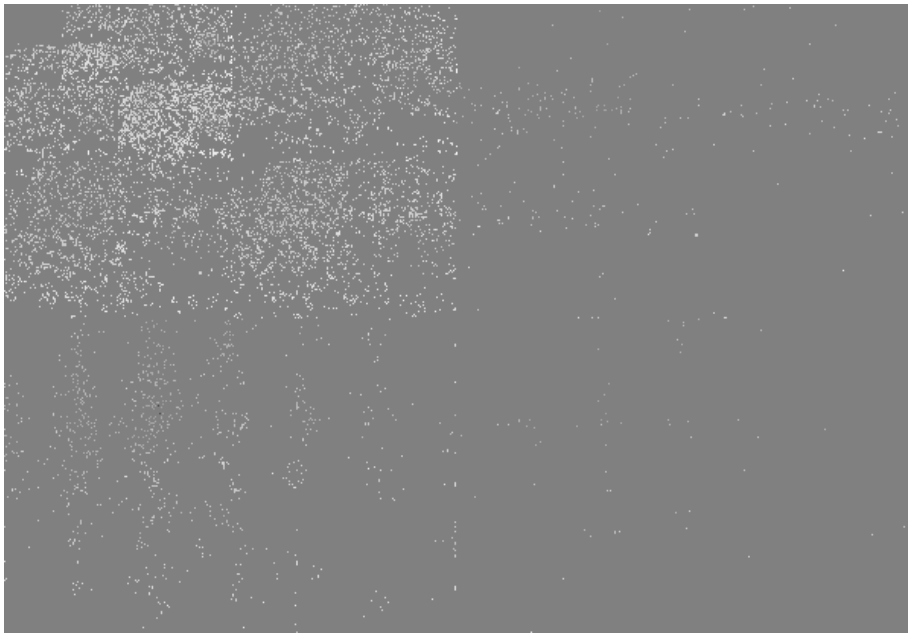
(a) Pills



(b) Generated texture map

Figure C.10: Pills and the corresponding texture map generated by algorithm 75, showing textured (dark), non-textured (light) and unselected (mid-grey) coefficients.

(a) Wildflowers

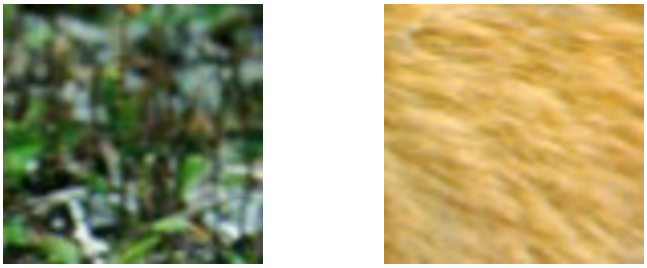


(b) Generated texture map

Figure C.11: Wildflowers and the corresponding texture map generated by algorithm 75, showing textured (dark), non-textured (light) and unselected (mid-grey) coefficients.

(a) Black Bear (b) Generated texture map

Figure C.12: Black bear and the corresponding texture map generated by algorithm 75, showing textured (dark), non-textured (light) and unselected (mid-grey) coefficients.



(a) Ground (b) Fur

Figure C.13: The ground texture is stronger than the fur texture

Appendix D

A Blind Scalable Watermark for JPEG2000: Basic Algorithm

D.1 Algorithm Examples

This section shows the operation of the embedding and detection algorithms on a single image coefficient. In the first detection there are no bits missing from the coefficient while in the second detection the two least significant bits have been lost due to quality scaling.

D.1.1 Embedding

This shows the process of embedding a watermark element $u_i = -3$ in a coefficient $v_i = 78$ from the 700×457 image nerr0787 with $C = 3$ components and $R = 6$ resolution layers. The coefficient v_i has coordinates $x_i = 9, y_i = 25$ in the subband with resolution $r_i = 2$ and orientation $o_i = 1$.

Initial Parameters	I = nerr0787.ppm $\alpha = 0.0625$ $t = 8$ $sk = 1$
Coefficient Selection $\lvert v\rvert \geq t$	$v_i = 78$
Indexing $i = cX[r]Y[r] + (C-c)X[r-1]Y[r-1] +$ $\sum_{a=0}^{a=o-1} X[r,a]Y[r,a] +$ $yX[r,o] + x$	i = 0 + (3-0)*44*29 + $44 * 29+$ $25 * 44 + 9$ $= 6123$
Element Generation $w^*_{i^*} = \mathtt{g}(i^*, sk^*)$ $u^*_{i^*} = \left\lfloor \alpha \frac{\bar{v}^*_{i^*}}{4} w^*_{i^*} \right\rfloor$ while $\bar{u}_i \geq \frac{\bar{v}_i}{2}$ set $u_i = \frac{u_i}{2}$	$w^*_{i^*} = \mathtt{g}(6123, 1)$ = -1.5086 $u^*_{i^*} = \lfloor 0.0625 \frac{128}{4} (-1.5086) \rfloor$ = -3 $4 \geq 128$ = false
Coefficient Quantization $Q_{2u_i}(v_i) = \left\lfloor \frac{\lvert v_i\rvert}{2\bar{u}_i} \right\rfloor 2\bar{u}_i$	$2\bar{u}_i = 8$ $Q_8(v_i) = \left\lfloor \frac{\lvert 78\rvert}{8} \right\rfloor 8$ $= 72$
Watermark Embedding $v'_i = \mathrm{sign}(v_i)(Q_{2u_i}(v_i) + \frac{\bar{u}_i - sign(u_i)\bar{u}_i}{2} + \lvert u_i\rvert)$	$v'_i = +(72 + \frac{4--4}{2} + 3)$ $= 79$

D.1.2 Detection - $m_i = 0$

This shows the detection algorithm where all 24 passes have been received (no bits lost to quality scaling). The decoder receives the entire watermarked coefficient, $v^*_{i^*} = v'_i = 79$.

Initial Parameters	$I^* =$ nerr0787.jpc $\alpha = 0.0625$ $t = 8$ $k_d = 1$ $m_i = 0$
Coefficient Selection $\lvert v^*_{i^*}\rvert \geq t$	$v^*_i = 79$
Indexing $i^* = c^*X^*[r^*]Y^*[r^*] + (C^* - c^*)X^*[r^* - 1]Y^*[r^* - 1] +$ $\sum_{a=0}^{a=o^*-1} X^*[r^*, a]Y^*[r^*, a] +$ $y^*X^*[r^*, o^*] + x^*$	i = 0 + (3-0)*44*29 + $44 * 29+$ $25 * 44 + 9$ $= 6123$
Element Generation $w^*_{i^*} = \mathsf{g}(i^*, sk^*)$	$w_i = \mathsf{g}(6123, 1)$ = -1.5086
$u^*_{i^*} = \left\lfloor \alpha \frac{\bar{v}^*_{i^*}}{4} w^*_{i^*} \right\rfloor$	$u^*_{i^*} = \left\lfloor 0.0625 \frac{128}{4} (-1.5086) \right\rfloor$ = -3
Missing Bits Calculation $m^*_{i^*} = M_{s^*_{i^*}} - Z_{i^*} - 1 - \left\lfloor \frac{P_{b^*_{i^*}}}{3} \right\rfloor -$ $\begin{cases} 1 & P_{b^*_{i^*}} \equiv 2 \pmod 3 \text{ and } \bar{v}^*_{i^*} = 2^{(M_{s^*_{i^*}} - Z_{i^*} - 1 - \lfloor \frac{P_{b^*_{i^*}}}{3} \rfloor)} \\ 0 & \text{otherwise.} \end{cases}$	$m_i = 13 - 4 - 1 - \left\lfloor \frac{24}{3} \right\rfloor - 0$ $= 0$
Watermark Bits Calculation $j^*_{i^*} = \log_2(\bar{u}^*_{i^*})$	$j_i = \log_2(4)$ $= 2$
Watermark Truncation $u^c_{i^*} = sign(u^*_{i^*}) \left\lfloor \frac{\lvert u^*_{i^*}\rvert}{2^{m^*_{i^*}}} \right\rfloor 2^{m^*_{i^*}}$	$u^c_i = - \left\lfloor \frac{3}{2^0} \right\rfloor 2^0$ = -3
Watermark Extraction $u^d_{i^*} = \left(\lvert v^*_{i^*}\rvert - Q_{u^*_{i^*}}(v^*_{i^*}) - \lfloor 2^{m^*_{i^*}-1} \rfloor \right) (-1)^{\left\lfloor \frac{\lvert v^*_{i^*}\rvert - Q_{2u^*_{i^*}}(v^*_{i^*})}{\bar{u}^*_{i^*}} \right\rfloor}$	$u^d_i = (79 - 76 - 0)(-1)^{\lfloor \frac{79-72}{4} \rfloor}$ = -3

D.1.3 Detection - $m_i = 2$

This shows the detection algorithm where only 18 of the original 24 passes have been received, so the 2 least significant bits of the coefficient have been lost due to quality scaling. The decoder knows only that the received coefficient is in the range [76, 79] and reconstructs the coefficient as $v^*_{i^*} = 78$.

Initial Parameters	$I' =$ nerr0787.jpc $\alpha = 0.0625$ $t = 8$ $k_d = 1$
Coefficient Selection $\lvert v^*_{i^*}\rvert \geq t$	$v^*_i = 78$
Indexing $i^* = c^*X^*[r^*]Y^*[r^*] + (C^* - c^*)X^*[r^* - 1]Y^*[r^* - 1] +$ $\sum_{a=0}^{a=o^*-1} X^*[r^*, a]Y^*[r^*, a] +$ $y^*X^*[r^*, o^*] + x^*$	i = 0 + (3-0)*44*29 + $44 * 29+$ $25 * 44 + 9$ $= 6123$
Element Generation $w^*_{i^*} = \mathsf{g}(i^*, sk^*)$	$w_i = \mathsf{g}(6123, 1)$ = -1.5086
$u^*_{i^*} = \left\lfloor \alpha \frac{\bar{v}^*_{i^*}}{4} w^*_{i^*} \right\rfloor$	$u^*_{i^*} = \left\lfloor 0.0625 \frac{128}{4} (-1.5086) \right\rfloor$ = -3
Missing Bits Calculation $m^*_{i^*} = M_{s^*_{i^*}} - Z_{i^*} - 1 - \left\lfloor \frac{P_{b^*_{i^*}}}{3} \right\rfloor -$ $\begin{cases} 1 & P_{b^*_{i^*}} \equiv 2 \pmod 3 \text{ and } \bar{v}^*_{i^*} = 2^{(M_{s^*_{i^*}} - Z_{i^*} - 1 - \lfloor \frac{P_{b^*_{i^*}}}{3} \rfloor)} \\ 0 & \text{otherwise.} \end{cases}$	$m_i = 13 - 4 - 1 - \left\lfloor \frac{18}{3} \right\rfloor - 0$ $= 2$
Watermark Truncation $u^c_{i^*} = sign(u^*_{i^*}) \left\lfloor \frac{\lvert u^* i^*\rvert}{2^{m^*_{i^*}}} \right\rfloor 2^{m^*_{i^*}}$	$u^c_i = - \left\lfloor \frac{3}{2^2} \right\rfloor 2^2$ = -0
Watermark Extraction $u^d_{i^*} = \left(\lvert v^*_{i^*}\rvert - Q_{u^*_{i^*}}(v^*_{i^*}) - \lfloor 2^{m^*_{i^*}-1} \rfloor \right) (-1)^{\left\lfloor \frac{\lvert v^*_{i^*}\rvert - Q_{2u^*_{i^*}}(v^*_{i^*})}{\bar{u}^*_{i^*}} \right\rfloor}$	$u^d_i = (78 - 76 - 2)(-1)^{\lfloor \frac{78-72}{4} \rfloor}$ = -0

D.2 Lemmas

D.2.1 Lemma

$x \in \mathbb{Z}$, $r \in \mathbb{R}, 0 \leq r < 1$, $k \in \mathbb{N}$.

$$\left\lfloor \frac{x}{k} \right\rfloor = \left\lfloor \frac{x+r}{k} \right\rfloor$$

Proof:
Let $\left\lfloor \frac{x}{k} \right\rfloor$ = t; that is, $t \in \mathbb{Z}$ and

$$t \leq \frac{x}{k} < t+1.$$

Multiplying by $k > 0$ gives

$$tk \leq x < tk + k$$

and, since $x, t, k \in \mathbb{Z}$,

$$tk \leq x \leq tk + k - 1$$
$$tk + r \leq x + r \leq tk + k - 1 + r.$$

Now $0 \leq r < 1$ so

$$tk + 0 \leq x + r < tk + k - 1 + 1.$$
$$tk \leq x + r < tk + k.$$

Dividing by $k > 0$ gives

$$t \leq \frac{x+r}{k} < t+1.$$

which, since $t \in \mathbb{Z}$, means

$$t = \left\lfloor \frac{x+r}{k} \right\rfloor$$

thus

$$\left\lfloor \frac{x}{k} \right\rfloor = \left\lfloor \frac{x+r}{k} \right\rfloor.$$

D.2.2 Lemma

$c = lb, b \in \mathbb{Z}, l \in \mathbb{N}.\ \forall a \in \mathbb{R}$

$$\left\lfloor \frac{\lfloor \frac{a}{b} \rfloor b}{c} \right\rfloor = \left\lfloor \frac{a}{c} \right\rfloor$$

Proof:

Let $\frac{a}{b} = x + r$ where $x \in \mathbb{Z}$, $r \in \mathbb{R}, 0 \leq r < 1$, that is

$$\left\lfloor \frac{a}{b} \right\rfloor = x$$

This gives us

$$\left\lfloor \frac{\lfloor \frac{a}{b} \rfloor b}{c} \right\rfloor = \left\lfloor \frac{xb}{c} \right\rfloor$$

which, as $c = lb$,

$$= \left\lfloor \frac{xb}{lb} \right\rfloor$$
$$= \left\lfloor \frac{x}{l} \right\rfloor$$

From the previous proof $x \in \mathbb{Z}, r \in \mathbb{R}, 0 \leq r < 1, l \in N \implies \lfloor \frac{x}{l} \rfloor = \lfloor \frac{x+r}{l} \rfloor$

$$= \left\lfloor \frac{x + r}{l} \right\rfloor$$

which, since $\frac{a}{b} = x + r$

$$= \left\lfloor \frac{\left(\frac{a}{b}\right)}{l} \right\rfloor$$
$$= \left\lfloor \frac{a}{lb} \right\rfloor$$

but, since $c = lb$, this is

$$= \left\lfloor \frac{a}{c} \right\rfloor$$

therefore

$$\left\lfloor \frac{\lfloor \frac{a}{b} \rfloor b}{c} \right\rfloor = \left\lfloor \frac{a}{c} \right\rfloor .$$

D.2.3 Lemma

$c = lb, b \in \mathbb{Z}, l \in N.\ \forall a \in \mathbb{R}$

$$\left\lceil \frac{\lceil \frac{a}{b} \rceil b}{c} \right\rceil = \left\lceil \frac{a}{c} \right\rceil$$

Proof:

$$\begin{aligned}
\left\lceil \frac{\lceil \frac{a}{b} \rceil b}{c} \right\rceil &= \left\lceil \frac{-\lfloor -\frac{a}{b} \rfloor b}{c} \right\rceil && \lceil x \rceil = -\lfloor -x \rfloor \\
&= -\left\lfloor -\frac{-\lfloor -\frac{a}{b} \rfloor b}{c} \right\rfloor && \lceil x \rceil = -\lfloor -x \rfloor \\
&= -\left\lfloor \frac{\lfloor \frac{-a}{b} \rfloor b}{c} \right\rfloor \\
&= -\left\lfloor \frac{-a}{c} \right\rfloor && \text{Lemma D.2.2 } c = lb, b \in \mathbb{Z}, l \in N\ \ -a \in \mathbb{R} \\
&= \left\lceil \frac{a}{c} \right\rceil && \lceil x \rceil = -\lfloor -x \rfloor.
\end{aligned}$$

D.3 Proofs of Basic Design Features

D.3.1 Proof for Coefficient Selection and Embedding

We show that the coefficient selection and embedding rules, presented in section 5.1.1, satisfy the required conditions. That is, given an unscaled, resolution scaled or quality scaled image and the correct embedding parameters $\Lambda^* = \Lambda$, all coefficients that were selected during embedding and were not 'completely lost' during scaling will be selected during detection:

- for the unscaled watermarked image, where v' is the coefficient corresponding to $v \in I$,

$$v' \in V' \iff v \in V. \tag{5.4a}$$

- for the resolution scaled image, where $v^{\mathcal{R}}$ is the coefficient corresponding to $v' \in I'$

$$v^{\mathcal{R}} \in V^{\mathcal{R}} \iff v' \in V' \wedge v^{\mathcal{R}} \in I^{\mathcal{R}} \tag{5.4b}$$

- for the quality scaled image, where $v^{\mathcal{Q}}$ is the coefficient corresponding to $v' \in I'$,

$$v^{\mathcal{Q}} \in V^{\mathcal{Q}} \iff v' \in V' \wedge v^{\mathcal{Q}} \neq 0 \tag{5.4c}$$

provided that, for any watermark element u embedded in a coefficient v,

$$\exists j \in \mathbb{N} \text{ s.t. } 0 \leq u < 2^j \leq |v|. \tag{5.5}$$

Let

$$V^* = \mathsf{Select}(I^*, \Lambda^*) \tag{5.2a}$$
$$= \{v \in I^* : |v| \geq t^* = 2^{n^*}, n^* \in \mathbb{N}\}, \tag{5.2b}$$

be the set of selected coefficients from the (potentially modified) image I^* given the threshold $t^* \in \Lambda^*$, where $n^* \in \mathbb{N}$, and let

$$v' = sign(v)(Q_{2^j}(v) + u), \tag{5.3c}$$

represent the watermarked coefficient corresponding to a selected coefficient $v \in V$ in the original image I. We assume that the watermark element u is such that there exists a quantization step size

$$\exists j \in \mathbb{N} \text{ s.t. } 0 \leq u < 2^j \leq |v|. \tag{5.5}$$

The generation of a watermark satisfying this requirement is discussed in section 5.1.4.2.

1. Unscaled

Given the unscaled image I', and the correct watermarking parameters $\Lambda' = \Lambda$, let $v \in I$ denote an original image coefficient and v' the corresponding watermarked coefficient.

If $v \in V$ then j watermark bits $u \in \mathbb{Z} \;\; 2^j > u \geq 0$ are embedded to produce the corresponding watermarked coefficient $v' \in I'$, where

$$\begin{aligned}
|v'| &= |sign(v)(Q_{2^j}(v) + u)| && \text{(5.3c)} \\
&= |Q_{2^j}(v) + u| && sign(v) = \pm 1 \;\; (2.16) \\
&= Q_{2^j}(v) + u && Q_{2^j}(v) \geq 0 \;\; (5.3a) \text{ and } u \geq 0 \;\; (5.5) \\
&\geq Q_{2^j}(v) && u \geq 0 \;\; (5.5) \\
&= \left\lfloor \frac{|v|}{2^j} \right\rfloor 2^j && (5.3a) \\
&= \left\lfloor \frac{\max(|v|, 2^j)}{2^j} \right\rfloor 2^j && |v| \geq 2^j \;\; (5.5) \\
&\geq \left\lfloor \frac{\max(2^n, 2^j)}{2^j} \right\rfloor 2^j && v \in V \implies |v| \geq t = 2^n \;\; (5.2) \\
&= \max(2^n, 2^j) && n, j \in \mathbb{N} \implies \frac{\max(2^n, 2^j)}{2^j} \in \mathbb{N} \\
&\geq 2^n
\end{aligned}$$

which, since the correct watermarking parameters are provided $\Lambda' = \Lambda$,

$$\begin{aligned} &= 2^{n'} && n' \in \Lambda' \equiv n \in \Lambda \implies 2^{n'} = 2^n \\ \therefore v' &\in V' && V' = \{v \in I' : |v| \geq t' = 2^{n'}\} \ \ (5.2). \end{aligned}$$

If $v \notin V$ then no watermark bits are embedded and the watermarked coefficient $v' \in I'$ has the value

$$\begin{aligned} v' &= v \qquad \text{so} \\ |v'| &= |v| \\ &< 2^n && v \notin V \implies |v| < t = 2^n \ \ (5.2) \end{aligned}$$

which, since the correct watermarking parameters are provided $\Lambda' = \Lambda$,

$$\begin{aligned} &= 2^{n'} && n' \in \Lambda' \equiv n \in \Lambda \implies 2^{n'} = 2^n \\ \therefore v' &\notin V' && V' = \{v \in I' : |v| \geq t' = 2^{n'}\} \ \ (5.2b). \end{aligned}$$

So $v' \in V' \iff v \in V$ and property (5.4a) is proven.

2. Resolution Scaled

Given the resolution scaled image $I^{\mathcal{R}}$ and the correct watermarking parameters $\Lambda^{\mathcal{R}} = \Lambda$, let $v' \in I'$ denote a watermarked coefficient and $v^{\mathcal{R}}$ the corresponding resolution scaled coefficient.

If both $v' \in V'$ and $v^{\mathcal{R}} \in I^{\mathcal{R}}$ then, since $v^{\mathcal{R}} \in I^{\mathcal{R}}$, the coefficient $v^{\mathcal{R}}$ was unaffected by resolution scaling,

$$v^{\mathcal{R}} = v' \qquad v^{\mathcal{R}} \in I^{\mathcal{R}} \iff v^{\mathcal{R}} = v' \ \ (5.6b)$$

so

$$\begin{aligned} |v^{\mathcal{R}}| &= |v'| \\ &\geq 2^n && v' \in V' \implies |v'| \geq 2^n \ \text{(from the } \textbf{1. Unscaled} \text{ proof)} \\ &= 2^{n^{\mathcal{R}}} && \Lambda^{\mathcal{R}} = \Lambda \implies 2^{n^{\mathcal{R}}} = 2^n \\ \therefore v^{\mathcal{R}} &\in V^{\mathcal{R}} && V^{\mathcal{R}} = \{v \in I^{\mathcal{R}} : |v| \geq t^{\mathcal{R}} = 2^{n^{\mathcal{R}}}\} \ \ (5.2). \end{aligned}$$

However, if this is not the case, then either $v^{\mathcal{R}} \notin I^{\mathcal{R}}$ and therefore

$$v^{\mathcal{R}} \notin V^{\mathcal{R}} \qquad V^{\mathcal{R}} = \{v \in I^{\mathcal{R}} : |v| \geq t^{\mathcal{R}} = 2^{n^{\mathcal{R}}}\} \ \ (5.2);$$

or, alternatively, $v^{\mathcal{R}} \in I^{\mathcal{R}}$ but $v' \notin V'$, in which case

$$\begin{aligned} v^{\mathcal{R}} &= v' && v^{\mathcal{R}} \in I^{\mathcal{R}} \iff v^{\mathcal{R}} = v' \quad (5.6b) \\ |v^{\mathcal{R}}| &= |v'| \\ &< 2^n && v' \notin V' \implies |v'| < 2^n \text{ (from the } \textbf{1. Unscaled} \text{ proof)} \\ &= 2^{n^{\mathcal{R}}} && \Lambda^{\mathcal{R}} = \Lambda \implies 2^{n^{\mathcal{R}}} = 2^n \\ \therefore v^{\mathcal{R}} &\notin V^{\mathcal{R}} && V^{\mathcal{R}} = \{v \in I^{\mathcal{R}} : |v| \geq t^{\mathcal{R}} = 2^{n^{\mathcal{R}}}\} \quad (5.2) \end{aligned}$$

This gives us

$$\begin{aligned} v' \in V' \wedge v^{\mathcal{R}} \in I^{\mathcal{R}} &\implies v^{\mathcal{R}} \in V^{\mathcal{R}} \quad \text{and} \\ v^{\mathcal{R}} \notin I^{\mathcal{R}} \vee v' \notin V' &\implies v^{\mathcal{R}} \notin V^{\mathcal{R}}, \end{aligned}$$

so

$$v^{\mathcal{R}} \in V^{\mathcal{R}} \iff v' \in V' \wedge v^{\mathcal{R}} \in I^{\mathcal{R}}$$

and property (5.4b) is proven.

3. Quality Scaled

Given the quality scaled image $I^{\mathcal{Q}}$ and the correct watermarking parameters $\Lambda^{\mathcal{Q}} = \Lambda$, we let $v' \in I'$ denote a watermarked coefficient and $v^{\mathcal{Q}}$ the corresponding quality scaled coefficient, where $m \in \mathbb{Z}, m \geq 0$ denotes the number of magnitude bits lost from the coefficient v' during quality scaling.

If $v' \in V'$ and $v^{\mathcal{Q}} \neq 0$, then, because quality scaling does not remove any coefficients,

$$v^{\mathcal{Q}} \in I^{Q} \tag{5.7a}$$

and, from the definition of $v^{\mathcal{Q}}$ (2.20a) we have

$$\begin{aligned} v' = 0 \implies v^{\mathcal{Q}} &= sign(v')0 \\ &= 0 \end{aligned}$$

so

$$v^{\mathcal{Q}} \neq 0 \implies v' \neq 0.$$

Since $v^{\mathcal{Q}} \neq 0$, we conclude $v' \neq 0$ so $\exists k \in \mathbb{Z}$ such that $2^{k-1} \leq |v'| < 2^k$. Hence

$$\begin{aligned} v' \in V' \ \wedge \ 2^{k-1} \leq |v'| < 2^k & \\ |v'| \geq 2^n \ \wedge \ |v'| < 2^k & \qquad v' \in V' \implies |v'| \geq 2^n \text{ (from the } \textbf{1. Unscaled} \text{ proof)} \\ 2^n \leq 2^{k-1} & \qquad n, k \in \mathbb{Z} \end{aligned}$$

so

$$
\begin{aligned}
|v^{\mathcal{Q}}| &= \left\lfloor \frac{|v'|}{2^m} \right\rfloor 2^m + \lfloor 2^m \mathrm{r} \rfloor && \text{(2.20a) } v^{\mathcal{Q}} \neq 0 \implies \left\lfloor \frac{|v'|}{2^m} \right\rfloor \neq 0 \\
&\geq \left\lfloor \frac{2^{k-1}}{2^m} \right\rfloor 2^m + \lfloor 2^m \mathrm{r} \rfloor && |v'| \geq 2^{k-1} \\
&= 2^{k-1} + \lfloor 2^m \mathrm{r} \rfloor && \left\lfloor \frac{|v'|}{2^m} \right\rfloor \neq 0, |v'| < 2^k \implies m < k \implies \frac{2^{k-1}}{2^m} \in \mathbb{Z} \\
&\geq 2^{k-1} && \text{(2.20b) } \mathrm{r} \geq 0 \\
&\geq 2^n && 2^n \leq 2^{k-1} \\
&= 2^{n^{\mathcal{Q}}} && \Lambda^{\mathcal{Q}} = \Lambda \implies 2^{n^{\mathcal{Q}}} = 2^n \\
\therefore v^{\mathcal{Q}} &\in V^{\mathcal{Q}} && V^{\mathcal{Q}} = \{v \in I^{\mathcal{Q}} : |v| \geq t^{\mathcal{Q}} = 2^{n^{\mathcal{Q}}}\} \text{ (5.2)}
\end{aligned}
$$

If this is not the case, then either $v^{\mathcal{Q}} = 0$ and

$$
\begin{aligned}
v^{\mathcal{Q}} &< 2^{n^{\mathcal{Q}}} && n^{\mathcal{Q}} \in \mathbb{N} \\
\therefore v^{\mathcal{Q}} &\notin V^{\mathcal{Q}} && V^{\mathcal{Q}} = \{v^{\mathcal{Q}} \in I^{\mathcal{Q}} : |v^{\mathcal{Q}}| \geq 2^{n^{\mathcal{Q}}}\} \text{ (5.2)}
\end{aligned}
$$

or, alternatively, $v' \notin V'$ and $v^{\mathcal{Q}} \neq 0$, in which case $\exists k \in \mathbb{Z}$ such that $2^{k-1} \leq |v'| < 2^k$, and not all bits of v' were lost during scaling, so $m < k$. Thus

$$
\begin{aligned}
|v^{\mathcal{Q}}| &= \left\lfloor \frac{|v'|}{2^m} \right\rfloor 2^m + \lfloor 2^m \mathrm{r} \rfloor && \text{(2.20a) } v^{\mathcal{Q}} \neq 0 \implies \left\lfloor \frac{|v'|}{2^m} \right\rfloor \neq 0 \\
&\leq \left\lfloor \frac{\sum_{x=0}^{k-1} 2^x}{2^m} \right\rfloor 2^m + \lfloor 2^m \mathrm{r} \rfloor && v' \in \mathbb{Z}, |v'| < 2^k \Rightarrow |v'| \leq \sum_{x=0}^{k-1} 2^x \\
&= \left\lfloor \frac{\sum_{x=m}^{k-1} 2^x + \sum_{x=0}^{m-1} 2^x}{2^m} \right\rfloor 2^m + \lfloor 2^m \mathrm{r} \rfloor \\
&= \frac{\sum_{x=m}^{k-1} 2^x}{2^m} 2^m + \left\lfloor \frac{\sum_{x=0}^{m-1} 2^x}{2^m} \right\rfloor 2^m + \lfloor 2^m \mathrm{r} \rfloor && \frac{\sum_{x=m}^{k-1} 2^x}{2^m} \in \mathbb{Z} \\
&= \sum_{x=m}^{k-1} 2^x + \lfloor 2^m \mathrm{r} \rfloor && 0 \leq \frac{\sum_{x=0}^{m-1} 2^x}{2^m} < 1 \\
&< \sum_{x=m}^{k-1} 2^x + 2^m && \text{(2.20b) } \mathrm{r} < 1 \\
&= 2^k && m < k
\end{aligned}
$$

which, because $v' \notin V'$ so $|v'| < 2^n$(5.2) and $2^{k-1} \leq |v'|$ thus $2^{k-1} < 2^n$ and $k, n \in \mathbb{Z}$ so $k \leq n$,

$$
\begin{aligned}
&\leq 2^n \\
&= 2^{n^{\mathcal{Q}}} && \Lambda^{\mathcal{Q}} = \Lambda \implies 2^{n^{\mathcal{Q}}} = 2^n \\
\therefore v^{\mathcal{Q}} &\notin V^{\mathcal{Q}} && V^{\mathcal{Q}} = \{v^{\mathcal{Q}} \in I^{\mathcal{Q}} : |v^{\mathcal{Q}}| \geq t^{\mathcal{Q}}\} \text{ (5.2)}
\end{aligned}
$$

For both alternatives we have

$$v^{\mathcal{Q}} \notin V^{\mathcal{Q}}.$$

This gives us

$$\begin{aligned} v' \in V' \wedge v^{\mathcal{Q}} \neq 0 &\implies v^{\mathcal{Q}} \in V^{\mathcal{Q}} \quad \text{and} \\ v^{\mathcal{Q}} = 0 \vee v' \notin V' &\implies v^{\mathcal{Q}} \notin V^{\mathcal{Q}}, \end{aligned}$$

so

$$v^{\mathcal{Q}} \in V^{\mathcal{Q}} \iff v' \in V' \wedge v^{\mathcal{Q}} \neq 0$$

and property (5.4c) is proven.

D.3.2 Proof for Watermark Extraction

We show that the watermark extraction procedure, presented in section 5.1.3, satisfies the properties defined in that section provided the correct number of watermarked and missing bits are known. That is, given an unscaled, resolution scaled or quality scaled watermarked image and the correct watermarking parameters $\Lambda^* = \Lambda$, each extracted watermark element is identical to the corresponding embedded watermark except for those bits which were lost during scaling. Any watermark bits which were lost are given the value 0 unless all bits were lost, in which case the watermark element is not extracted.

Assuming that both the number of embedded bits and the number of missing bits are calculated correctly ($j^* = j$, $m^* = m$), we show that

- for the unscaled watermarked image, where $v' \in V'$ is the coefficient corresponding to $v \in V$ and $u^d = \mathsf{Extract}(v', I', \Lambda)$

$$\begin{aligned} &\text{if } j > 0 \text{ then } u^d = u \\ &\text{if } j = 0 \text{ then } \nexists u^d \end{aligned} \tag{5.9a}$$

- for the resolution scaled image, where $v^{\mathcal{R}} \in V^{\mathcal{R}}$ is the coefficient corresponding to $v' \in I'$ and $u^d = \mathsf{Extract}(v^{\mathcal{R}}, V^{\mathcal{R}}, \Lambda)$

$$\begin{aligned} &\text{if } j > 0 \text{ then } u^d = u \\ &\text{if } j = 0 \text{ then } \nexists u^d \end{aligned} \tag{5.9b}$$

- for the quality scaled image, where $v^{\mathcal{Q}} \in V^{\mathcal{Q}}$, the quality scaled coefficient corresponding to $v' \in V'$, has m missing least significant bits and $u^d = \mathsf{Extract}(v^{\mathcal{Q}}, V^{\mathcal{Q}}, \Lambda)$

$$\begin{aligned} &\text{if } m < j \text{ then } u^d = \lfloor \frac{u}{2^m} \rfloor 2^m \\ &\text{if } m \geq j \text{ then } \nexists u^d \end{aligned} \tag{5.9c}$$

Let

$$u^d = \mathsf{Extract}(v^*, I^*, \Lambda^*) = \begin{cases} |v^*| - Q_{2^{j^*}}(v^*) - \lfloor 2^{m^*} \mathrm{r} \rfloor & \text{if } m^* < j^* \\ \nexists & \text{if } m^* \geq j^*, \end{cases} \tag{5.8}$$

where $0 \leq r < 1$ is the JPEG2000 coefficient reconstruction parameter and the number of missing bits m^* and the number of embedded bits j^* have been computed from (v^*, I^*, Λ^*) for each selected coefficient $v^* \in V^*$.

Assume that for any scaled image $I^{\mathcal{F}} = \mathcal{F}(\mathsf{Embed}(I, \Lambda^{\mathcal{F}}))$, that has undergone only JPEG2000 resolution and/or quality scaling, the correct number of missing bits can be calculated $m^* = m$ and the correct number of watermark bits can be calculated $j^* = j$.

1. Unscaled

If $I' = \mathsf{Embed}(I, \Lambda)$ is the unscaled watermarked image, $v' \in V'$ is the coefficient corresponding to $v \in V$ and $u^d = \mathsf{Extract}(v', I', \Lambda)$, then

I' is unscaled, so no bits are missing: $m' = 0$, and

$\Lambda' = \Lambda$ and no additional processing occurred, so $j' = j$.

If $j > 0$ then since $j' = j$ and $m' = 0$, we must have $m' < j'$ and hence

$$\begin{aligned}
u^d &= |v'| - Q_{2^{j'}}(v') - \left\lfloor 2^{m'} \mathrm{r} \right\rfloor && (5.8)\ m' < j' \\
&= |v'| - Q_{2^j}(v') - \lfloor \mathrm{r} \rfloor && m' = 0, j' = j \\
&= |v'| - Q_{2^j}(v') && (2.20\mathrm{b}) \\
&= |v'| - \left\lfloor \frac{|v'|}{2^j} \right\rfloor 2^j && (5.3\mathrm{a}) \\
&= Q_{2^j}(v) + u - \left\lfloor \frac{Q_{2^j}(v) + u}{2^j} \right\rfloor 2^j && (5.3\mathrm{c}) \\
&= \left\lfloor \frac{|v|}{2^j} \right\rfloor 2^j + u - \left\lfloor \frac{\left\lfloor \frac{|v|}{2^j} \right\rfloor 2^j + u}{2^j} \right\rfloor 2^j && (5.3\mathrm{a}) \\
&= \left\lfloor \frac{|v|}{2^j} \right\rfloor 2^j + u - \left\lfloor \left\lfloor \frac{|v|}{2^j} \right\rfloor + \frac{u}{2^j} \right\rfloor 2^j \\
&= \left\lfloor \frac{|v|}{2^j} \right\rfloor 2^j + u - \left\lfloor \frac{|v|}{2^j} \right\rfloor 2^j && \left\lfloor \frac{|v|}{2^j} \right\rfloor \in Z,\ 0 \leq \frac{u}{2^j} < 1\ \ (5.3\mathrm{b}) \\
&= u
\end{aligned}$$

If $j = 0$ then since $j' = j$ and $m' = 0$, we must have $m' = j'$ and hence

$$u^d = \nexists \qquad (5.8), m' \geq j'$$

and property (5.9a) is proven.

2. Resolution Scaled

If $I^{\mathcal{R}} = \mathcal{R}(\mathsf{Embed}(I, \Lambda))$ is the resolution scaled watermarked image, $v^{\mathcal{R}} \in V^{\mathcal{R}}$ is the coefficient corresponding to $v' \in V'$ and $u^d = \mathsf{Extract}(v^{\mathcal{R}}, I^{\mathcal{R}}, \Lambda)$, then

$I^{\mathcal{R}}$ is only resolution scaled, so no bits are missing: $m^{\mathcal{R}} = 0$ and $v^{\mathcal{R}} = v'$ (5.6b) and

$\Lambda^{\mathcal{R}} = \Lambda$ and only resolution scaling was applied, so $j^{\mathcal{R}} = j \in \mathbb{N}$.

If $j > 0$ then since $j^{\mathcal{R}} = j$ and $m^{\mathcal{R}} = 0$, we must have $m^{\mathcal{R}} < j^{\mathcal{R}}$ and hence

$$\begin{aligned} u^d &= |v^{\mathcal{R}}| - Q_{2^{j^{\mathcal{R}}}}(v^{\mathcal{R}}) - \left\lfloor 2^{m^{\mathcal{R}}} \mathrm{r} \right\rfloor && (5.8)\ m^{\mathcal{R}} < j^{\mathcal{R}} \\ &= |v'| - Q_{2^{j^{\mathcal{R}}}}(v') - \left\lfloor 2^{m^{\mathcal{R}}} \mathrm{r} \right\rfloor && v^{\mathcal{R}} = v' \\ &= |v'| - Q_{2^j}(v') - \lfloor \mathrm{r} \rfloor && m^{\mathcal{R}} = 0, j^{\mathcal{R}} = j \\ &= u && \text{from the } \textbf{1. Unscaled} \text{ proof} \end{aligned}$$

If $j = 0$ then since $j^{\mathcal{R}} = j$ and $m^{\mathcal{R}} = 0$, we must have $m^{\mathcal{R}} = j^{\mathcal{R}}$ and hence

$$u^d = \nexists \qquad (5.8), m^{\mathcal{R}} \geq j^{\mathcal{R}}$$

and property (5.9b) is proven.

3. Quality Scaled

If $I^{\mathcal{Q}} = \mathcal{Q}(\mathsf{Embed}(I, \Lambda))$ is the quality scaled watermarked image, $v^{\mathcal{Q}} \in V^{\mathcal{Q}}$ is the coefficient corresponding to $v' \in V'$ and $u^d = \mathsf{Extract}(v^{\mathcal{Q}}, I^{\mathcal{Q}}, \Lambda)$, then

the correct number of missing bits is calculated $m^{\mathcal{Q}} = m, \quad 0 \leq m,$

$0 \neq v^{\mathcal{Q}} \in \mathbb{Z}$ so $\exists k \in \mathbb{N}, 2^{k-1} \leq |v^{\mathcal{Q}}| < 2^k,$

$v^{\mathcal{Q}} \in V^{\mathcal{Q}} \implies m < k$ (5.2) and

$\Lambda^{\mathcal{Q}} = \Lambda$ and only quality scaling was applied, so $j^{\mathcal{Q}} = j \in \mathbb{N}$.

Now

$$u^d = \begin{cases} |v^{\mathcal{Q}}| - Q_{2^{j^{\mathcal{Q}}}}(v^{\mathcal{Q}}) - \left\lfloor 2^{m^{\mathcal{Q}}} \mathrm{r} \right\rfloor & \text{if } m^{\mathcal{Q}} < j^{\mathcal{Q}} \\ \nexists & \text{if } m^{\mathcal{Q}} \geq j^{\mathcal{Q}}, \end{cases} \tag{5.8}$$

which, since $m^{\mathcal{Q}} = m$ and $j^{\mathcal{Q}} = j$

$$= \begin{cases} |v^{\mathcal{Q}}| - Q_{2^j}(v^{\mathcal{Q}}) - \lfloor 2^m \mathrm{r} \rfloor & \text{if } m < j \\ \nexists & \text{if } m \geq j. \end{cases}$$

Expanding the case in which $m < j$ we have

$$
\begin{aligned}
u^d &= |v^{\mathcal{Q}}| - Q_{2^j}(v^{\mathcal{Q}}) - \lfloor 2^m \mathrm{r} \rfloor \\
&= |v^{\mathcal{Q}}| - \left\lfloor \frac{|v^{\mathcal{Q}}|}{2^j} \right\rfloor 2^j - \lfloor 2^m \mathrm{r} \rfloor && \text{(5.3a)} \\
&= \left\lfloor \frac{|v'|}{2^m} \right\rfloor 2^m + \lfloor 2^m \mathrm{r} \rfloor - \left\lfloor \frac{\left\lfloor \frac{|v'|}{2^m} \right\rfloor 2^m + \lfloor 2^m \mathrm{r} \rfloor}{2^j} \right\rfloor 2^j - \lfloor 2^m \mathrm{r} \rfloor && \text{(2.20a)} \;\; v^{\mathcal{Q}} \neq 0 \\
&= \left\lfloor \frac{|v'|}{2^m} \right\rfloor 2^m - \left\lfloor \frac{\left\lfloor \frac{|v'|}{2^m} \right\rfloor 2^m + \lfloor 2^m \mathrm{r} \rfloor}{2^j} \right\rfloor 2^j \\
&= \left\lfloor \frac{|v'|}{2^m} \right\rfloor 2^m - \left\lfloor \frac{\left\lfloor \frac{|v'|}{2^m} \right\rfloor + \frac{\lfloor 2^m \mathrm{r} \rfloor}{2^m}}{2^{j-m}} \right\rfloor 2^j && \left\lfloor \frac{|v|}{2^m} \right\rfloor \in \mathbb{Z},\; 0 \leq \frac{\lfloor 2^m \mathrm{r} \rfloor}{2^m} < 1,\; 2^{j-m} \in \mathbb{N} \\
&= \left\lfloor \frac{|v'|}{2^m} \right\rfloor 2^m - \left\lfloor \frac{\left\lfloor \frac{|v'|}{2^m} \right\rfloor}{2^{j-m}} \right\rfloor 2^j && \text{Lemma D.2.2} \\
&= \left\lfloor \frac{|v'|}{2^m} \right\rfloor 2^m - \left\lfloor \frac{\left\lfloor \frac{|v'|}{2^m} \right\rfloor 2^m}{2^j} \right\rfloor 2^j && 2^j = 2^{j-m} 2^m,\; 2^{j-m} \in \mathbb{N},\; 2^m \in \mathbb{Z} \\
&= \left\lfloor \frac{|v'|}{2^m} \right\rfloor 2^m - \left\lfloor \frac{|v'|}{2^j} \right\rfloor 2^j && \text{Lemma D.2.3} \\
&= \left\lfloor \frac{Q_{2^j}(v) + u}{2^m} \right\rfloor 2^m - \left\lfloor \frac{Q_{2^j}(v) + u}{2^j} \right\rfloor 2^j && \text{(5.3c)} \\
&= \left\lfloor \frac{\left\lfloor \frac{|v|}{2^j} \right\rfloor 2^j + u}{2^m} \right\rfloor 2^m - \left\lfloor \frac{\left\lfloor \frac{|v|}{2^j} \right\rfloor 2^j + u}{2^j} \right\rfloor 2^j && \text{(5.3a)} \\
&= \left\lfloor \frac{\left\lfloor \frac{|v|}{2^j} \right\rfloor 2^j + u}{2^m} \right\rfloor 2^m - \left\lfloor \frac{|v|}{2^j} \right\rfloor 2^j && \left\lfloor \frac{|v|}{2^j} \right\rfloor \in \mathbb{Z},\; 0 \leq \frac{u}{2^j} < 1 \;\; \text{(5.3)} \\
&= \left\lfloor \left\lfloor \frac{|v|}{2^j} \right\rfloor 2^{j-m} + \frac{u}{2^m} \right\rfloor 2^m - \left\lfloor \frac{|v|}{2^j} \right\rfloor 2^j \\
&= \left\lfloor \frac{|v|}{2^j} \right\rfloor 2^{j-m} 2^m + \left\lfloor \frac{u}{2^m} \right\rfloor 2^m - \left\lfloor \frac{|v|}{2^j} \right\rfloor 2^j && \left\lfloor \frac{|v|}{2^j} \right\rfloor 2^{j-m} \in \mathbb{Z} \\
&= \left\lfloor \frac{|v|}{2^j} \right\rfloor 2^j + \left\lfloor \frac{u}{2^m} \right\rfloor 2^m - \left\lfloor \frac{|v|}{2^j} \right\rfloor 2^j \\
&= \left\lfloor \frac{u}{2^m} \right\rfloor 2^m
\end{aligned}
$$

so the entire expression simplifies to

$$u^d = \begin{cases} \left\lfloor \frac{u}{2^m} \right\rfloor 2^m & \text{if } m < j \\ \nexists & \text{if } m \geq j \end{cases}$$

and property (5.9c) is proven.

D.3.3 Proof for Indexing

We show that the indexing formula, in section 5.1.4.1, satisfies the properties defined in that section. That is, given the correct watermarking parameters $\Lambda^* = \Lambda$, the index of any selected coefficient from a scaled but otherwise unmodified watermarked image will be equal to that of the corresponding coefficient in the original image:

- for the unscaled watermarked image $I' = \mathsf{Embed}(I, \Lambda)$, where $v'_{i'} \in I'$ is the coefficient corresponding to $v_i \in I$

$$i' = i \tag{5.13a}$$

- for the resolution scaled image, where $v^{\mathcal{R}}_{i^{\mathcal{R}}} \in I^{\mathcal{R}}$ is the coefficient corresponding to $v'_{i'} \in I'$

$$i^{\mathcal{R}} = i' \tag{5.13b}$$

- for the quality scaled image, where $v^{\mathcal{Q}}_{i^{\mathcal{Q}}} \in I^{\mathcal{Q}}$, is the coefficient corresponding to $v'_{i'} \in I'$

$$i^{\mathcal{Q}} = i' \tag{5.13c}$$

For a given image I^*, watermark parameters Λ^* and a selected coefficient $v^* \in V^*$ in component c^*, resolution r^*, subband orientation o^*, column x^* and row y^*, let

$$\begin{aligned} i^* &= \mathsf{Index}(v^*, I^*) \\ &= c^* X^*[r^*] Y^*[r^*] + (C^* - c^*) X^*[r^* - 1] Y^*[r^* - 1] + \\ &\qquad \sum_{a=0}^{a=o^*-1} X^*[r^*, a] Y^*[r^*, a] + y^* X^*[r^*, o^*] + x^* \end{aligned} \tag{5.12}$$

1. Unscaled

If $I' = \mathsf{Embed}(I, \Lambda)$ is the unscaled watermarked image, $v'_{i'} \in I'$ is the coefficient corresponding to $v_i \in I$ then all image dimensions are unchanged, so $X' = X$, $Y' = Y$, $C' = C$ and $R' = R$ thus, substituting into (2.14), (5.10), and (5.11), all resolutions and

subbands also have the same dimensions:

$$X'[r] = X[r]$$
$$Y'[r] = Y'[r]$$

and

$$X'[r, o] = X[r, o]$$
$$Y'[r, o] = Y[r, o],$$

for $0 \leq r < R, o \in \{0, 1, 2\}$. Furthermore, watermarking does not alter the position of the coefficient within the image, so if v_i has position (c, r, o, x, y) in the original image then $v'_{i'}$ has position $(c', r', s', x', y') = (c, r, o, x, y)$ in the watermarked image.

Thus

$$\begin{aligned}
i' &= \mathsf{Index}(v', I') \\
&= c'X'[r']Y'[r'] + (C' - c')X'[r' - 1]Y'[r' - 1] + \sum_{a=0}^{a=o'-1} X'[r', a]Y'[r', a] + y'X'[r', o'] + x' \\
&= cX[r]Y[r] + (C - c)X[r - 1]Y[r - 1] + \sum_{a=0}^{a=o-1} X[r, a]Y[r, a] + yX[r, o] + x \\
&= \mathsf{Index}(v, I) \\
&= i
\end{aligned}$$

and property (5.13a) is proven.

2. Resolution Scaled

If $I^{\mathcal{R}} = \mathcal{R}(\mathsf{Embed}(I, \Lambda))$ is the resolution scaled image, and $v_{i^{\mathcal{R}}}^{\mathcal{R}} \in I^{\mathcal{R}}$ is the coefficient corresponding to $v'_i \in I'$, then the number of components is unchanged $C^{\mathcal{R}} = C$ but the K highest image resolutions have been lost so $R^{\mathcal{R}} = R - K$, and the dimensions of each component are reduced to $X^{\mathcal{R}} = \left\lceil \frac{X}{2^K} \right\rceil$ and $Y^{\mathcal{R}} = \left\lceil \frac{Y}{2^K} \right\rceil$.

The loss of K resolutions from the image will not affect the dimensions (2.14) of any

of the remaining resolutions $r < R - K$

$$\begin{aligned} X^{\mathcal{R}}[r] &= \left\lceil \frac{X^{\mathcal{R}}}{2^{R^{\mathcal{R}}-r-1}} \right\rceil \\ &= \left\lceil \frac{\left\lceil \frac{X}{2^K} \right\rceil}{2^{R-K-r-1}} \right\rceil \\ &= \left\lceil \frac{\left\lceil \frac{X}{2^K} \right\rceil 2^K}{2^{R-r-1}} \right\rceil \\ &= \left\lceil \frac{X}{2^{R-r-1}} \right\rceil \qquad 2^R = 2^{R-K-r-1}2^K, r < R-K \implies 2^{R-K-r-1} \in \mathbb{N} \\ &= X[r] \end{aligned}$$

and, similarly,

$$Y^{\mathcal{R}}[r] = Y[r]$$

Thus, from equations (5.10) and (5.11),

$$\begin{aligned} X^{\mathcal{R}}[r,o] &= X[r,o] \\ Y^{\mathcal{R}}[r,o] &= Y[r,o], \end{aligned}$$

for $r < R - K$ and $o \in \{0,1,2\}$.

Furthermore, the loss of resolution layers during scaling does not alter the positions of any remaining coefficients, so if v'_i has position (c,r,o,x,y) then the corresponding selected coefficient $v^{\mathcal{R}}_{i^{\mathcal{R}}} \in V^{\mathcal{R}} \subseteq I^{\mathcal{R}}$ has position $(c^{\mathcal{R}}, r^{\mathcal{R}}, o^{\mathcal{R}}, x^{\mathcal{R}}, y^{\mathcal{R}}) = (c,r,o,x,y)$ thus

$$\begin{aligned} i^{\mathcal{R}} &= c^{\mathcal{R}} X^{\mathcal{R}}[r^{\mathcal{R}}] Y^{\mathcal{R}}[r^{\mathcal{R}}] + (C^{\mathcal{R}} - c^{\mathcal{R}}) X^{\mathcal{R}}[r^{\mathcal{R}} - 1] Y^{\mathcal{R}}[r^{\mathcal{Q}} - 1] \\ &\qquad + \sum_{o=0}^{o=o^{\mathcal{R}}-1} X^{\mathcal{R}}[r^{\mathcal{R}}, o] Y^{\mathcal{R}}[r^{\mathcal{R}}, o] + y^{\mathcal{R}} X^{\mathcal{R}}[r^{\mathcal{R}}, o^{\mathcal{R}}] + x^{\mathcal{R}} \\ &= cX[r]Y[r] + (C-c)X[r-1]Y[r-1] + \sum_{a=0}^{a=o-1} X[r,a]Y[r,a] + yX[r,o] + x \\ &= i' \end{aligned}$$

and property (5.13b) is proven.

3. Quality Scaled

If $I^{\mathcal{Q}} = \mathcal{Q}(\mathsf{Embed}(I, \Lambda))$ is the quality scaled image, and $v^{\mathcal{Q}}_{i^{\mathcal{Q}}} \in I^{\mathcal{Q}}$ is the coefficient corresponding to $v'_i \in I'$ then, because even completely lost coefficients are still represented,

all image dimensions are unchanged so $X^{\mathcal{Q}} = X$, $Y^{\mathcal{Q}} = Y$, $C^{\mathcal{Q}} = C$ and $R^{\mathcal{Q}} = R$ thus, from (2.14), (5.10), and (5.11), all resolutions and subbands also have the same dimensions:

$$X^{\mathcal{Q}}[r] = X[r]$$
$$Y^{\mathcal{Q}}[r] = Y'[r]$$

and

$$X^{\mathcal{Q}}[r, o] = X[r, o]$$
$$Y^{\mathcal{Q}}[r, o] = Y[r, o],$$

for $0 \leq r < R, o \in \{0, 1, 2\}$.

Nor does quality scaling affect the coefficients' positions, so if v'_i has position (c, r, o, x, y) then $v^{\mathcal{Q}}_{i^{\mathcal{Q}}}$ has position $(c^{\mathcal{Q}}, r^{\mathcal{Q}}, o^{\mathcal{Q}}, x^{\mathcal{Q}}, y^{\mathcal{Q}}) = (c, r, o, x, y)$

So, as with the unscaled image,

$$\begin{aligned} i^{\mathcal{Q}} &= c^{\mathcal{Q}} X^{\mathcal{Q}}[r^{\mathcal{Q}}] Y^{\mathcal{Q}}[r^{\mathcal{Q}}] + (C^{\mathcal{Q}} - c^{\mathcal{Q}}) X^{\mathcal{Q}}[r^{\mathcal{Q}} - 1] Y^{\mathcal{Q}}[r^{\mathcal{Q}} - 1] \\ &\qquad + \sum_{a=0}^{a=o^{\mathcal{Q}}-1} X^{\mathcal{Q}}[r^{\mathcal{Q}}, a] Y^{\mathcal{Q}}[r^{\mathcal{Q}}, a] + y^{\mathcal{Q}} X^{\mathcal{Q}}[r^{\mathcal{Q}}, o^{\mathcal{Q}}] + x^{\mathcal{Q}} \\ &= cX[r]Y[r] + (C - c)X[r - 1]Y[r - 1] + \sum_{a=0}^{a=o-1} X[r, o]Y[r, o] + yX[r, o] + x \\ &= i' \end{aligned}$$

and property (5.13c) is proven.

D.3.4 Proof of Watermark Element Generation

We show that the watermark generation function defined in section 5.1.4.2 satisfies the requirement in section 5.1.1, that

1. given the original image, where v is a selected coefficient, embedding will not disturb the most significant bit of v

   $$\exists j \in \mathbb{N} \text{ s.t. } 0 \leq u < 2^j \leq |v|. \tag{5.5}$$

   and those of section 5.1.4.2, that the same watermark element will be generated for corresponding coefficients, from the original and an unscaled, resolution scaled or quality scaled version.

2. for the unscaled watermarked image, where $v'_i \in V'$ is the coefficient corresponding to $v_i \in V$

   $$\mathsf{G}(v'_i, i, \Lambda, I') = \mathsf{G}(v_i, i, \Lambda, I) \tag{5.19a}$$

3. for the resolution scaled image, where $v_i^{\mathcal{R}} \in V^{\mathcal{R}}$ is the coefficient corresponding to $v_i' \in V'$ and $v \in V$

$$\mathsf{G}(v_i^{\mathcal{R}}, i, \Lambda, I^{\mathcal{R}}) = \mathsf{G}(v_i, i, \Lambda, I), \text{ and} \tag{5.19b}$$

4. for the quality scaled image, where $v_i^{\mathcal{Q}} \in V^{\mathcal{Q}}$, is the quality scaled coefficient corresponding to $v_i' \in V'$ and $v \in V$

$$\mathsf{G}(v_i^{\mathcal{Q}}, i, \Lambda, I^{\mathcal{Q}}) = \mathsf{G}(v_i, i, \Lambda, I). \tag{5.19c}$$

Let the watermark generation function be

$$\mathsf{G}(v_{i^*}^*, i^*, \Lambda^*, I^*) = \lfloor \alpha^* 2^{-(h+1)} \bar{v}_{i^*}^* w_{i^*}^* \rfloor, \tag{5.18}$$

where $w_{i^*}^* = \mathsf{g}(sk^*, i^*)$ (5.15), is in the range $0 \leq w_i^* < 2^h$ for some $h \in \mathbb{N}$, is pseudo-randomly generated using the key $sk^* \in \Lambda^*$ and the index i^*, where $\bar{v}_{i^*}^*$ is the smallest non-negative-integer power of two that exceeds $v_{i^*}^*$ (5.16) and where $\alpha^* \in \Lambda^*$ is a global strength parameter in the range $0 \leq \alpha^* < 1$.

1. Original

If I is the original image and $v_i \in V$ is a selected coefficient then

$$\mathsf{G}(v_i, i, \Lambda, I) = \lfloor \alpha 2^{-(h+1)} \bar{v}_i w_i \rfloor$$

which, since $0 \leq \alpha,\ 0 < 2^{-(h+1)},\ 0 < \bar{v}_i$ and $0 \leq w_i$

$$\geq 0.$$

Let $2^{j_i} = \bar{\mathsf{G}}(v_i, i, \Lambda, I), j_i \in \mathbb{N}$. Then

$$\begin{aligned} \lfloor 2^{j_i - 1} \rfloor \leq |\mathsf{G}(v_i, i, \Lambda, I)| < 2^{j_i} \qquad & (5.16) \\ \lfloor 2^{j_i - 1} \rfloor \leq \mathsf{G}(v_i, i, \Lambda, I) < 2^{j_i} \qquad & \mathsf{G}(v_i, i, \Lambda, I) \geq 0 \\ \mathsf{G}(v_i, i, \Lambda, I) < 2^{j_i}, j_i \in \mathbb{N} & \end{aligned}$$

and

$$\begin{aligned}
\mathsf{G}(v_i,i,\Lambda,I) &= \lfloor \alpha 2^{-(h+1)}\bar{v}_i w_i \rfloor \\
&\leq \alpha 2^{-(h+1)}\bar{v}_i w_i \\
&< 2^{-(h+1)}\bar{v}_i w_i && \alpha < 1 \\
&< 2^{-1}\bar{v}_i && w_i < 2^h \\
\lfloor 2^{j_i-1} \rfloor &< 2^{-1}\bar{v}_i && \lfloor 2^{j_i-1} \rfloor \leq \mathsf{G}(v_i,i,\Lambda,I) \\
\lfloor \frac{\lfloor \frac{2^{j_i}}{2} \rfloor 2}{1} \rfloor &< \bar{v}_i \\
\lfloor \frac{2^{j_i}}{1} \rfloor &< \bar{v}_i && \text{Lemma D.2.2} \\
2^{j_i} &< \bar{v}_i && j_i \in \mathbb{N} \implies \frac{2^{j_i}}{1} \in \mathbb{N} \\
2^{j_i} &< 2^{k_i} && \text{Let} k_i \in \mathbb{N} : \bar{v}_i = 2^{k_i} \quad (5.16) \\
2^{j_i} &\leq 2^{k_i-1} && j_i, k_i \in \mathbb{N} \\
2^{j_i} &\leq |v_i| && \bar{v}_i = 2^{k_i} \implies 2^{k_i-1} \leq |v_i| \quad (5.16)
\end{aligned}$$

So $j_i \in \mathbb{N}$ and $0 \leq \mathsf{G}(v_i,i,\Lambda,I) < 2^{j_i} \leq |v_i|$. Therefore $\exists j_i = \log_2(\bar{\mathsf{G}}(v_i,i,\Lambda,I)) \in \mathbb{N}$ such that $0 \leq \mathsf{G}(v_i,i,\Lambda,I) < 2^{j_i} \leq |v_i|$ and property (5.5) is proven.

2. Unscaled

If I' is the unscaled watermarked image, where $v'_i \in V'$ is the coefficient corresponding to $v_i \in V$ then, for the original coefficient, we let $\bar{v}_i = 2^{k_i}$ where $sk_i \in \mathbb{N}$ so

$$\begin{aligned}
2^{k_i-1} &\leq |\, v_i \,| < 2^{k_i} && (5.16) \\
2^{k_i-1} &\leq |\, v_i \,| \leq 2^{k_i} - 1 && v_i \in \mathbb{Z} \\
2^{k_i-1} &\leq |\, v_i \,| \leq \sum_{x=1}^{k_i-1} 2^x.
\end{aligned}$$

Consider the magnitude of the coefficient $v'_i \in I'$

$$\begin{aligned}
Q_{2^{j_i}}(v_i) + u_i = |\, v'| &= Q_{2^{j_i}}(v_i) + u_i && (5.3c) \\
\left\lfloor \frac{|v_i|}{2^{j_i}} \right\rfloor 2^{j_i} + u_i = |\, v'_i| &= \left\lfloor \frac{|v_i|}{2^{j_i}} \right\rfloor 2^{j_i} + u_i && (5.3a) \\
\left\lfloor \frac{2^{k_i-1}}{2^j} \right\rfloor 2^j + u_i \leq |\, v'_i| &\leq \left\lfloor \frac{\sum_{x=1}^{k_i-1} 2^x}{2^{j_i}} \right\rfloor 2^{j_i} + u_i && 2^{k_i-1} \leq |v_i| \leq \sum_{x=1}^{k_i-1} 2^x.
\end{aligned}$$

Now, $j_i \in \mathbb{Z},\ 2^{j_i} \le |v_i|$ and $k_i \in \mathbb{Z},\ |v_i| < 2^{k_i}$ so $j_i \le k_i - 1$. Thus

$$\begin{aligned} 2^{k_i-1} + u_i \le |\,v_i'| &\le \sum_{x=j}^{k_i-1} 2^x + u_i \\ 2^{k_i-1} \le |\,v_i'| &< \sum_{x=j}^{k_i-1} 2^x + 2^{j_i} && 0 \le u_i < 2^{j_i} \quad (5.3b) \\ 2^{k_i-1} \le |\,v_i'| &< 2^{k_i} \\ \bar{v}_i' &= 2^{k_i}, && (5.16) \end{aligned}$$

so

$$\bar{v}_i' = \bar{v}_i$$

The sequence element

$$\begin{aligned} w_i' &= \mathsf{g}(sk', i) && (5.15) \\ &= \mathsf{g}(sk, i) && \Lambda' = \Lambda \implies sk' = sk \\ &= w_i. \end{aligned}$$

Thus,

$$\begin{aligned} \mathsf{G}(v_i', i, \Lambda, I') &= \lfloor \alpha 2^{-(h+1)} \bar{v}_i' w_i' \rfloor \\ &= \lfloor \alpha 2^{-(h+1)} \bar{v}_i w_i \rfloor \\ &= \mathsf{G}(v_i, i, \Lambda, I) \end{aligned}$$

and property (5.19a) is proven.

3. Resolution Scaled

If $I^{\mathcal{R}}$ is the resolution scaled image and $v_i^{\mathcal{R}} \in V^{\mathcal{R}}$ is the coefficient corresponding to $v_i' \in V'$ and $v_i \in V$ then, since $v^{\mathcal{R}} \in V^{\mathcal{R}}$, the coefficient was unaffected by resolution scaling so

$$\begin{aligned} v_i^{\mathcal{R}} &= v_i' && (5.6b) \\ \bar{v}_i^{\mathcal{R}} &= \bar{v}_i' \\ &= \bar{v}_i. && \text{from the } \textbf{2. Unscaled} \text{ proof} \end{aligned}$$

The sequence element

$$\begin{aligned} w_i^{\mathcal{R}} &= \mathsf{g}(sk^{\mathcal{R}}, i) && (5.15)\\ &= \mathsf{g}(sk, i) && \Lambda^{\mathcal{R}} = \Lambda \implies sk^{\mathcal{R}} = sk \\ &= w_i. \end{aligned}$$

Thus,

$$\begin{aligned} \mathsf{G}(v_i^{\mathcal{R}}, i, \Lambda, I^{\mathcal{R}}) &= \lfloor \alpha 2^{-(h+1)} \bar{v}_i^{\mathcal{R}} w_i^{\mathcal{R}} \rfloor \\ &= \lfloor \alpha 2^{-(h+1)} \bar{v}_i w_i \rfloor \\ &= \mathsf{G}(v_i, i, \Lambda, I) \end{aligned}$$

and property (5.19b) is proven.

4. Quality Scaled

We let $k_i \in \mathbb{N}$ s.t. $\bar{v}_i' = 2^{k_i}$ and use $m_i \in \mathbb{Z}, m_i \geq 0$ to denote the number of missing magnitude bits and $\mathrm{r} \in \mathbb{R}, 0 \leq \mathrm{r} < 1$ to denote the coefficient reconstruction parameter at the decoder.

If $I^{\mathcal{Q}}$ is the quality scaled image and $v_i^{\mathcal{Q}} \in V^{\mathcal{Q}}$ is the coefficient corresponding to $v_i' \in V'$ and $v_i \in V$ then,

$$|v_i^{\mathcal{Q}}| = \left\lfloor \frac{|v_i'|}{2^{m_i}} \right\rfloor 2^{m_i} + \lfloor 2^{m_i} \mathrm{r} \rfloor$$

so, since $2^{k_i - 1} \leq |v_i'| \leq \sum_{x=1}^{k_i - 1} 2^x < 2^{k_i}$,

$$\left\lfloor \frac{2^{k_i-1}}{2^{m_i}} \right\rfloor 2^{m_i} + \lfloor 2^{m_i} \mathrm{r} \rfloor \leq |v_i^{\mathcal{Q}}| \leq \left\lfloor \frac{\sum_{x=1}^{k_i-1} 2^x}{2^{m_i}} \right\rfloor 2^{m_i} + \lfloor 2^{m_i} \mathrm{r} \rfloor .$$

If all coefficient bits have been lost, $m_i \geq k_i$, $v_i^{\mathcal{Q}} = 0$ and thus $v_i^{\mathcal{Q}} \notin V^{\mathcal{Q}}$, so we need only consider the case in which at least one bit remains, i.e. $m_i \leq k_i - 1$:

$$\begin{aligned} \left\lfloor \frac{2^{k_i-1}}{2^{m_i}} \right\rfloor 2^{m_i} + \lfloor 2^{m_i} \mathrm{r} \rfloor \leq |v_i^{\mathcal{Q}}| &\leq \left\lfloor \frac{\sum_{x=1}^{k_i-1} 2^x}{2^{m_i}} \right\rfloor 2^{m_i} + \lfloor 2^{m_i} \mathrm{r} \rfloor \\ 2^{k_i-1} + \lfloor 2^{m_i} \mathrm{r} \rfloor \leq |v_i^{\mathcal{Q}}| &\leq \sum_{x=m_i}^{k_i-1} 2^x + \lfloor 2^{m_i} \mathrm{r} \rfloor && m_i \leq k_i - 1 \\ 2^{k_i-1} \leq |v_i^{\mathcal{Q}}| &< \sum_{x=m_i}^{k_i-1} 2^x + 2^{m_i} && 0 \leq \mathrm{r} < 1 \\ 2^{k_i-1} \leq |v_i^{\mathcal{Q}}| &< 2^{k_i} \\ \bar{v}_i^{\mathcal{Q}} &= 2^{k_i} && (5.16) \end{aligned}$$

so

$$\begin{aligned}\bar{v}_i^{\mathcal{Q}} &= \bar{v}'_i \\ &= \bar{v}_i. \qquad \text{from the } \textbf{2. Unscaled} \text{ proof}\end{aligned}$$

The sequence element

$$\begin{aligned}w_i^{\mathcal{Q}} &= \mathsf{g}(sk^{\mathcal{Q}}, i) && (5.15)\\ &= \mathsf{g}(sk, i) && \Lambda^{\mathcal{Q}} = \Lambda \implies sk^{\mathcal{Q}} = sk\\ &= w_i.\end{aligned}$$

Thus,

$$\begin{aligned}\mathsf{G}(v_i^{\mathcal{Q}}, i, \Lambda, I^{\mathcal{Q}}) &= \lfloor \alpha 2^{-(h+1)} \bar{v}_i^{\mathcal{Q}} w_i^{\mathcal{Q}} \rfloor \\ &= \lfloor \alpha 2^{-(h+1)} \bar{v}_i w_i \rfloor \\ &= \mathsf{G}(v_i, i, \Lambda, I)\end{aligned}$$

and property (5.19c) is proven.

D.3.5 Proof of Candidate Truncation

We show that the candidate truncation defined in section 5.1.4.2 satisfies the requirements of that section. Specifically that, given the correct watermarking parameters, number of missing bits and number of watermark bits, each truncated candidate element will exactly match the corresponding extracted watermark element

- for the unscaled watermarked image, where $v'_i \in V'$ is the coefficient corresponding to $v_i \in V$

$$\begin{aligned}&\text{if } j > 0 \text{ then } u_{i\mathcal{Q}}^c = u_i \\ &\text{if } j = 0 \text{ then } \nexists u_{i\mathcal{Q}}^c\end{aligned} \qquad (5.22a)$$

- for the resolution scaled image, where $v_i^{\mathcal{R}} \in V^{\mathcal{R}}$ is the coefficient corresponding to $v'_i \in V'$

$$\begin{aligned}&\text{if } j > 0 \text{ then } u_{i\mathcal{Q}}^c = u_i \\ &\text{if } j = 0 \text{ then } \nexists u_{i\mathcal{Q}}^c\end{aligned} \qquad (5.22b)$$

- for the quality scaled image, where $v_i^{\mathcal{Q}} \in V^{\mathcal{Q}}$, is the quality scaled coefficient corresponding to $v_i' \in V'$,

$$\begin{aligned} &\text{if } m_i < j_i \text{ then } u_{i\mathcal{Q}}^c = \lfloor \frac{u_i}{2^{m_i}} \rfloor 2^{m_i} \\ &\text{if } m_i \geq j_i \text{ then } \nexists u_i^c \end{aligned} \tag{5.22c}$$

Let

$$u_{i^*}^c = Candidate(v_{i^*}^*, i^*, \Lambda^*) = \begin{cases} \left\lfloor \frac{u_{i^*}^*}{2^{m_{i^*}^*}} \right\rfloor 2^{m_{i^*}^*} & \text{if } m_{i^*}^* < j_{i^*}^* \\ \nexists & \text{if } m_{i^*}^* \geq j_{i^*}^*, \end{cases} \tag{5.21}$$

Assume that for any scaled image $I^{\mathcal{F}} = \mathcal{F}(\mathsf{Embed}(I, \Lambda^{\mathcal{F}}))$, that has undergone only JPEG2000 resolution and/or quality scaling, the correct number of missing bits can be calculated $m^{\mathcal{F}} = m$ and the correct number of watermark bits can be calculated $j^{\mathcal{F}} = j$.

1. Unscaled

If $I' = \mathsf{Embed}(I, \Lambda)$ is the unscaled watermarked image, where $v_i' \in V'$ is the coefficient corresponding to $v_i \in V$ then, given the correct watermarking parameters $\Lambda' = \Lambda$,

If $j_i > 0$,

$$\begin{aligned} u_{i'}^c &= Candidate(v_{i'}', i', \Lambda') \\ &= \left\lfloor \frac{u_{i'}'}{2^{m_i'}} \right\rfloor 2^{m_{i'}'} && m_{i'}' = m_i = 0 < j_{i'}' = j_i \\ &= \lfloor u_{i'}' \rfloor && m_{i'}' = m_i = 0 \\ &= \mathsf{G}(v_{i'}', i', \Lambda', I') && (5.18) \\ &= \mathsf{G}(v_i', i, \Lambda, I') && (5.13a), \Lambda' = \Lambda \\ &= \mathsf{G}(v_i, i, \Lambda, I) && (5.19a) \\ &= u_i && (5.18). \end{aligned}$$

If $j_i = 0$, then

$$\begin{aligned} u_{i'}^c &= Candidate(v_{i'}', i', \Lambda) \\ &= \nexists && m_{i'}' = m_i = 0 = j_{i'}' = j_i \end{aligned}$$

and property (5.22a) is proven.

2. Resolution Scaled

If $I^{\mathcal{R}} = \mathcal{R}(\mathsf{Embed}(I, \Lambda))$ is the resolution scaled image and $v_{i^{\mathcal{R}}}^{\mathcal{R}} \in V^{\mathcal{R}}$ is the coefficient corresponding to $v_i' \in V'$ and $v_i \in V$ then, since $v_{i^{\mathcal{R}}}^{\mathcal{R}} \in V^{\mathcal{R}}, v_{i^{\mathcal{R}}}^{\mathcal{R}} \in I^{\mathcal{R}}$, so the coefficient was unaffected by resolution scaling, thus, given the correct watermarking parameters $\Lambda^{\mathcal{R}} = \Lambda$,

If $j_i > 0$,

$$\begin{aligned}
u^c_{i^{\mathcal{R}}} &= Candidate(v^{\mathcal{R}}_{i^{\mathcal{R}}}, i^{\mathcal{R}}, \Lambda^{\mathcal{R}}) \\
&= \left\lfloor \frac{u^{\mathcal{R}}_{i^{\mathcal{R}}}}{2^{m^{\mathcal{R}}_{i^{\mathcal{R}}}}} \right\rfloor 2^{m^{\mathcal{R}}_{i^{\mathcal{R}}}} && m^{\mathcal{R}}_{i^{\mathcal{R}}} = m_i = 0 < j^{\mathcal{R}}_{i^{\mathcal{R}}} = j_i \\
&= \left\lfloor u^{\mathcal{R}}_{i^{\mathcal{R}}} \right\rfloor && m^{\mathcal{R}}_{i^{\mathcal{R}}} = m_i = 0 \\
&= \mathsf{G}(v^{\mathcal{R}}_{i^{\mathcal{R}}}, i^{\mathcal{R}}, \Lambda^{\mathcal{R}}, I^{\mathcal{R}}) && (5.18) \\
&= \mathsf{G}(v^{\mathcal{R}}_i, i, \Lambda, I^{\mathcal{R}}) && (5.13\text{b}), \Lambda^{\mathcal{R}} = \Lambda \\
&= \mathsf{G}(v_i, i, \Lambda, I) && (5.19\text{b}) \\
&= u_i && (5.18).
\end{aligned}$$

If $j_i = 0$, then

$$\begin{aligned}
u^c_{i^{\mathcal{R}}} &= Candidate(v^{\mathcal{R}}_{i^{\mathcal{R}}}, i^{\mathcal{R}}, \Lambda) \\
&= \nexists && m^{\mathcal{R}}_{i^{\mathcal{R}}} = m_i = 0 = j^{\mathcal{R}}_{i^{\mathcal{R}}} = j_i
\end{aligned}$$

and property (5.22b) is proven.

3. Quality Scaled

If $I^{\mathcal{Q}} = \mathcal{Q}(\mathsf{Embed}(I, \Lambda))$ is the quality scaled image and $v^{\mathcal{Q}}_{i^{\mathcal{Q}}} \in V^{\mathcal{Q}}$ is the coefficient corresponding to $v'_{i'} \in V'$ and $v_i \in V$ then, given the correct watermarking parameters $\Lambda^{\mathcal{Q}} = \Lambda$,

If $m_i < j_i$,

$$\begin{aligned}
u^c_{i^{\mathcal{Q}}} &= Candidate(v^{\mathcal{Q}}_{i^{\mathcal{Q}}}, i^{\mathcal{Q}}, \Lambda^{\mathcal{Q}}) \\
&= \left\lfloor \frac{u^{\mathcal{Q}}_{i^{\mathcal{Q}}}}{2^{m^{\mathcal{Q}}_{i^{\mathcal{Q}}}}} \right\rfloor 2^{m^{\mathcal{Q}}_{i^{\mathcal{Q}}}} && m^{\mathcal{Q}}_{i^{\mathcal{Q}}} = m_i < j^{\mathcal{Q}}_{i^{\mathcal{Q}}} = j_i \\
&= \left\lfloor \frac{u^{\mathcal{Q}}_{i^{\mathcal{Q}}}}{2^{m_i}} \right\rfloor 2^{m_i} && m^{\mathcal{Q}}_{i^{\mathcal{Q}}} = m_i \\
&= \left\lfloor \frac{\mathsf{G}(v^{\mathcal{Q}}_{i^{\mathcal{Q}}}, i^{\mathcal{Q}}, \Lambda^{\mathcal{Q}}, I^{\mathcal{Q}})}{2^{m_i}} \right\rfloor 2^{m_i} && (5.18) \\
&= \left\lfloor \frac{\mathsf{G}(v^{\mathcal{Q}}_i, i, \Lambda, I^{\mathcal{Q}})}{2^{m_i}} \right\rfloor 2^{m_i} && (5.13\text{c}), \Lambda^{\mathcal{Q}} = \Lambda \\
&= \left\lfloor \frac{\mathsf{G}(v_i, i, \Lambda, I)}{2^{m_i}} \right\rfloor 2^{m_i} && (5.19\text{c}) \\
&= \left\lfloor \frac{u_i}{2^{m_i}} \right\rfloor 2^{m_i} && (5.18).
\end{aligned}$$

If $m_i \geq j_i$, then

$$\begin{aligned} u^c_{i^{\mathcal{Q}}} &= Candidate(v^{\mathcal{Q}}_{i^{\mathcal{Q}}}, i^{\mathcal{Q}}, \Lambda) \\ &= \nexists \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad m^{\mathcal{Q}}_{i^{\mathcal{Q}}} = m_i \geq j^{\mathcal{Q}}_{i^{\mathcal{Q}}} = j_i \end{aligned}$$

and property (5.22c) is proven.

So the truncated candidate

$$u^c_{i^*} = Candidate(v^*_{i^*}, i^*, \Lambda^*, I^*) = \begin{cases} \left\lfloor \frac{u^*_{i^*}}{2^{m^*_{i^*}}} \right\rfloor 2^{m^*_{i^*}} & \text{if } m^*_{i^*} < j^*_{i^*} \\ \nexists & \text{if } m^*_{i^*} \geq j^*_{i^*}, \end{cases}$$

satisfies our requirements, *provided that* for any scaled image $I^{\mathcal{F}} = \mathcal{F}(\mathsf{Embed}(I, \Lambda^{\mathcal{F}}))$, that has undergone only JPEG2000 resolution and/or quality scaling, the correct number of missing bits are calculated $m^{\mathcal{F}} = m$ and the correct number of watermark bits are calculated $j^{\mathcal{F}} = j$. Given the correct watermarking parameters and provided the image has not been damaged during transmission, these calculations will be correct (eqns (5.25) and (5.27)).

D.3.6 Proof of Exact Matching

We show that the algorithm described in section 5.1.8 satisfies the requirements of that section. Specifically that, given the correct watermarking parameters, the watermark extracted from any JPEG2000 scaled watermarked image will exactly match the truncated candidate watermark, resulting in the detection output 'True'.

We can consider resolution and quality scaling to be separable.[1] Thus, although we have demonstrated the properties of our watermarking scheme separately for a resolution scaled image $I^{\mathcal{R}}$ and a quality scaled image $I^{\mathcal{Q}}$, they will hold equally well for the resolution and quality scaled image $I^{\mathcal{F}} = \mathcal{R}(\mathcal{Q}(I'))$.

Let $I^{\mathcal{F}} = \mathcal{R}(I^{\mathcal{Q}})$ be a scaled watermarked image, where $I^{\mathcal{Q}} = \mathcal{Q}(I')$ and $I' = \mathsf{Embed}(I, \Lambda)$ and let $\Lambda^{\mathcal{F}} = \Lambda = \{\alpha, t, sk\}$. Let $v^{\mathcal{F}}_{i^{\mathcal{F}}}$, $v^{\mathcal{Q}}_{i^{\mathcal{Q}}}$ and $v'_{i'}$ represent the corresponding coefficients to $v_i \in I$.

[1] From the encoded image I', we can produce a resolution and quality scaled image $F(I') = \mathcal{RQ}(I')$ by discarding all packets in resolution layers greater than r or in quality layers greater than q in a single scaling operation. The same image is also obtained if we first produce a quality scaled image, by discarding all packets in quality layers greater than q, and then apply resolution scaling, discarding all packets in resolution layers greater than r, in a subsequent operation or vice versa $F(I') = \mathcal{Q}(\mathcal{R}(I'))$.

Similarly an arbitrary sequence of resolution and quality scalings, e.g. $\mathcal{R}_1(\mathcal{Q}_1(\mathcal{Q}_2(\mathcal{R}_2(\mathcal{Q}_3(I')))))$ is equivalent to a single scaling operation F which discards all such layers, e.g. resolution layers greater than $r = \min(r1, r2)$ and quality layers greater than $q = \min(q_1, q_2, q_3)$. However, we are unlikely to encounter any applications for which more than one scaling operation would be desirable.

Coefficient Selection

Because $v_{i^{\mathcal{F}}}^{\mathcal{F}}$ is the resolution scaled coefficient corresponding to $v_{i^{\mathcal{Q}}}^{\mathcal{Q}}$, either it has been lost by resolution scaling $v_{i^{\mathcal{F}}}^{\mathcal{F}} \notin I^{\mathcal{F}}$ or it has the same value as $v_{i^{\mathcal{Q}}}^{\mathcal{Q}}$:

$$v_{i^{\mathcal{F}}}^{\mathcal{F}} = v_{i^{\mathcal{Q}}}^{\mathcal{Q}} \iff v^{\mathcal{F}} \in I^{\mathcal{F}} \tag{5.6b}$$

thus

$$v_{i^{\mathcal{F}}}^{\mathcal{F}} \in I^{\mathcal{F}} \implies (v_{i^{\mathcal{F}}}^{\mathcal{F}} \neq 0 \iff v_{i^{\mathcal{Q}}}^{\mathcal{Q}} \neq 0)$$

So, for any selected coefficient $v_i \in V$ that has not been completely lost due to scaling $v_{i^{\mathcal{F}}}^{\mathcal{F}} \in I^{\mathcal{F}} \wedge v_{i^{\mathcal{F}}}^{\mathcal{F}} \neq 0$,

$$\begin{aligned}
v_i \in V \wedge v_{i^{\mathcal{F}}}^{\mathcal{F}} \in I^{\mathcal{F}} \wedge v_{i^{\mathcal{F}}}^{\mathcal{F}} \neq 0 &\iff v_i \in V \wedge v_{i^{\mathcal{F}}}^{\mathcal{F}} \in I^{\mathcal{F}} \wedge v_{i^{\mathcal{Q}}}^{\mathcal{Q}} \neq 0 \\
&\iff v'_{i'} \in V' \wedge v_{i^{\mathcal{F}}}^{\mathcal{F}} \in I^{\mathcal{F}} \wedge v_{i^{\mathcal{Q}}}^{\mathcal{Q}} \neq 0 && \text{(5.4a)} \\
&\iff v_{i^{\mathcal{Q}}}^{\mathcal{Q}} \in V^{\mathcal{Q}} \wedge v_{i^{\mathcal{F}}}^{\mathcal{F}} \in I^{\mathcal{F}} && \text{(5.4c)} \\
&\iff v_{i^{\mathcal{F}}}^{\mathcal{F}} \in V^{\mathcal{F}} && \text{(5.4b)}
\end{aligned}$$

Indexing

The image $I^{\mathcal{F}}$ is a resolution scaled version of $I^{\mathcal{Q}}$ so they have the same number of components

$$C^{\mathcal{F}} = C^{\mathcal{Q}},$$

but the K highest image resolutions have been lost so $R^{\mathcal{F}} = R^{\mathcal{Q}} - K$, and the dimensions of each component are reduced to $X^{\mathcal{F}} = \left\lceil \frac{X^{\mathcal{Q}}}{2^K} \right\rceil$ and $Y^{\mathcal{F}} = \left\lceil \frac{Y^{\mathcal{Q}}}{2^K} \right\rceil$.

The loss of K resolutions from the image will not affect the dimensions (2.14) of any of the remaining resolutions $r < R^{\mathcal{Q}} - K$

$$\begin{aligned}
X^{\mathcal{F}}[r] &= \left\lceil \frac{X^{\mathcal{F}}}{2^{R^{\mathcal{F}}-r-1}} \right\rceil \\
&= \left\lceil \frac{\left\lceil \frac{X^{\mathcal{Q}}}{2^K} \right\rceil}{2^{R^{\mathcal{Q}}-K-r-1}} \right\rceil \\
&= \left\lceil \frac{\left\lceil \frac{X^{\mathcal{Q}}}{2^K} \right\rceil 2^K}{2^{R^{\mathcal{Q}}-r-1}} \right\rceil \\
&= \left\lceil \frac{X^{\mathcal{Q}}}{2^{R^{\mathcal{Q}}-r-1}} \right\rceil \qquad 2^{R^{\mathcal{Q}}} = 2^{R^{\mathcal{Q}}-K-r-1} 2^K, r < R^{\mathcal{Q}} - K \implies 2^{R^{\mathcal{Q}}-K-r-1} \in \mathbb{N} \\
&= X^{\mathcal{Q}}[r]
\end{aligned}$$

and, similarly,

$$Y^{\mathcal{F}}[r] = Y^{\mathcal{Q}}[r]$$

Thus, from equations (5.10) and (5.11),

$$X^{\mathcal{F}}[r,o] = X^{\mathcal{Q}}[r,o]$$
$$Y^{\mathcal{F}}[r,o] = Y^{\mathcal{Q}}[r,o],$$

for $r < R^{\mathcal{Q}} - K$ and $o \in \{0,1,2\}$. nor does it affect the positions of the remaining coefficients so $v^{\mathcal{F}}_{i^{\mathcal{F}}} \in I^{\mathcal{F}}$ has position

$(c^{\mathcal{F}},r^{\mathcal{F}},o^{\mathcal{F}},x^{\mathcal{F}},y^{\mathcal{F}}) = (c^{\mathcal{Q}},r^{\mathcal{Q}},o^{\mathcal{Q}},x^{\mathcal{Q}},y^{\mathcal{Q}})$,

giving it the index

$$\begin{aligned} i^{\mathcal{F}} &= c^{\mathcal{F}}X^{\mathcal{F}}[r^{\mathcal{F}}]Y^{\mathcal{F}}[r^{\mathcal{F}}] + (C^{\mathcal{F}} - c^{\mathcal{F}})X^{\mathcal{F}}[r^{\mathcal{F}}-1]Y^{\mathcal{F}}[r^{\mathcal{F}}-1] \\ &\qquad + \sum_{o=0}^{o=o^{\mathcal{F}}-1} X^{\mathcal{F}}[r^{\mathcal{F}},o]Y^{\mathcal{F}}[r^{\mathcal{F}},o] + y^{\mathcal{F}}X^{\mathcal{F}}[r^{\mathcal{F}},o^{\mathcal{F}}] + x^{\mathcal{F}} \qquad (5.12) \\ &= c^{\mathcal{Q}}X^{\mathcal{Q}}[r^{\mathcal{Q}}]Y^{\mathcal{Q}}[r^{\mathcal{Q}}] + (C^{\mathcal{Q}} - c^{\mathcal{Q}})X^{\mathcal{Q}}[r^{\mathcal{Q}}-1]Y^{\mathcal{Q}}[r^{\mathcal{Q}}-1] \\ &\qquad + \sum_{o=0}^{o=o^{\mathcal{Q}}-1} X^{\mathcal{Q}}[r^{\mathcal{Q}},o]Y^{\mathcal{Q}}[r^{\mathcal{Q}},o] + y^{\mathcal{Q}}X^{\mathcal{Q}}[r^{\mathcal{Q}},o^{\mathcal{Q}}] + x^{\mathcal{Q}} \\ &= i^{\mathcal{Q}} \qquad (5.12) \\ &= i' \qquad (5.13c) \\ &= i \qquad (5.13a) \end{aligned}$$

Element Generation

In the scaled image $I^{\mathcal{F}} = \mathcal{R}(I^{\mathcal{Q}})$, any selected coefficient $v^{\mathcal{F}}_{i^{\mathcal{F}}} \in V^{\mathcal{F}}$, was unaffected by resolution scaling (5.6b) so

$$\begin{aligned} v^{\mathcal{F}}_{i^{\mathcal{F}}} &= v^{\mathcal{Q}}_{i\rangle^{\mathcal{F}}} \\ &= v^{\mathcal{Q}}_{i} \qquad\qquad i^{\mathcal{F}} = i \\ \bar{v}^{\mathcal{F}}_{i} &= \bar{v}^{\mathcal{Q}}_{i}. \end{aligned}$$

For the selected coefficient, $v^{\mathcal{F}}_{i} \in V^{\mathcal{F}}$, the corresponding coefficient $v^{\mathcal{Q}}_{i} \in V^{\mathcal{Q}}$ in the quality scaled image $I^{\mathcal{Q}}$ is, in turn, the corresponding coefficient to $v'_i \in V'$ and $v_i \in V$, so this value is

$$\begin{aligned} &= \bar{v}'_i \qquad\qquad \text{from the proof D.3.4 } \textbf{4. Quality Scaled} \\ &= v_i \qquad\qquad \text{from the proof D.3.4 } \textbf{2. Unscaled} \end{aligned}$$

The sequence element

$$\begin{aligned} w_i^{\mathcal{F}} &= \mathsf{g}(sk^{\mathcal{F}}, i) && (5.15)\\ &= \mathsf{g}(sk, i) && \Lambda^{\mathcal{F}} = \Lambda \implies sk^{\mathcal{F}} = sk \\ &= w_i. \end{aligned}$$

Thus,

$$\begin{aligned} \mathsf{G}(v_i^{\mathcal{F}}, i, \Lambda, I^{\mathcal{F}}) &= \lfloor \alpha 2^{-(h+1)} \bar{v}_i^{\mathcal{F}} w_i^{\mathcal{F}} \rfloor \\ &= \lfloor \alpha 2^{-(h+1)} \bar{v}_i w_i \rfloor \\ &= \mathsf{G}(v_i, i, \Lambda, I). \end{aligned}$$

Missing Bits Calculation

$I^{\mathcal{F}}$ has undergone only JPEG2000 resolution and quality scaling, so the calculations of section 5.1.6 will produce the correct number of bits that are missing due to scaling

$$m_{i^{\mathcal{F}}}^{\mathcal{F}} = m_i \qquad (5.25)$$

Watermark Bits Calculation

$I^{\mathcal{F}}$ has undergone only JPEG2000 resolution and quality scaling and the correct parameters have been provided $\Lambda^{\mathcal{F}} = \Lambda$, so the calculations of section 5.1.6 will produce the correct number of bits that are missing due to scaling

$$j_{i^{\mathcal{F}}}^{\mathcal{F}} = j_i \qquad (5.27b)\ (5.27c)$$

Watermark Extraction

The extracted watermark is the sequence

$$U^d = \{u_i^d : v_i^{\mathcal{F}} \in V^{\mathcal{F}}\}$$

Any selected coefficient $v_i^{\mathcal{F}} \in V^{\mathcal{F}}$ exists in the scaled image $I^{\mathcal{F}} = \mathcal{F}(I^{\mathcal{Q}})$ and so cannot have been affected by resolution scaling

$$v_i^{\mathcal{F}} = v_i^{\mathcal{Q}} \qquad (5.6b)$$

hence

$$u^d_{i^{\mathcal{F}}} = \begin{cases} |v^{\mathcal{F}}_{i^{\mathcal{F}}}| - Q_{2^{j^{\mathcal{F}}_{i^{\mathcal{F}}}}}(v^{\mathcal{F}}_{i^{\mathcal{F}}}) - \left\lfloor 2^{m^{\mathcal{F}}_{i^{\mathcal{F}}}} \mathrm{r} \right\rfloor & \text{if } m^{\mathcal{F}}_{i^{\mathcal{F}}} < j^{\mathcal{F}}_{i^{\mathcal{F}}} \\ \nexists & \text{if } m^{\mathcal{F}}_{i^{\mathcal{F}}} \geq j^{\mathcal{F}}_{i^{\mathcal{F}}}, \end{cases} \tag{5.8}$$

$$= \begin{cases} |v^{\mathcal{Q}}_{i^{\mathcal{Q}}}| - Q_{2^{j^{\mathcal{F}}_{i^{\mathcal{F}}}}}(v^{\mathcal{Q}}_{i^{\mathcal{Q}}}) - \left\lfloor 2^{m^{\mathcal{F}}_{i^{\mathcal{F}}}} \mathrm{r} \right\rfloor & \text{if } m^{\mathcal{F}}_{i^{\mathcal{F}}} < j^{\mathcal{F}}_{i^{\mathcal{F}}} \\ \nexists & \text{if } m^{\mathcal{F}}_{i^{\mathcal{F}}} \geq j^{\mathcal{F}}_{i^{\mathcal{F}}}, \end{cases} \quad i^{\mathcal{F}} = i = i^{\mathcal{Q}}, v^{\mathcal{F}}_i = v^{\mathcal{Q}}_i$$

$$= \begin{cases} |v^{\mathcal{Q}}_{i^{\mathcal{Q}}}| - Q_{2^{j_i}}(v^{\mathcal{Q}}_{i^{\mathcal{Q}}}) - \lfloor 2^{m_i} \mathrm{r} \rfloor & \text{if } m_i < j_i \\ \nexists & \text{if } m_i \geq j_i, \end{cases} \quad m^{\mathcal{F}}_{i^{\mathcal{F}}} = m_i, j^{\mathcal{F}}_{i^{\mathcal{F}}} = j_i$$

which, from the proof **3. Quality Scaled** in section D.3.2,

$$= \begin{cases} \left\lfloor \frac{u_i}{2^{m_i}} \right\rfloor 2^{m_i} & m_i < j_i \\ \nexists & m_i \geq j_i \end{cases}$$

so

$$U^d = \left\{ \left\lfloor \frac{u_i}{2^{m_i}} \right\rfloor 2^{m_i} : v^{\mathcal{F}}_i \in V^{\mathcal{F}} \wedge m_i < j_i \right\}.$$

Candidate Truncation

The truncated candidate watermark is the sequence

$$U^c = \{u^c_{i^{\mathcal{F}}} : v^{\mathcal{F}}_i \in V^{\mathcal{F}}\}$$

where

$$u^c_{i^{\mathcal{F}}} = \begin{cases} \left\lfloor \frac{u^{\mathcal{F}}_{i^{\mathcal{F}}}}{2^{m^{\mathcal{F}}_{i^{\mathcal{F}}}}} \right\rfloor 2^{m^{\mathcal{F}}_{i^{\mathcal{F}}}} & m^{\mathcal{F}}_{i^{\mathcal{F}}} < j^{\mathcal{F}}_{i^{\mathcal{F}}} \\ \nexists & m^{\mathcal{F}}_{i^{\mathcal{F}}} \geq j^{\mathcal{F}}_{i^{\mathcal{F}}} \end{cases}$$

If $m^{\mathcal{F}}_{i^{\mathcal{F}}} < j^{\mathcal{F}}_{i^{\mathcal{F}}}$ then, since the number of missing bits is always non-negative $m^{\mathcal{F}}_{i^{\mathcal{F}}} \geq 0$, we must have $j^{\mathcal{F}}_{i^{\mathcal{F}}} > 0$ and thus $j_i > 0$. Thus the watermark element $u^{\mathcal{F}}_{i^{\mathcal{F}}}$ exists (5.27c) and is equal to u_i (5.27a). Thus

$$u^c_{i^{\mathcal{F}}} = \begin{cases} \left\lfloor \frac{u^{\mathcal{F}}_{i^{\mathcal{F}}}}{2^{m^{\mathcal{F}}_{i^{\mathcal{F}}}}} \right\rfloor 2^{m^{\mathcal{F}}_{i^{\mathcal{F}}}} & m^{\mathcal{F}}_{i^{\mathcal{F}}} < j^{\mathcal{F}}_{i^{\mathcal{F}}} \\ \nexists & m^{\mathcal{F}}_{i^{\mathcal{F}}} \geq j^{\mathcal{F}}_{i^{\mathcal{F}}}. \end{cases}$$

$$= \begin{cases} \left\lfloor \frac{u_i}{2^{m_i}} \right\rfloor 2^{m_i} & m_i < j_i \\ \nexists & m_i \geq j_i \end{cases} \quad m^{\mathcal{F}}_{i^{\mathcal{F}}} = m_i, j^{\mathcal{F}}_{i^{\mathcal{F}}} = j_i, m_i < j_i \implies u^{\mathcal{F}}_{i^{\mathcal{F}}} = u_i$$

so

$$U^c = \left\{ \left\lfloor \frac{u_i}{2^{m_i}} \right\rfloor 2^{m_i} : v^{\mathcal{F}}_i \in V^{\mathcal{F}} \wedge m_i < j_i \right\}.$$

Output

As a result there is an exact match between the candidate and extracted watermarks

$$\begin{aligned} U^c &= \left\{ \left\lfloor \frac{u_i}{2^{m_i}} \right\rfloor 2^{m_i} : v_i^{\mathcal{F}} \in V^{\mathcal{F}} \wedge m_i < j_i \right\} \\ &= U^d \end{aligned}$$

so the output of $\mathsf{Detect}(I^{\mathcal{F}}, \Lambda^{\mathcal{F}})$, where $I^{\mathcal{F}} = \mathcal{R}(\mathcal{Q}(\mathsf{Embed}(I, \Lambda)))$ is a resolution and quality scaled image and the correct watermarking parameters $\Lambda^{\mathcal{F}} = \Lambda$ are given, will be True.

D.4 Additional Details on the Evaluation of the Basic Algorithm

This section contains the more detailed results from the experiments of section 5.2.3.

D.4.1 Correctness and Fragility

D.4.1.1 Exact Match under Resolution/Quality Scaling

Although a zero BER represents an exact match between the candidate and extracted watermarks, this does not always provide sufficient evidence to decide that an image is authentic.

If the watermark embedding strength is low then, for low resolution or quality layers, the number of extracted bits may be too low to provide sufficient confidence that the extracted watermark truly corresponds to the candidate watermark for the given detection key.

The harder it is to achieve a zero error rate using a randomly selected watermark (representing an unwatermarked image or one watermarked with an incorrect key), the more confident we can be that a zero error rate really does correspond to the untampered watermarked image.

The expected bit error rate for an unwatermarked image is 50%, so the number of bit errors X in n extracted bits has a binomial distribution $X\ B(n, 0.5)$. The probability of obtaining 0 errors in an unwatermarked image with 30 extracted bits is

$$P(X <= 0), X \sim B(30, 0.5) = 9.31 \times 10^{-10}$$

less than one in a million. Whereas if only 2 bits could be extracted, even an unwatermarked image would have a 25% chance of producing zero errors. Thus we may, for

example, require at least 30 extracted bits (with 0 total errors) before we accept that the watermark is correct.

Figure D.1 depicts all cases in which the number of bits extracted from a quality scaled subimage falls below 40. Each original image is numbered from 1 to 20, and results for individual detection keys appear in a dithered column above the image number. Only subimages containing one quality layer (rate 0.0025) or two layers (rate 0.005) had a sufficiently low number of extracted bits to appear in this graph, and these are coloured black and red respectively.

For several original images (numbered 5,7,9,12,13,17,19) the number of extracted bits is less than 30 in subimages of one quality layer for some keys (figure D.1). For image 9, the same problem occurs in the second quality layer. Depending on the application, it

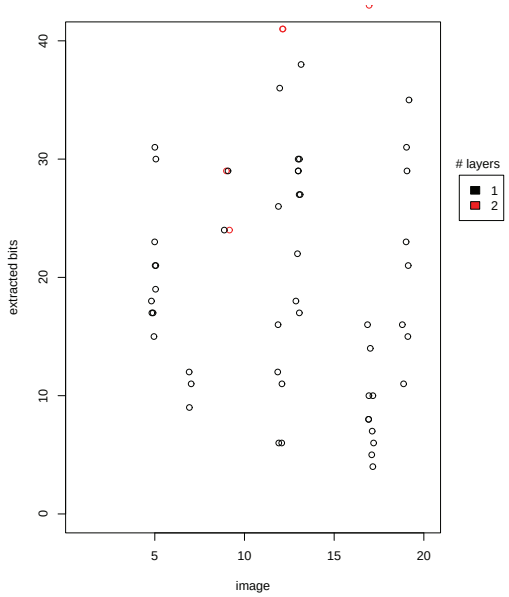


Figure D.1: Low numbers of extracted bits in some low quality subimages.

may be best to err on the side of caution in such cases and reject the subimage.

D.4.1.2 Scalability

Section 5.2.2.2 contains estimates of the fraction of original images for which less than 30 watermark bits are expected to be correctly extracted after resolution scaling to $\frac{1}{1024}$th the original area or quality scaling to a compression rate of 0.0025, 0.005 or 0.01.

These values are derived by assuming that the numbers of correctly extracted watermark bits are log-normally distributed across all original images. Log-normal quantile plots (figures D.2 and D.3) for resolution and quality detectability suggest that this is a reasonable assumption.

The appropriate mean and standard deviation for this distribution are estimated using the mean and standard deviation of the average number of extracted watermark bits, taken across the 10 keys, for all 20 tested original images. The fraction of original images for which the expected number of correctly extracted watermark bits is greater than 30 is then calculated, for the lowest resolution layer and 3 lowest quality layers. The results are shown in table D.1.

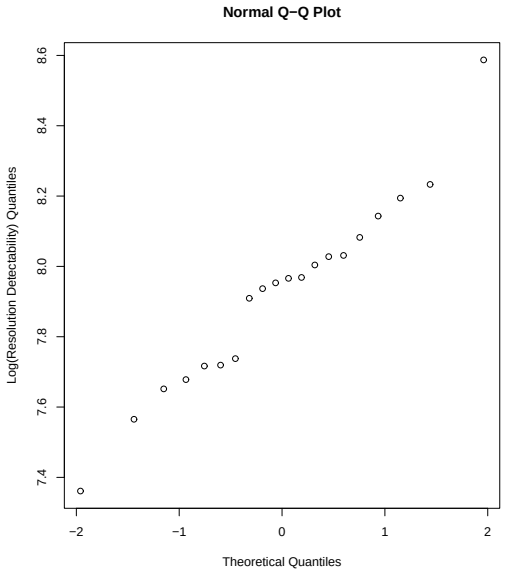


Figure D.2: Log-normal quantile plot for resolution detectability, averaged across keys.

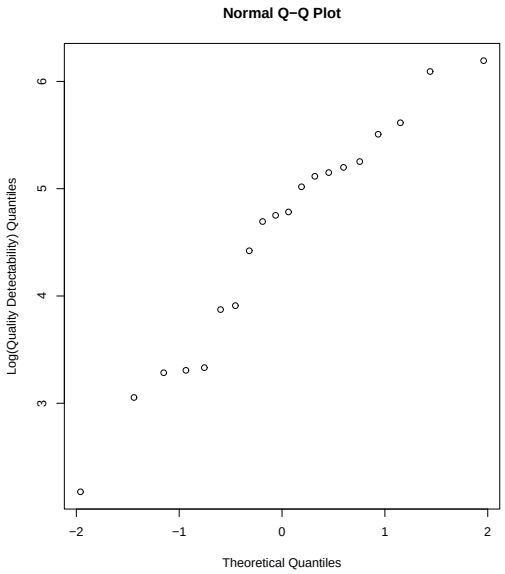


Figure D.3: Log-normal quantile plot for quality detectability, averaged across keys.

Table D.1: Estimated fraction of images for which the key-averaged detectability will be below thirty $P(\bar{\mathcal{D}} < 30)$ where the processing type $\mathcal{F}$ is either resolution scaling $\mathcal{R}$ to roughly 0.1% of the original image area or quality scaling to 0.25, 0.5 or 1% of the original file size.

$\mathcal{F}$	% of original	mean($\log \mathcal{D}$)	sd($\log \mathcal{D}$)	$P(\bar{\mathcal{D}} < 30)$
$\mathcal{R}$	0.1	7.923	0.2729	5.699×10^{-62}
$\mathcal{Q}$	0.25	4.536	1.087	0.1481
$\mathcal{Q}$	0.5	5.385	0.8945	0.01326
$\mathcal{Q}$	1	6.215	0.7226	4.949×10^{-5}

D.4.1.3 Recompression Errors

For individual subimages, the errors caused by decompression and recompression are consistent with the patterns observed in the average results (section 5.2.1.4).

There is no clear trend across subimages with decreasing numbers of resolution layers generated from the same original image (figure D.4). The bit error rates for individual resolution scaled subimages vary between 32 and 46 percent.

For quality scaled subimages, the variation in bit error rates is far more dramatic: between 0 and 46 percent. There is still substantial variation in BER for different original images, but we can clearly see (figure D.5) that the trend of lower error rates in subimages with fewer quality layers, observed in the summary data (table 5.4), is present amongst the subimages generated from each original. This occurs because recompression causes only minor changes to the watermarked coefficients and thus the majority of the changes are in the less significant bits which are generally included in the higher quality layers.

Different images will be more or less affected by recompression depending on the content of the image. Images where many of the pixels have red, green or blue values at or near the extremes of 0 and 255 are more affected than those with more central values. This is because the watermarking algorithm is not designed to ensure that watermarked coefficients will not result in pixels with red, green or blue values less than 0 or greater than 255. Where this occurs (most commonly in images with RGB values already close to 0 or 255), the decompressed image is forced to have RGB values in the range $[0, 255]$, hence the coefficients of the recompressed image will be those which would result in RGB values in the range $[0, 255]$ and therefore will be different from the coefficient values which would result in RGB values outside that range. This causes the extracted watermark from the recompressed image to be different from that expected for the unprocessed image, resulting in the observed errors.

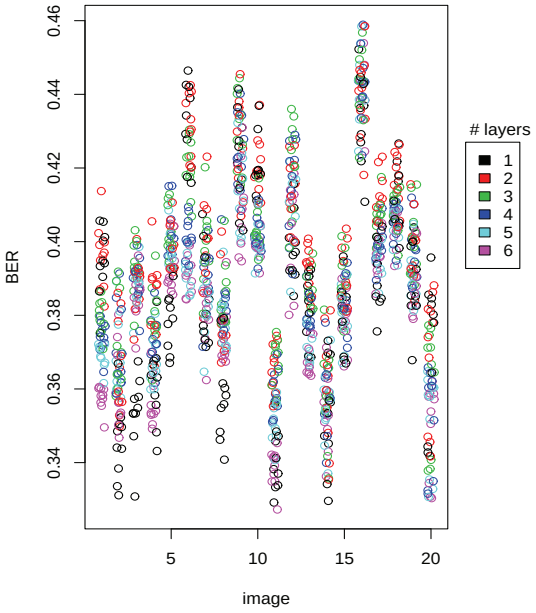


Figure D.4: Bit error rates per image, coloured by the number of layers in the subimage, for a resolution decomposition.

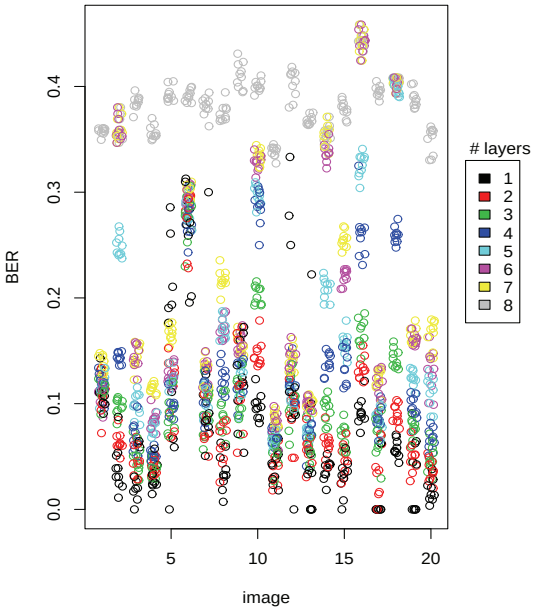


Figure D.5: Bit error rates per image, coloured by the number of layers in the subimage, for a quality decomposition.

D.4.1.4 Gaussian

The error rates for individual Gaussian blurred, watermarked images (figs. D.6 and D.7), follow the same pattern as the average results (section 5.2.1.5), with error rates increasing towards higher layers. Error rates are high, relative to the corresponding recompressed

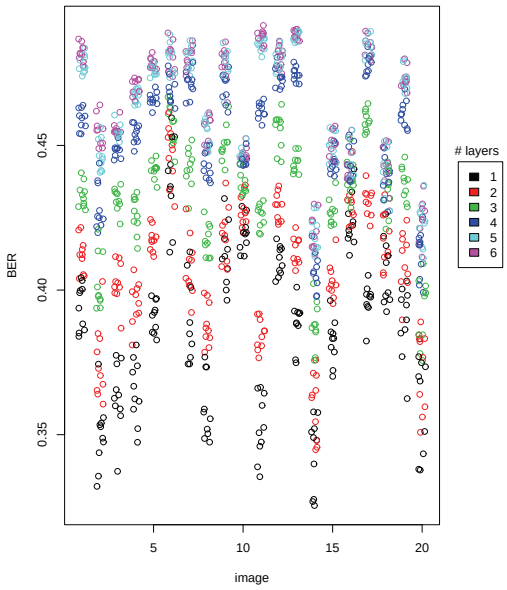


Figure D.6: BER for resolution scaled subimages: Gaussian.

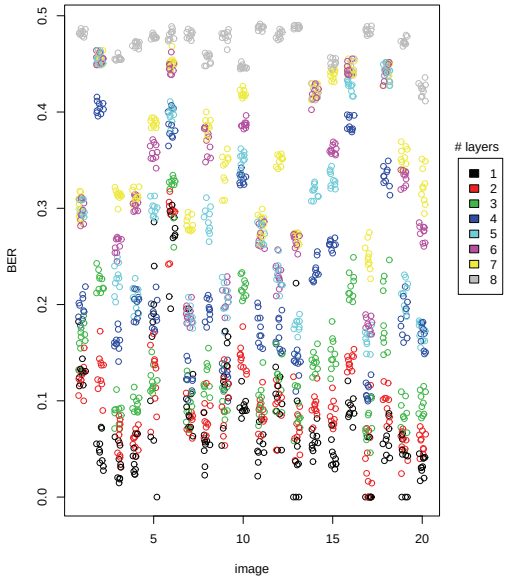


Figure D.7: BER for quality scaled subimages: Gaussian.

image, for images with high contrast, high frequency edges and textures (e.g. 11 and 13); smooth images (16 and 18) are less affected.

D.4.1.5 Hard Thresholding

The error rates for individual hard thresholded, watermarked images (figures D.8 and D.9), follow the same pattern as the average results (section 5.2.1.5), with error rates

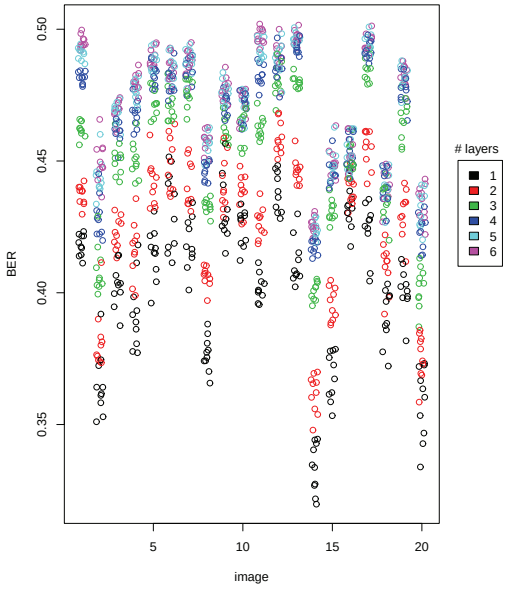


Figure D.8: BER for resolution scaled subimages: hardthresh.

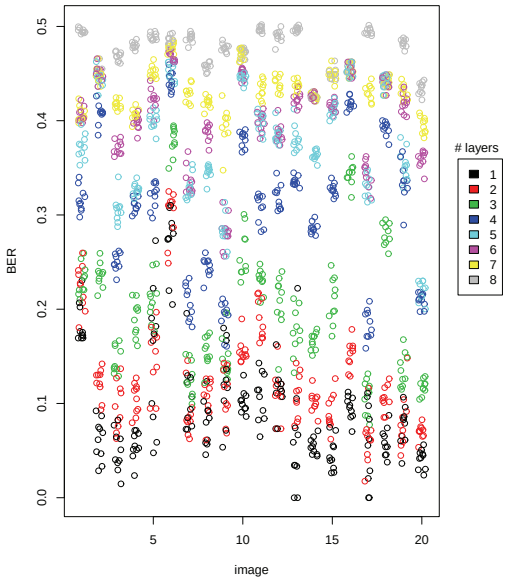


Figure D.9: BER for quality scaled subimages: hardthresh.

increasing towards higher layers for both resolution and quality scaled subimages. As with the Gaussian attack, error rates are high, relative to the corresponding recompressed image, for highly textured images; and low for smooth images.

D.4.1.6 JPEG compression

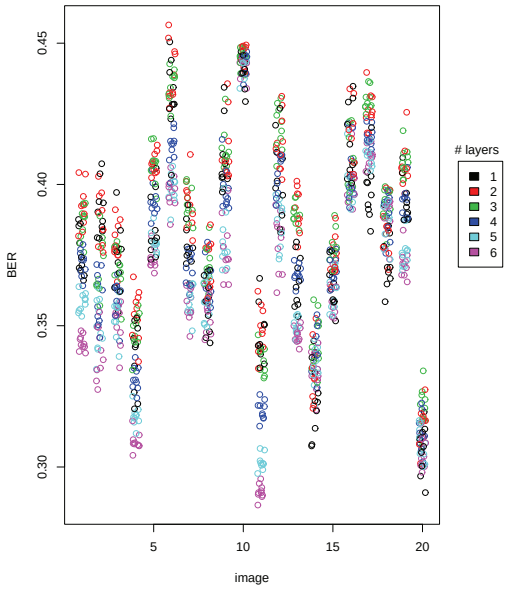


Figure D.10: BER for resolution scaled subimages: JPEG100.

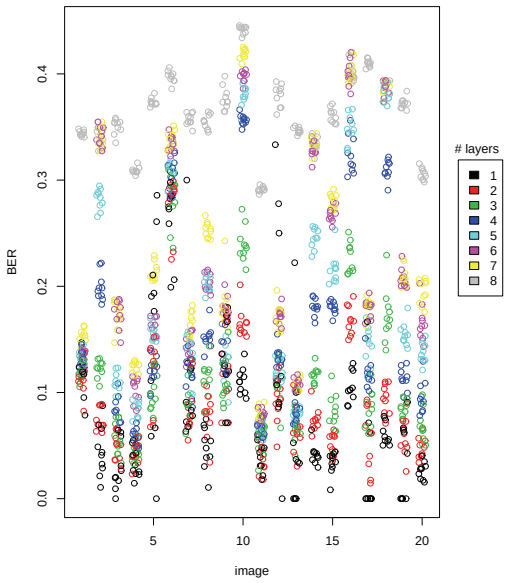


Figure D.11: BER for quality scaled subimages: JPEG100.

High quality JPEG compression shows very similar results to recompression using JPEG2000 with the original embedding parameters. Although the precise error rates vary somewhat, images which produced higher BERs under recompression also produce higher BERs under quality 100 JPEG compression (figures D.10 and D.11).

With stronger compression (JPEG quality 40) the results are more similar to Gaussian blur and hard thresholding.

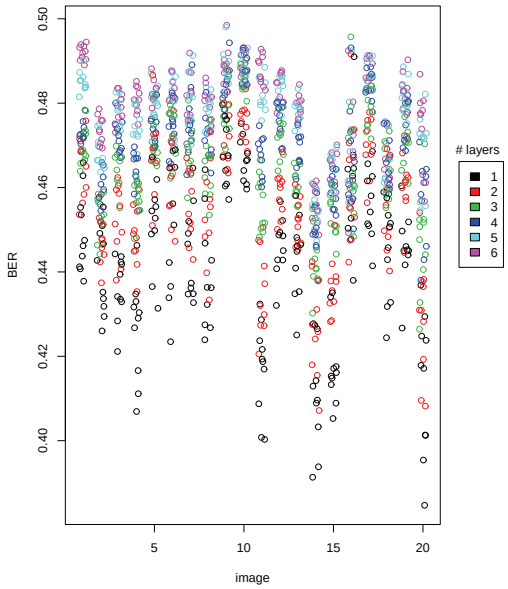


Figure D.12: BER for resolution scaled subimages: JPEG40.

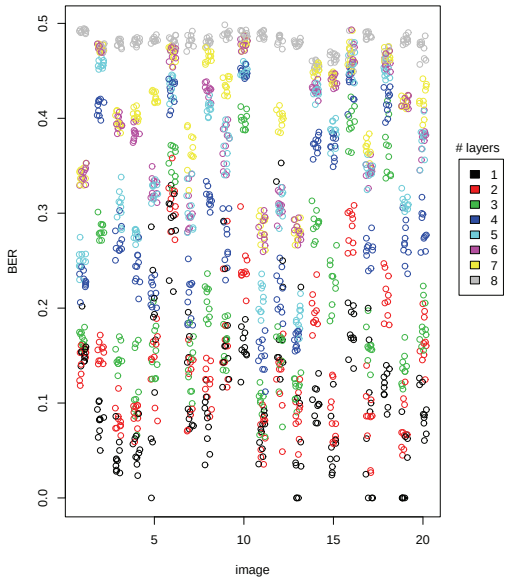


Figure D.13: BER for quality scaled subimages: JPEG40.

D.4.1.7 Median

The individual image results for median filtering (figures D.14 and D.15) are similar to those of Gaussian blur and hard thresholding. Error rates are higher, relative to decompress/recompress, for images with many edges and textured regions (e.g. images 4, 11 and 13) and lower for images with large smooth areas (e.g. images 14 and 16).

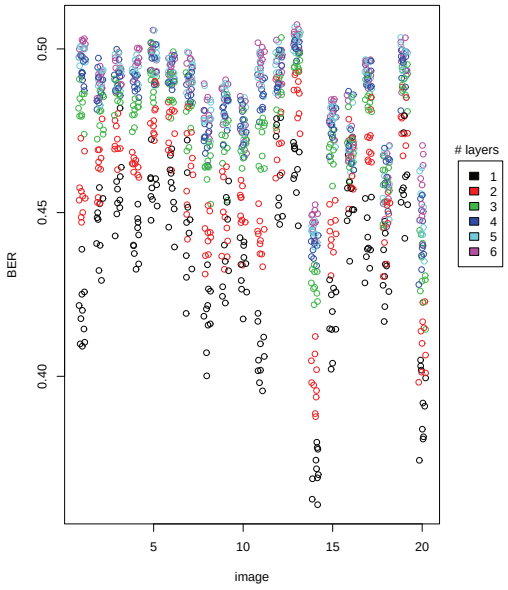


Figure D.14: BER for resolution scaled subimages: Median2x2.

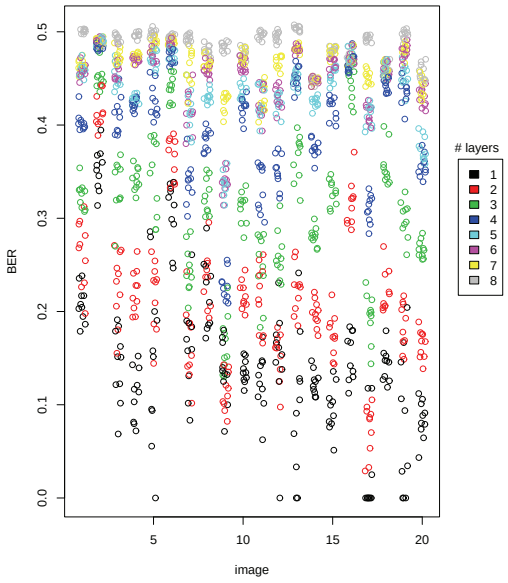


Figure D.15: BER for quality scaled subimages: Median2x2.

D.4.1.8 Midpoint

The results of midpoint filtering (figures D.16 and D.17) are similar to those of median filtering. Error rates are higher, relative to decompress/recompress, for images with many edges and textured regions and lower for images with large smooth areas.

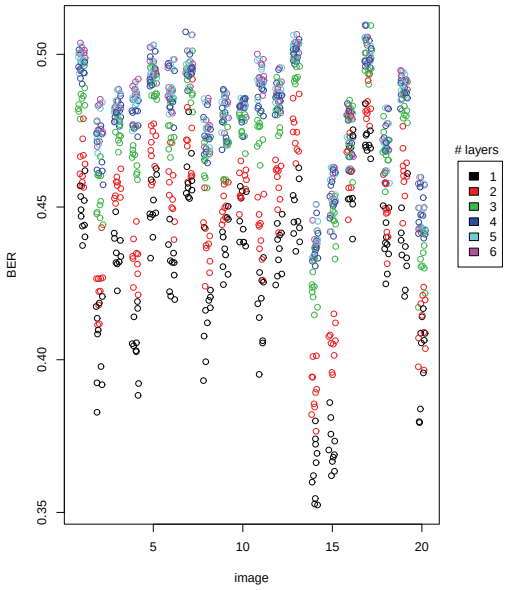


Figure D.16: BER for resolution scaled subimages: Midpoint3x3.

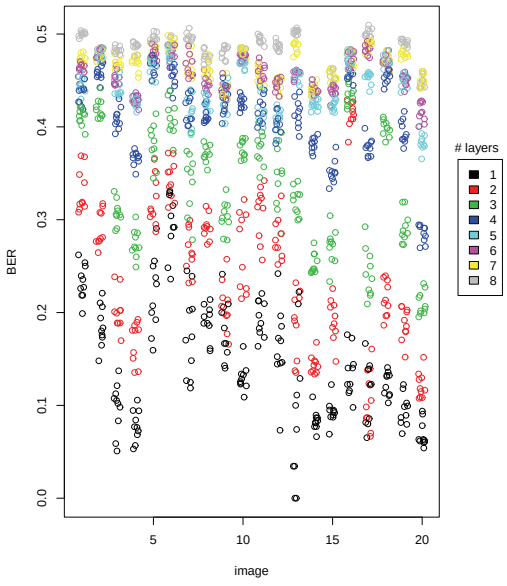


Figure D.17: BER for quality scaled subimages: Midpoint3x3.

D.4.1.9 Trimmed Mean

The trimmed mean results for individual images (figures D.18 and D.19) show that the greatest increase in error from the baseline recompress/decompress occurs in images with high contrast edges and texture (images 1, 11, 13) and those with the least difference have large smooth areas and low contrast texture (images 16, 18).

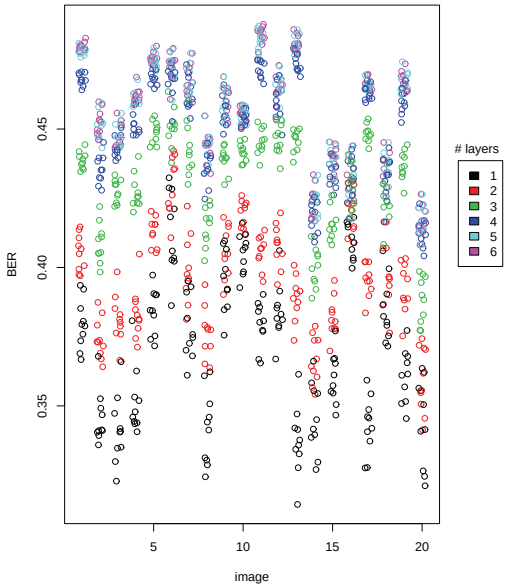


Figure D.18: Total BER for resolution scaled subimages: Trimmed Mean.

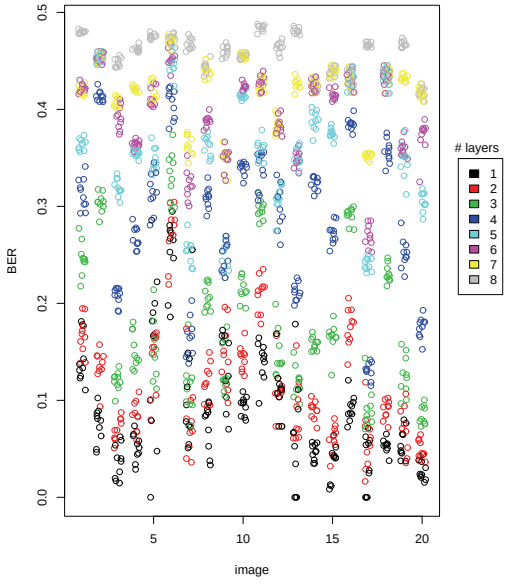


Figure D.19: BER for quality scaled subimages: Trimmed Mean.

D.4.1.10 Down Sampling

The individual image results for downsampling are similar to those of other nongeometric attacks (figures D.20 and D.21).

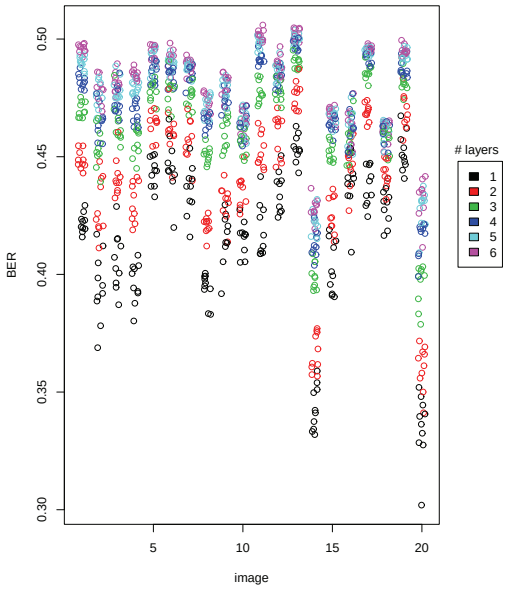


Figure D.20: BER for resolution scaled subimages: sampledownup75.

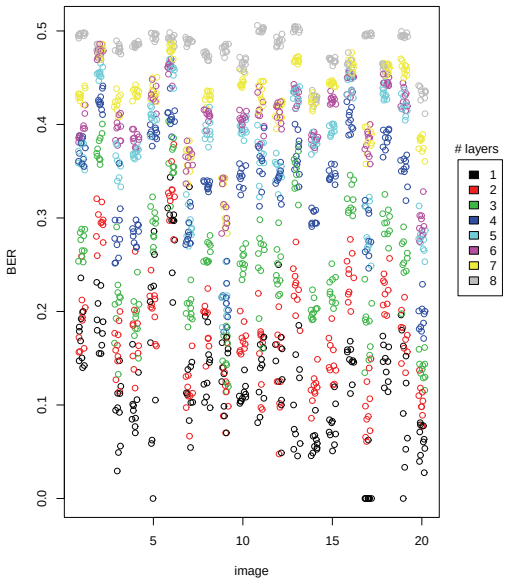


Figure D.21: BER for quality scaled subimages: sampledownup75.

D.4.1.11 Sharpening

The pattern of individual BERs for the sharpening attack is still very similar to those of the other nongeometric attacks. This is clear for resolution scaling, but is harder to see in the quality scaled case due to the dramatic change in the scale of figure D.23 relative to corresponding figures for other attacks (the change in scale is a result of error rates in low quality layers for some images, with high-contrast detail, reaching 100% for some keys).

More errors occur in images with fine, high contrast texture and edges (11, 13, 20) and less in images composed primarily of smooth regions (16, 18)

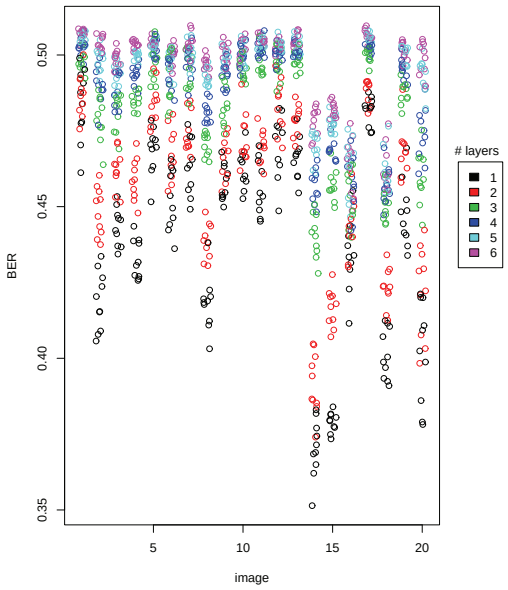


Figure D.22: BER for resolution scaled subimages: sharpening.

D.4.1.12 JPEG2000 Compression

As with most other non-geometric attacks, bit error rates for individual images under JPEG2000 compression at rate 0.0125 are increased for images with many textured regions (e.g. images 4, 11, 13) and decreased for images with mainly smooth areas (e.g. images 2, 14, 16, 18), relative to the recompressed/decompressed baseline.

The twelfth image has one secret key that produces an error rate greater than 50% at the lowest resolution layer (fig. D.24). In this case, we see an unusual *decreasing* error rate as the number of layers increases. This occurs simply because the low resolution error rate is unusually high for that image and key; with additional layers the bit error rate tends towards the 50% error rate expected for a destroyed watermark.

The gap in error rates, between the third and fourth quality layers, that was observed in the average results is also apparent for individual images (fig. D.25).

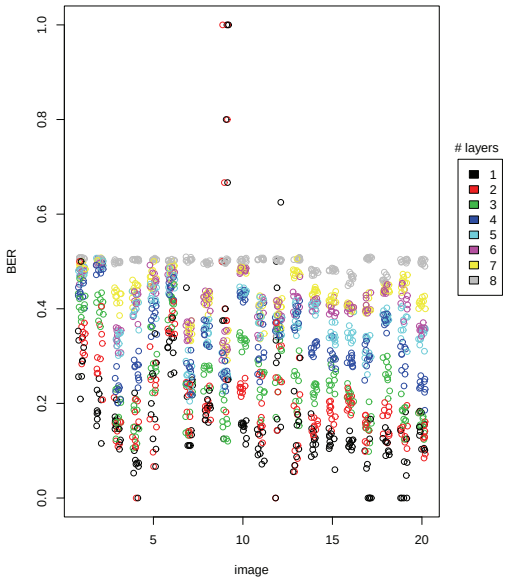


Figure D.23: BER for quality scaled subimages: sharpening.

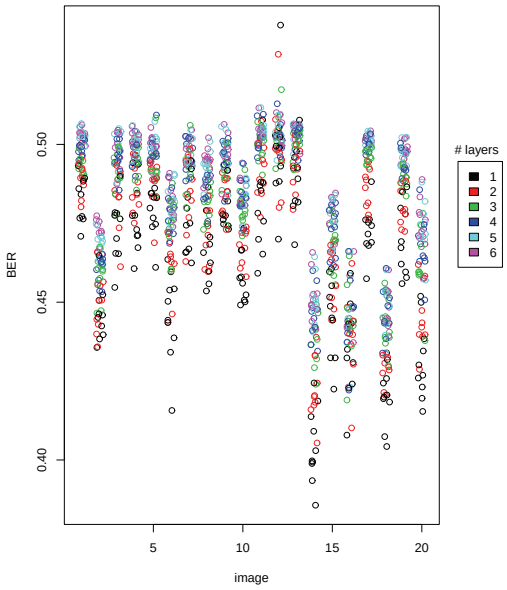


Figure D.24: Bit error rates for resolution scaled subimages: JPEG2000 rate 0.0125.

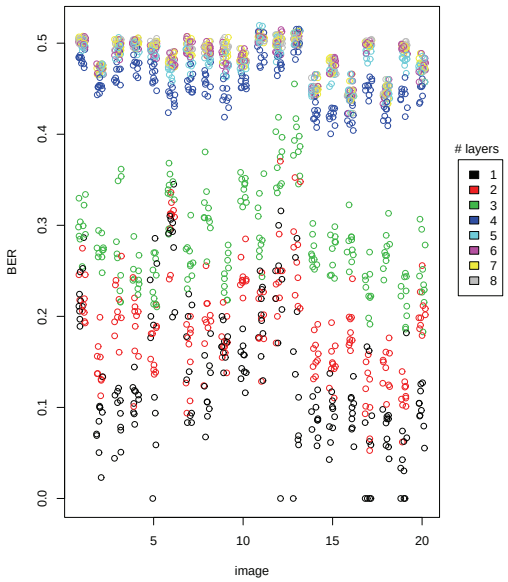


Figure D.25: Bit error rates for quality scaled subimages: JPEG2000 rate 0.0125.

D.4.1.13 Cropping

The bit error rates for individual images after a cropping attack (figures D.26 and D.27) are similar to the results for an average image (section 5.2.1.6); except for the third and fifth images, which have substantially lower error rates at the lowest resolution and quality layers.

In most of the images considered, the 10% reduction in area caused by the cropping attack, is sufficient to reduce the dimensionality of the lowest resolution by at least 1 pixel in the x direction, causing mismatched elements to be produced for all coefficients but those in the first row of the first component of the lowest resolution layer. For the images 3 and 5, however, the dimensions are such that only the y dimension is reduced in the lowest resolution layer; so all candidate watermark elements in the first component of the lowest resolution layer are correctly generated, resulting in the lower error rates observed for these images. As more data from higher resolutions is included, the correctly generated candidate elements comprise a progressively smaller fraction of the whole, so the bit error rate increases towards 50%. A similar trend is observed in the quality layers because the lowest quality layer contains a high proportion of data from the lowest resolution layer.

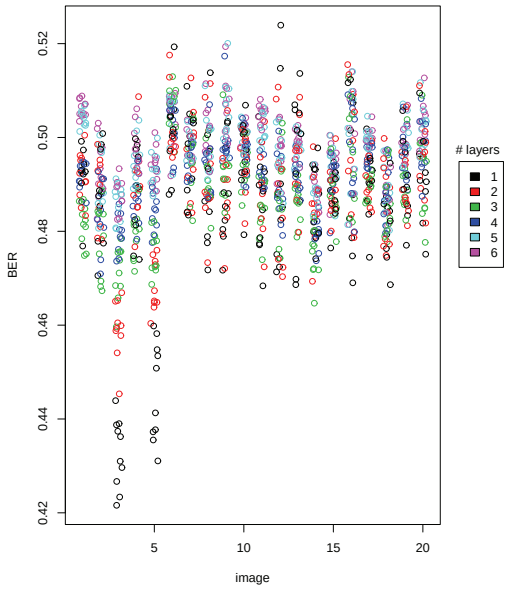


Figure D.26: BER for resolution scaled subimages: Cropping 10%.

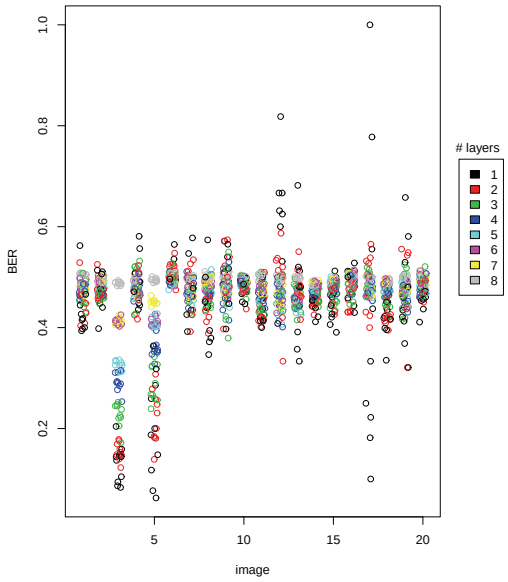


Figure D.27: BER for quality scaled subimages: Cropping 10%.

D.4.1.14 Linear Transformation

Although the error rates for individual images are more variable when there are fewer layers, particularly in the case of quality layers in which fewer bits are extracted, it is clear (figures 5.27 and 5.28) that all images fit well with the observed overall behaviour (section 5.2.1.6) of 50% errors in all subimages. Images 6 (most noticeably), 9, 10 and 16

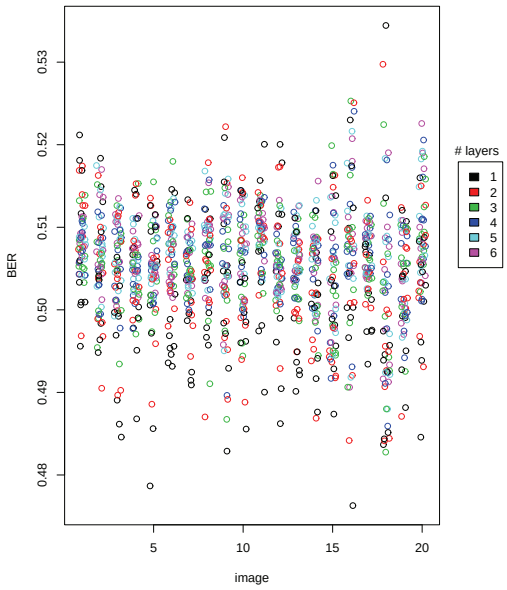


Figure D.28: BER for resolution scaled subimages: Linear.

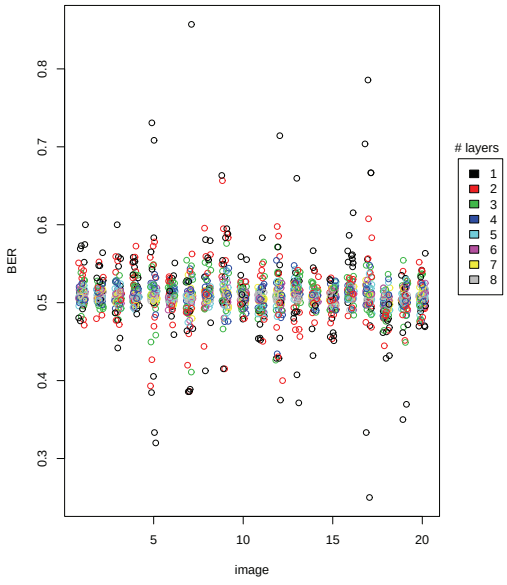


Figure D.29: BER for quality scaled subimages: Linear.

have lower errors, relative to the baseline of the corresponding recompressed image, than do other images. This appears to be an artifact of the higher baseline error rates for these images rather than anything intrinsic to linear transformation.

D.4.1.15 Projective Transformation

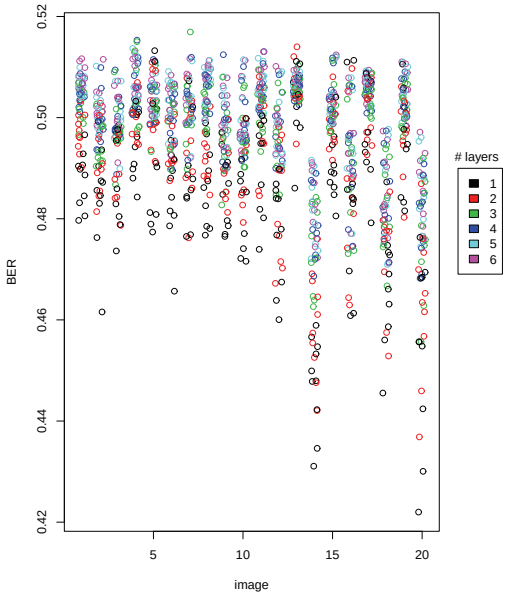


Figure D.30: Total bit error rates for resolution scaled subimages: Projective.

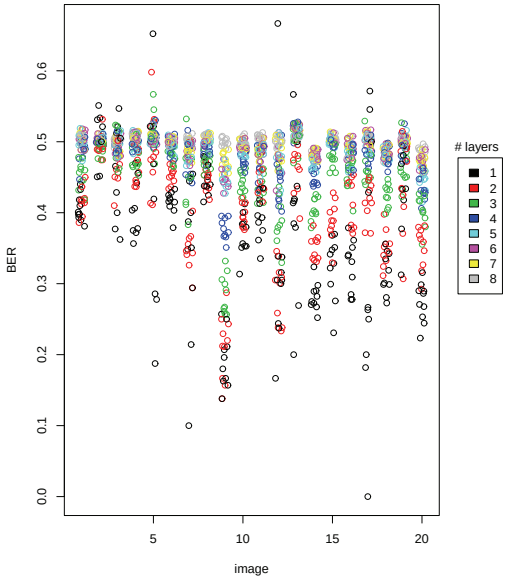


Figure D.31: Total bit error rates for quality scaled subimages: Projective.

The majority of individual images have bit error rates (figures 5.29 and 5.30) that are similar to the overall results for projective transformation (section 5.2.1.6). However, some

individual images (14, 18, 20) have noticeably lower error rates, particularly when only a few resolution or quality layers have been received. These are images in which the borders are relatively smooth and the significant coefficients are clustered towards the centre of the image, which is less affected by the transformation.

D.4.1.16 Rotation

The bit error rates for individual images (figures D.32, D.33, D.34 and D.35) follow the same pattern as the overall rates for both the 1 deg and 45 deg rotations (section 5.2.1.6). Subimages composed of fewer layers have more variable BERs, because they contain fewer watermark bits.

D.4.1.17 Scale

Results for individual images (figures D.36 and D.37) are consistent with the overall results (section 5.2.1.5), although the error rates are more variable for images with fewer layers as these contain fewer extracted bits. In the two lowest quality subimages for some image and seed combinations[2] all watermark bits were lost and error rates for these subimages could not be computed.

D.4.1.18 Copy

Results for individual images (figures D.38 and D.39) are consistent with the overall 50% BER results (section 5.2.1.5). The error rates are more variable for images with fewer layers as these contain fewer extracted bits. This is most evident in the quality scaled images, in which the number of non-missing watermark bits is particularly low.

[2]No watermark bits could be extracted from the lowest quality subimage for: image 7, seed 1; image 12, seeds 2 and 10; image 13, seed 2 and image 17, seeds 8 and 9. In the second lowest quality subimage, there were still no watermark bits extractable from image 12, seeds 2 and 10.

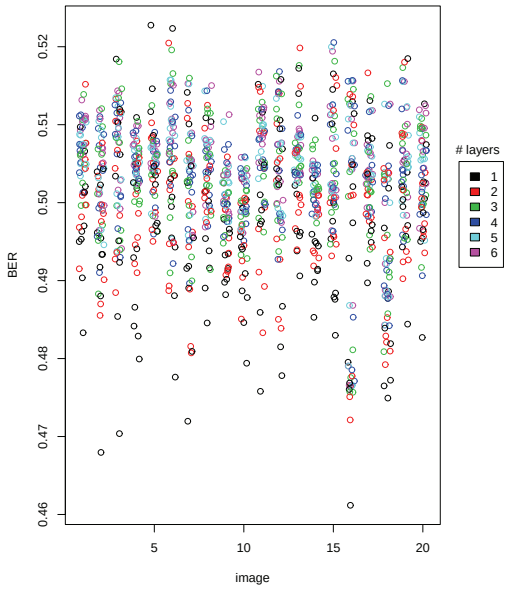


Figure D.32: Total bit error rates for resolution scaled subimages: Rotation 1 deg.

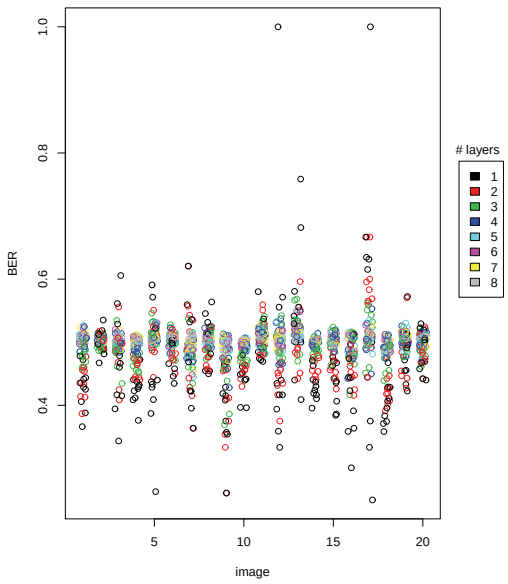


Figure D.33: Total bit error rates for quality scaled subimages: Rotation 1 deg.

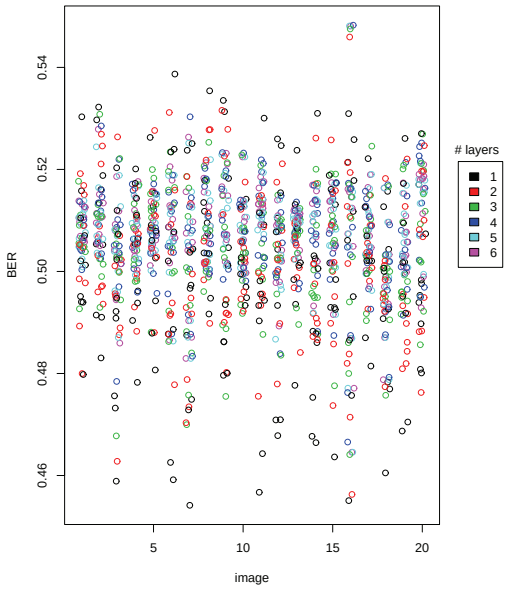


Figure D.34: Total bit error rates for resolution scaled subimages: Rotation 45 deg.

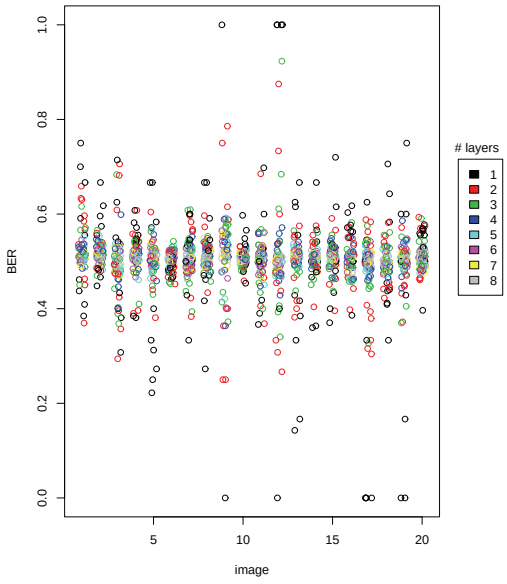


Figure D.35: Total bit error rates for quality scaled subimages: Rotation 45 deg.

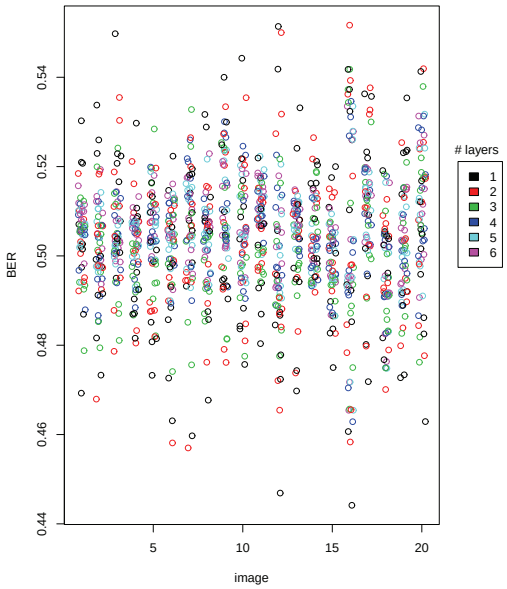


Figure D.36: Total bit error rates for resolution scaled subimages: Scale50.

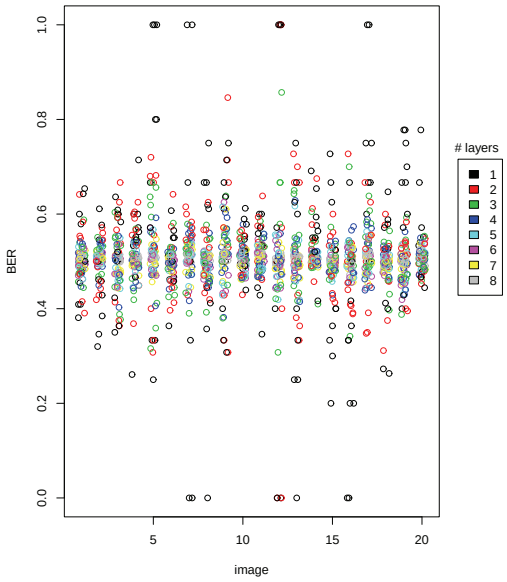


Figure D.37: Total bit error rates for quality scaled subimages: Scale50.

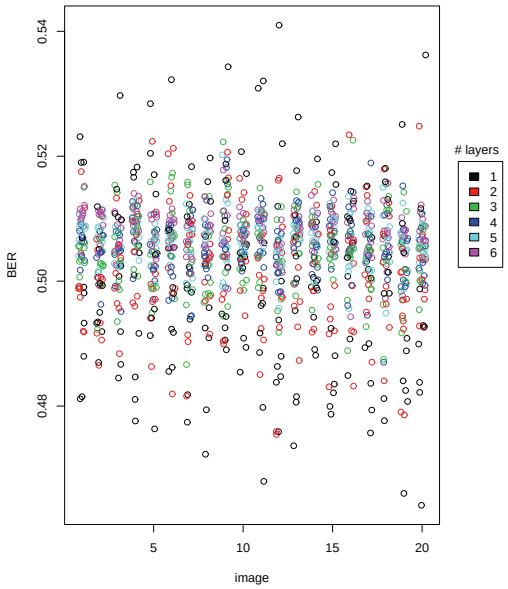


Figure D.38: Total bit error rates for resolution scaled subimages: Copy.

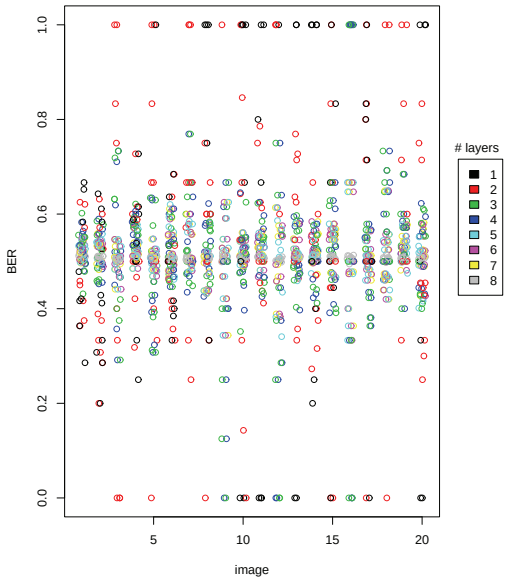


Figure D.39: Total bit error rates for quality scaled subimages: Copy.

D.4.2 Tamper Detection

This section contains full set of tampered images and corresponding tamper maps, i.e. at all tested levels of resolution and quality scaling, for the various methods of deliberate tampering discussed in section 5.2.3.

D.4.2.1 Detection Without Tampering

After resolution scaling (figs. D.40 through D.49), all five reduced resolution subimages produce the output $\gamma = true$, indicating no tampering. While all extracted watermark bits are correct, as resolution layers are dropped fewer watermark bits can be extracted (Tab. D.2), so the security of lower resolution versions is less than that of higher resolution versions.

Table D.2: Number of extracted watermark bits at varying resolutions.

# resolution layers	errors	watermark bits
6	0	38653
5	0	28956
4	0	17022
3	0	9529
2	0	4814
1	0	2670

Figure D.40: Watermarked image, $\frac{1}{4}$ area of original.

Figure D.41: Tamper map for the watermarked image, $\frac{1}{4}$ area of original.

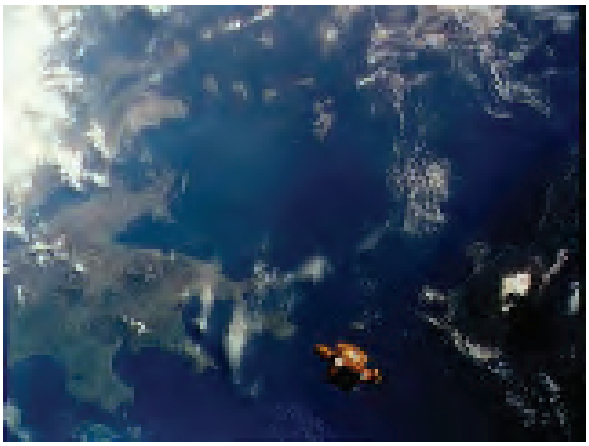


Figure D.42: Watermarked image, $\frac{1}{16}$th area of original.

Figure D.43: Tamper map for the watermarked image, $\frac{1}{16}$th area of original.

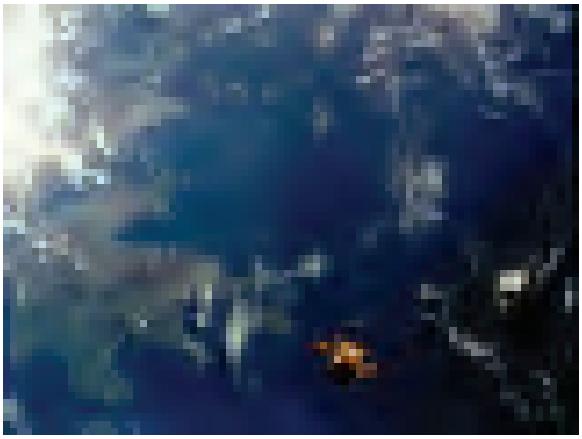


Figure D.44: Watermarked image, $\frac{1}{64}$th area of original.

Figure D.45: Tamper map for the watermarked image, $\frac{1}{64}$th area of original.

Figure D.46: Watermarked image, $\frac{1}{256}$th area of original.

Figure D.47: Tamper map for the watermarked image, $\frac{1}{256}$th area of original.

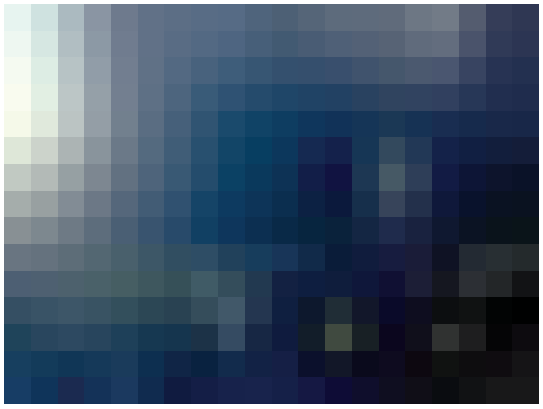


Figure D.48: Watermarked image, $\frac{1}{1024}$th area of original.

Figure D.49: Tamper map for the watermarked image, $\frac{1}{1024}$th area of original.

After quality scaling (figs. D.50 through D.57), all subimages give $\gamma = true$, indicating no tampering. All extracted watermark bits are correct, and fewer watermark bits can be extracted from lower quality images (Tab. D.3).

Table D.3: Number of extracted watermark bits at varying quality levels.

# quality layers	compression rate	errors	watermark bits
5	0.9999	0	38653
4	0.06	0	9036
3	0.04	0	6895
2	0.02	0	4622
1	0.01	0	3691

Figure D.50: Watermarked image, compression rate 0.06.

Figure D.51: Tamper map for the watermarked image, compression rate 0.06.

Figure D.52: Watermarked image, compression rate 0.04.

Figure D.53: Tamper map for the watermarked image, compression rate 0.04.

Figure D.54: Watermarked image, compression rate 0.02.

Figure D.55: Tamper map for the watermarked image, compression rate 0.02.

Figure D.56: Watermarked image, compression rate 0.01.

Figure D.57: Tamper map for the watermarked image, compression rate 0.01.

D.4.2.2 Spatial Tampering

After resolution scaling (figs. D.58 through D.49), all subimages produce the output $\gamma = false$, indicating the spatial tampering has been detected. Detection on the lowest resolution subimage (figs. D.66 and D.67) gives $\gamma = false$ with 949 errors from 2680 bits.

After quality scaling (figs. D.68 through D.57) all subimages produce the output $\gamma = false$, indicating the spatial tampering has been detected. Detection on the lowest quality subimage (figs. D.74 and D.75) gives $\gamma = false$ with 1188 errors from 4280 bits.

Figure D.58: Spatially tampered image, $\frac{1}{4}$ area of original.

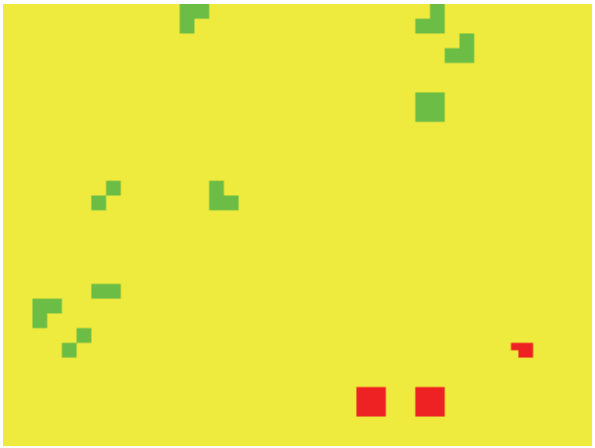


Figure D.59: Tamper map for the spatially tampered image, $\frac{1}{4}$ area of original.

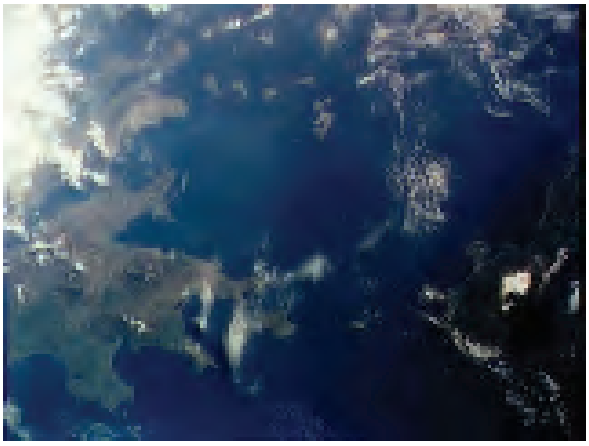


Figure D.60: Spatially tampered image, $\frac{1}{16}$th area of original.

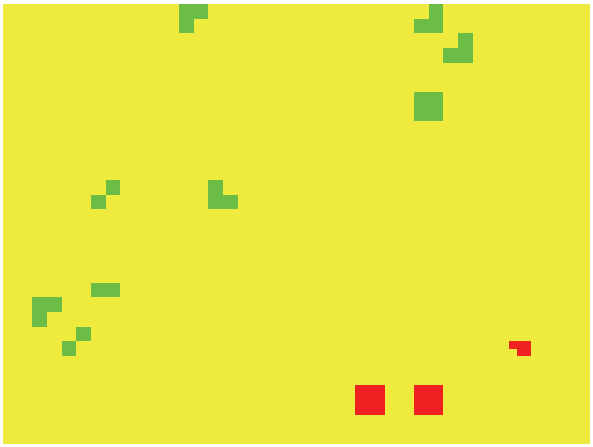


Figure D.61: Tamper map for the spatially tampered image, $\frac{1}{16}$th area of original.

Figure D.62: Spatially tampered image, $\frac{1}{64}$th area of original.

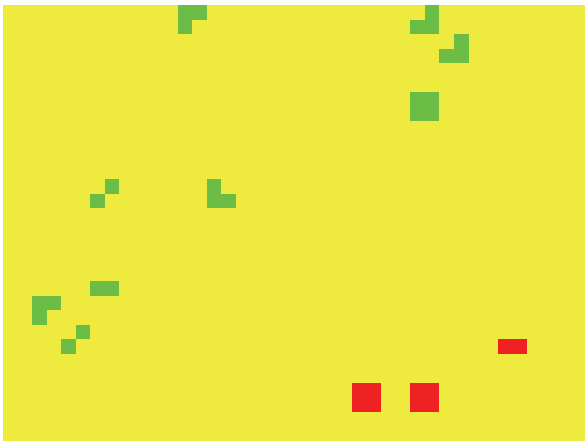


Figure D.63: Tamper map for the spatially tampered image, $\frac{1}{64}$th area of original.

Figure D.64: Spatially tampered image, $\frac{1}{256}$th area of original.

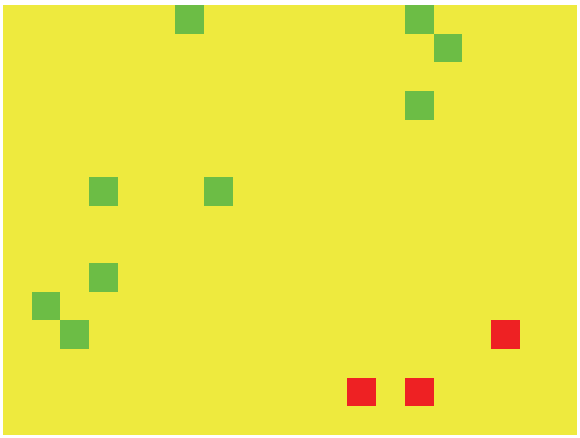


Figure D.65: Tamper map for the spatially tampered image, $\frac{1}{256}$th area of original.

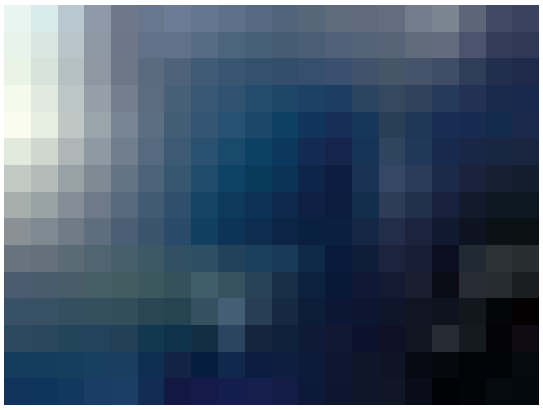


Figure D.66: Spatially tampered image, $\frac{1}{1024}$th area of original.

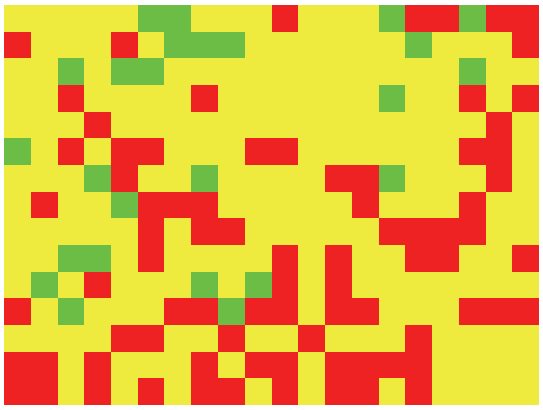


Figure D.67: Tamper map for the spatially tampered image, $\frac{1}{1024}$th area of original.

Figure D.68: Spatially tampered image, compression rate 0.06.

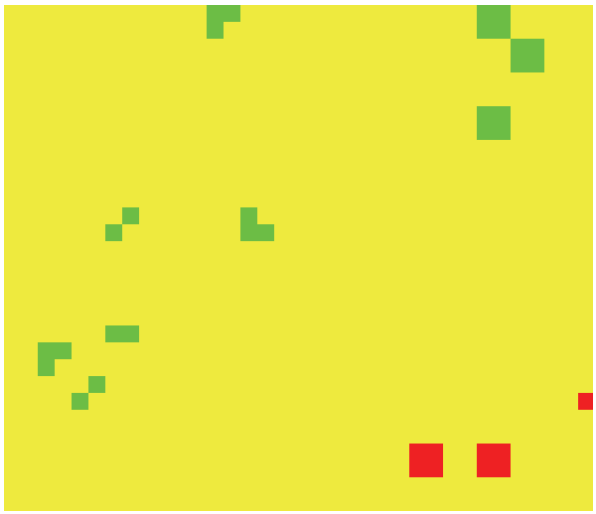


Figure D.69: Tamper map for the spatially tampered image, compression rate 0.06.

D.4.2.3 Tampering in the Wavelet Domain

After resolution scaling (figs. D.76 through D.49), all subimages produce the output $\gamma = false$, indicating the wavelet tampering has been detected. After quality scaling (figs. D.86 through D.57) all subimages produce the output $\gamma = false$, indicating the wavelet tampering has been detected.

Detection on the lowest resolution subimage (fig. D.85) gives $\gamma = false$ with 209 errors from 2680 bits and detection on the lowest quality subimage (fig. D.93) gives $\gamma = false$ with 259 errors from 3691 bits.

Figure D.70: Spatially tampered image, compression rate 0.04.

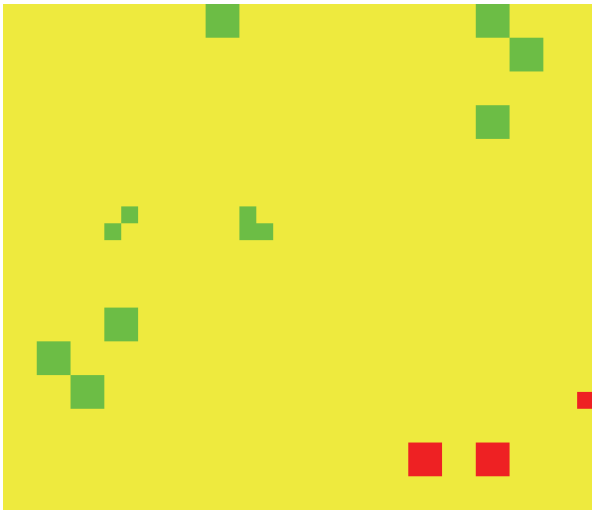


Figure D.71: Tamper map for the spatially tampered image, compression rate 0.04.

Figure D.72: Spatially tampered image, compression rate 0.02.

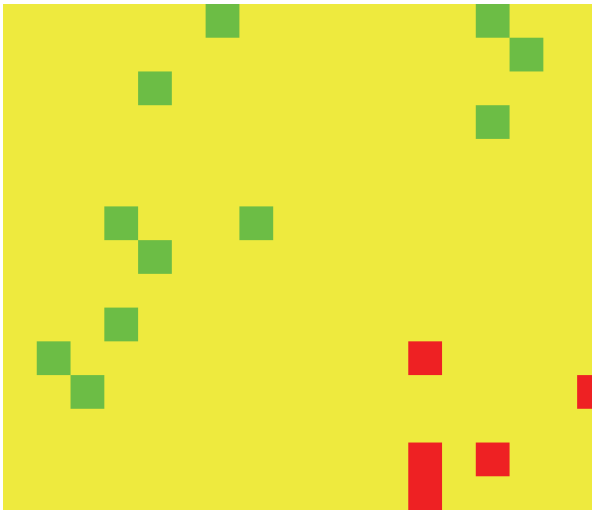


Figure D.73: Tamper map for the spatially tampered image, compression rate 0.02.

Figure D.74: Spatially tampered image, compression rate 0.01.

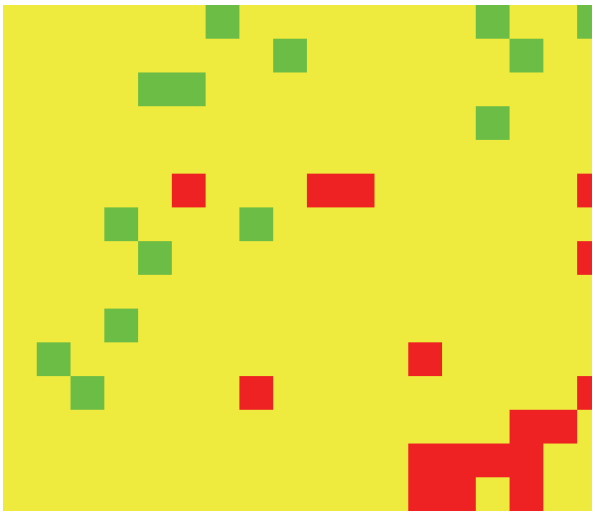


Figure D.75: Tamper map for the spatially tampered image, compression rate 0.01.

Figure D.76: Wavelet tampered image, $\frac{1}{4}$ area of original.

Figure D.77: Tamper map for the wavelet tampered image, $\frac{1}{4}$ area of original.

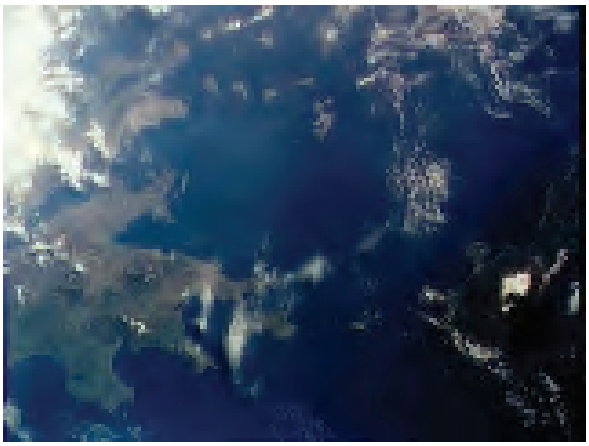


Figure D.78: Wavelet tampered image, $\frac{1}{16}$th area of original.

Figure D.79: Tamper map for the wavelet tampered image, $\frac{1}{16}$th area of original.

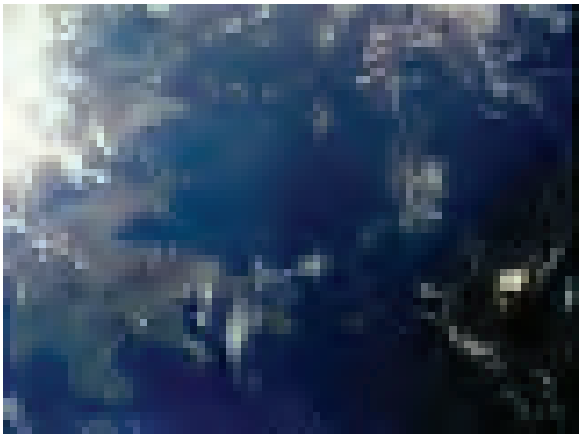


Figure D.80: Wavelet tampered image, $\frac{1}{64}$th area of original.

Figure D.81: Tamper map for the wavelet tampered image, $\frac{1}{64}$th area of original.

Figure D.82: Wavelet tampered image, $\frac{1}{256}$th area of original.

Figure D.83: Tamper map for the wavelet tampered image, $\frac{1}{256}$th area of original.

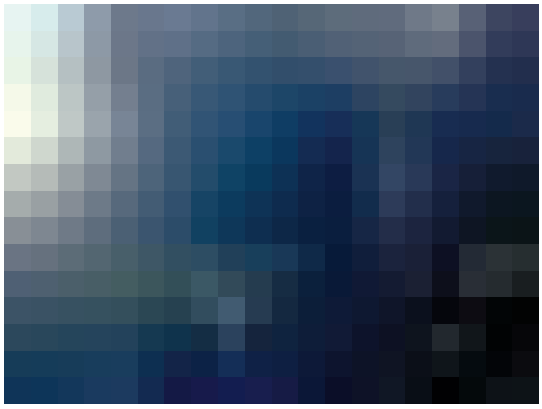


Figure D.84: Wavelet tampered image, $\frac{1}{1024}$th area of original.

Figure D.85: Tamper map for the wavelet tampered image, $\frac{1}{1024}$th area of original.

Figure D.86: Wavelet tampered image, compression rate 0.06.

Figure D.87: Tamper map for the wavelet tampered image, compression rate 0.06.

D.4.2.4 Mark Transfer

Tampering using the mark transfer attack is identified at all levels of scaling (figs. D.94 through D.110). At the lowest resolution level (fig. D.102) there are 9 errors from 2680 bits. The tamper map (fig. D.103) still shows the tampered region and the detector output is still $\gamma = false$. At the lowest quality level (fig. D.110) we obtain 16 errors from 3691 bits. The tamper map (fig. D.111) still shows the tampered region and the detector output is still $\gamma = false$.

Figure D.88: Wavelet tampered image, compression rate 0.04.

Figure D.89: Tamper map for the wavelet tampered image, compression rate 0.04.

Figure D.90: Wavelet tampered image, compression rate 0.02.

Figure D.91: Tamper map for the wavelet tampered image, compression rate 0.02.

Figure D.92: Wavelet tampered image, compression rate 0.01.

Figure D.93: Tamper map for the wavelet tampered image, compression rate 0.01.

Figure D.94: Mark-transferred image, $\frac{1}{4}$ area of original.

Figure D.95: Tamper map for the mark-transferred image, $\frac{1}{4}$ area of original.

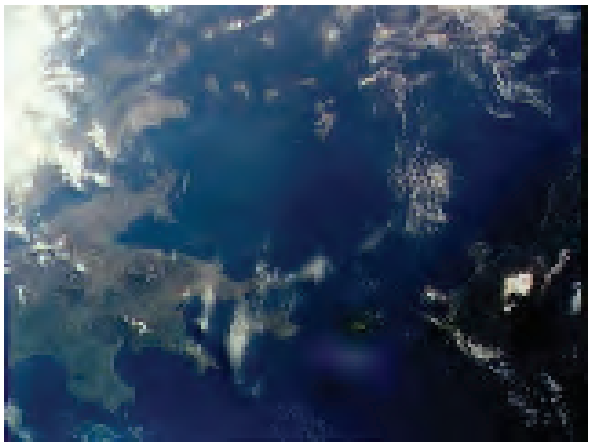


Figure D.96: Mark-transferred image, $\frac{1}{16}$th area of original.

Figure D.97: Tamper map for the mark-transferred image, $\frac{1}{16}$th area of original.

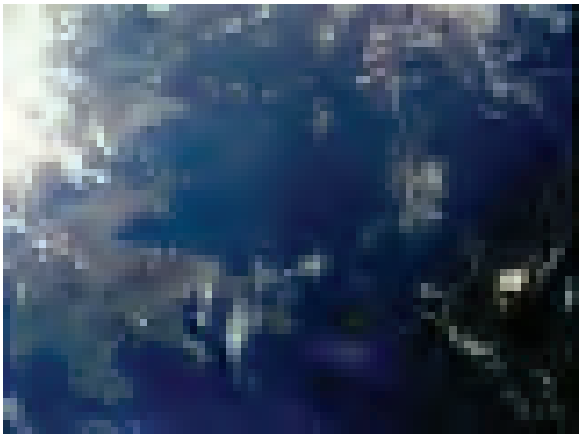


Figure D.98: Mark-transferred image, $\frac{1}{64}$th area of original.

Figure D.99: Tamper map for the mark-transferred image, $\frac{1}{64}$th area of original.

Figure D.100: Mark-transferred image, $\frac{1}{256}$th area of original.

Figure D.101: Tamper map for the mark-transferred image, $\frac{1}{256}$th area of original.

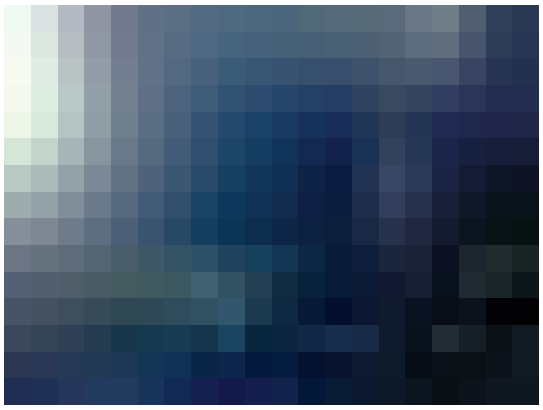


Figure D.102: Mark-transferred image, $\frac{1}{1024}$th area of original.

Figure D.103: Tamper map for the mark-transferred image, $\frac{1}{1024}$th area of original.

Figure D.104: Mark-transferred image, compression rate 0.06.

Figure D.105: Tamper map for the mark-transferred image, compression rate 0.06.

Figure D.106: Mark-transferred image, compression rate 0.04.

Figure D.107: Tamper map for the mark-transferred image, compression rate 0.04.

Figure D.108: Mark-transferred image, compression rate 0.02.

Figure D.109: Tamper map for the mark-transferred image, compression rate 0.02.

Figure D.110: Mark-transferred image, compression rate 0.01.

Figure D.111: Tamper map for the mark-transferred image, compression rate 0.01.

D.4.3 Tampering in a Small Region

Section 5.2.3.7.1 showed that for a sufficiently small tampered region, mark transfer with quality scaling was successful. This leaves open the possibility that less sophisticated attacks may also be successful for a sufficiently small tampered region. This section shows that all the less sophisticated attacks with the same target image and tampered region are unsuccessful.

The original Lena (fig. 5.19) is watermarked using the same parameters (section 5.2.3.2), producing the same watermarked image (fig. D.113), which is not visibly different from the original.

Figure D.112: The original Lena image.

Detection using the untampered watermarked image produces the correct output $\gamma = true$, with 0 errors from 26419 extracted bits (fig. D.114). The same is true for the lowest resolution scaled image (figs. D.115 and D.116), which gives 0 errors from 2670 bits, and for the lowest quality scaled image (figs. D.117 and D.118), which gives 0 errors from 3691 bits.

D.4.3.1 Tampering in the Spatial Domain

In the spatially tampered image (figure D.119), the change is visible and the image quality has not been impaired. Detection shows $\gamma = false$ with 1533 errors from 26429 extracted bits (fig. D.120).

Figure D.113: Watermarked image.

Figure D.114: Tamper map for the watermarked Lena image.

Figure D.115: Watermarked image, $\frac{1}{1024}$th area of original.

Figure D.116: Tamper map for the watermarked Lena image, $\frac{1}{1024}$th area of original.

Figure D.117: Watermarked image, compression rate 0.01.

Figure D.118: Tamper map for the watermarked Lena image, compression rate 0.01.

Figure D.119: Spatially tampered image.

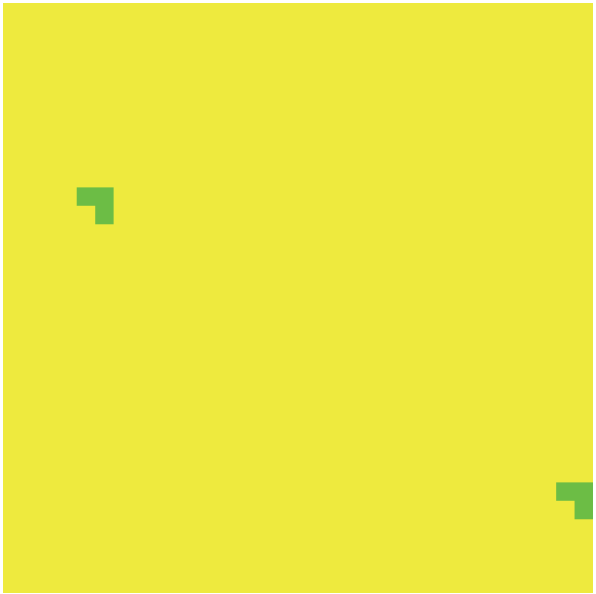


Figure D.120: Tamper map for the spatially tampered image.

Because the tampered region is so small, when the resolution scaled image reaches $\frac{1}{64}$th the original area (fig. D.121), the loss of spatial detail renders the change no longer visible. This is the case regardless of which method of tampering is used.

For the spatially tampered image, watermark detection still reveals errors even at the lowest resolution level (figs. D.122 and D.123), with 673 errors from 2359 bits. At the lowest quality level we find the change is also not visible (fig. D.124), yet watermark detection reveals 949 errors from 3696 bits (fig. D.125.

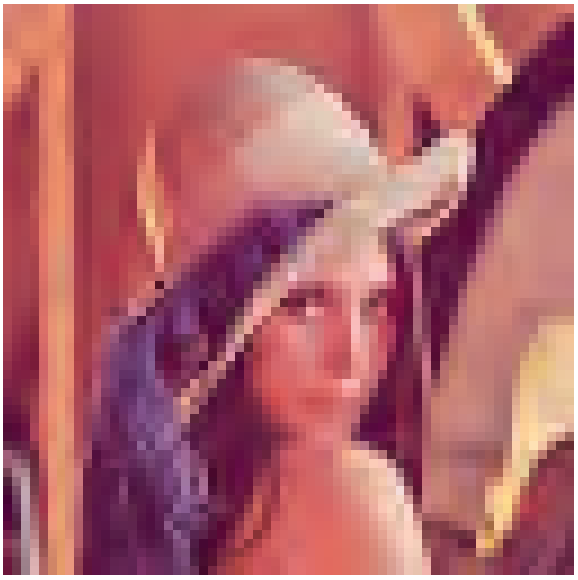


Figure D.121: Spatially image, $\frac{1}{64}$th area of original.

Figure D.122: Spatially image, $\frac{1}{1024}$th area of original.

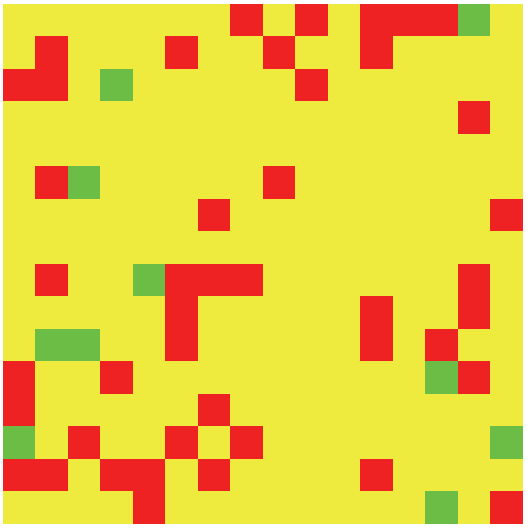


Figure D.123: Tamper map for the spatially tampered image, $\frac{1}{1024}$th area of original.

Figure D.124: Watermarked image, compression rate 0.01.

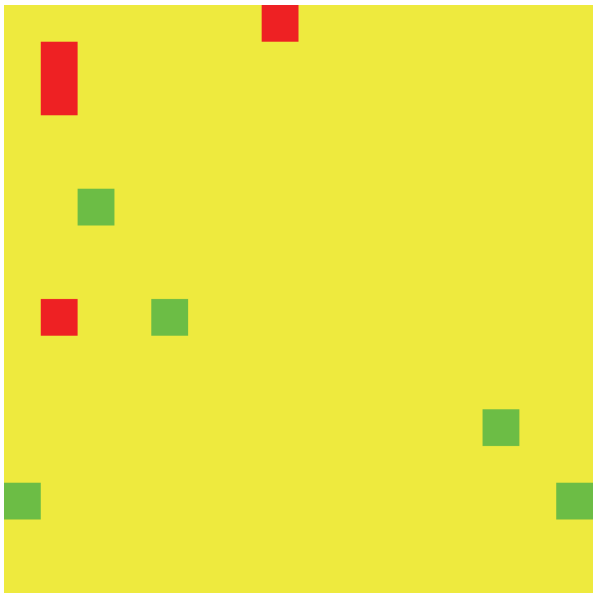


Figure D.125: Tamper map for the watermarked Lena image, compression rate 0.01.

D.4.3.2 Tampering in the Wavelet Domain

In the wavelet tampered image (figure D.126), the spot is clearly visible and the image quality has not been impaired. Detection produces $\gamma = false$ with 142 errors from 26390 extracted bits and the tamper map shown in figure D.127.

As with the spatially tampered image, the spot is no longer visible after resolution scaling to $\frac{1}{64}$th the original area, or at compression rate 0.01, yet detection still shows errors even for the lowest resolution and lowest quality images.

At the lowest resolution (fig. D.128), the detector outputs $\gamma = false$ with 68 errors from 2360 extracted bits (fig. D.129). At the lowest quality (fig. D.130), the detector outputs $\gamma = false$ with 111 errors from 3721 extracted bits (fig. D.131).

D.4.3.3 Mark Transfer Attack

After applying mark transfer, replacing all but the MSB of each coefficient in the tampered region, the change is clearly visible and, unlike the same method applied to the Greek isles image (in section 5.2.3.7), does not impair the image quality (fig. D.132). Detection gives $\gamma = false$, with 6 errors from 26390 bits (fig. D.133).

As with previous methods, the change is no longer visible at the third-lowest resolution, $\frac{1}{64}$th the area of the original, or below. Thus, although the output of detection at $\frac{1}{256}$th and $\frac{1}{1024}$th the are of the original is $\gamma = true$, no meaningful image alterations are present at these resolutions so the attack is considered unsuccessful.

At $\frac{1}{64}$th the area of the original or above, we obtain the output $\gamma = false$. For the resolution scaled image with $\frac{1}{64}$th the area of the original we obtain 2 errors from 9044

Figure D.126: Wavelet tampered image.

Figure D.127: Tamper map for the wavelet tampered image.

Figure D.128: Wavelet tampered image, $\frac{1}{1024}$th area of original.

Figure D.129: Tamper map for the wavelet tampered image, $\frac{1}{1024}$th area of original.

Figure D.130: Wavelet tampered image, compression rate 0.01.

Figure D.131: Tamper map for the wavelet tampered image, compression rate 0.01.

bits (fig. D.135).

For the third-lowest quality image, compression rate 0.04, the change is still visible (fig. D.136); we obtain the output $\gamma = false$, with 1 error from 7454 bits (fig. D.137).

However, as can be seen in section 5.2.3.7.1, mark transfer with a small tampered region and a correct amount of quality scaling can result in a successful attack.

Figure D.132: Tampered image with transferred watermark.

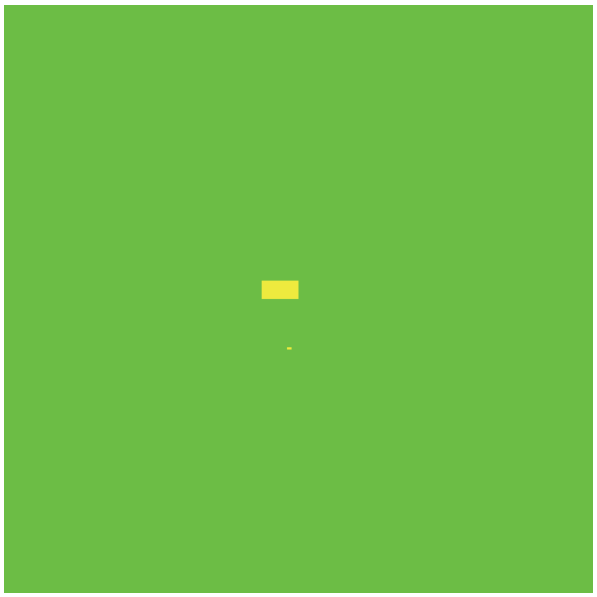


Figure D.133: Tamper map for the tampered image with transferred watermark.

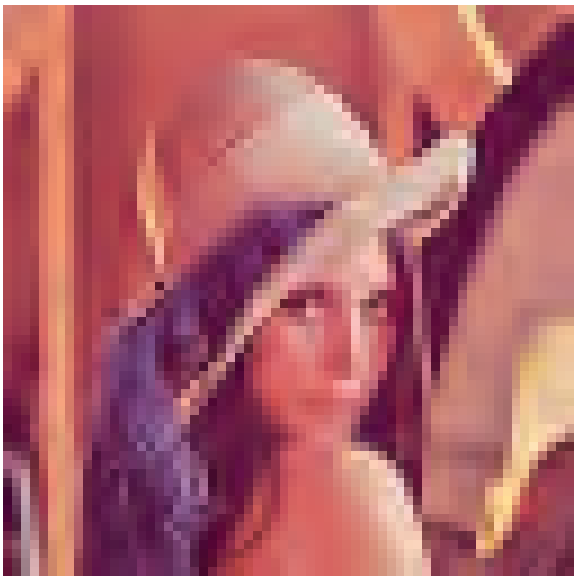


Figure D.134: Tampered image with transferred watermark, $\frac{1}{64}$th area of original.

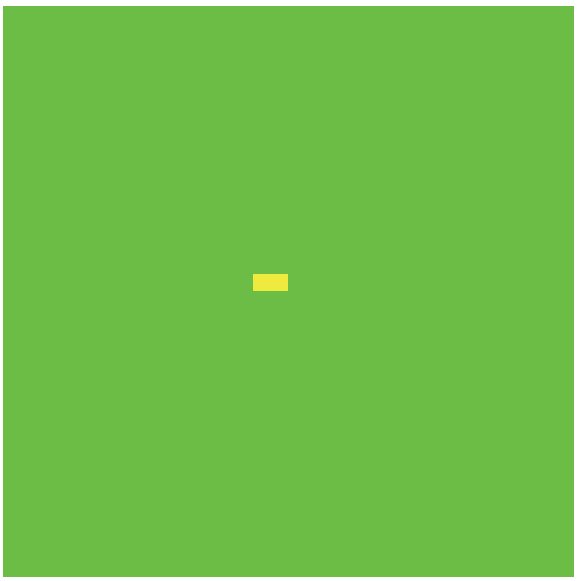


Figure D.135: Tamper map for the tampered image with transferred watermark, $\frac{1}{64}$th area of original.

Figure D.136: Tampered image with transferred watermark, compression rate 0.04.

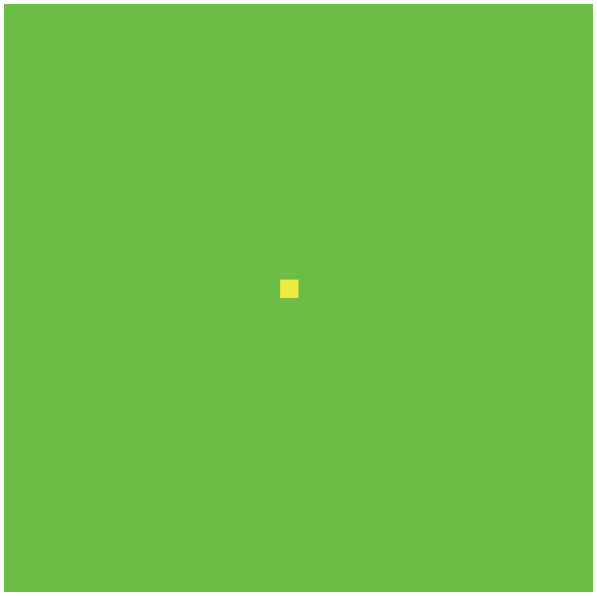


Figure D.137: Tamper map for the tampered image with transferred watermark, compression rate 0.04.

D.4.3.4 Collage Attack

Using a collage of the watermarked image and five additional (identically watermarked) images, we obtain the image in figure D.138. In contrast to the Greek isles image, there are no visible artifacts in the tampered region but, instead, the modification is not yet visible. Thus, despite a detection output of $\gamma = true$ with 0 errors from 26380 bits (fig D.139), the tampered image is not meaningfully different from the watermarked image so the attack is unsuccessful.

Figure D.138: Tampered image with 6-image collage.

Figure D.139: Tamper map for the tampered image with 6-image collage.

Increasing the number of additional images used in the collage to twenty, produces the tampered image in figure D.140. The modification is clearly visible in this image, though not as strongly as in the wavelet tampered image, and there are no visible artifacts. We obtain the output $\gamma = true$ with 0 errors from 26382 bits (fig D.141), so the collage attack with twenty-one images is successful.

Figure D.140: Tampered image with 21-image collage.

Figure D.141: Tamper map for the tampered image with 21-image collage.

A collage attack with eighty-one images is also successful, with 0 errors from 26386 bits (fig D.143). The availability of a larger range of watermarked coefficients allows a better reproduction of the tampered region, improving the match between the tampered

image (fig. D.142) and the intended modification (fig. D.119) over the twenty-one image collage.

Figure D.142: Tampered image with 21-image collage.

Figure D.143: Tamper map for the tampered image with 81-image collage.

D.5 Identifying Coefficients Corresponding to a Rectangular Region

This section discusses the implementation of tampering in the wavelet domain used in section 5.2.3.5 of the thesis. Specifically, it describes precisely how we may identify the "coefficients required to reconstruct a rectangular region of tampered pixels", that are copied from the spatially tampered image to the valid watermarked image.

Tampering in the wavelet domain is somewhat more complex than spatial tampering, because it is not immediately apparent exactly which coefficients must be altered and to what, in order to make the desired changes to the image content without introducing any visible artifacts.

We can determine this information using the spatially tampered image. Let the *tampered region* be the set of pixels that have been deliberately altered to change the image content and let the *untampered region* be all other pixels in the image, for which there should be no perceptible difference between the spatially tampered and watermarked images.

Knowing the compression parameters, an attacker can determine the set of *tampered coefficients* required for reconstruction of the tampered region during decompression (details of how this is done can be found in section D.5). If the tampered coefficients from the spatially tampered image, are used to replace the corresponding set of coefficients in the watermarked image, while leaving the remaining coefficients untouched, the resulting wavelet tampered image will contain the desired content changes in the tampered region, without either visible artifacts or watermark errors in the untampered region[3].

Given the coordinates of a rectangular image region in the spatial domain, if we know the wavelet transform and number of decomposition levels, we can determine which wavelet coefficients will be required to reconstruct the given spatial region.

This is done by first identifying the coefficients required to reconstruct a single pixel from a single level decomposition in a single dimension, and then extending this result to two dimensions, multiple decomposition levels and a rectangular region of pixels.

D.5.1 Reconstruction of pixel x from a 1D transformed row of pixels

Because each 2D wavelet decomposition can be thought of as two separate applications of a 1D wavelet decomposition in the row and column directions. We will first consider the

[3]This is equivalent to removing the watermark errors from the spatially tampered image by replacing all coefficients which do not contribute to the tampered region with their counterparts in the watermarked image.

reconstruction of a pixel at coordinate x in the image row Y,

$$Y = \{Y(0), Y(1), \ldots, Y(x), \ldots, Y(l)\}$$

from the row-transformed coefficients

$$\mathrm{L} = \{\mathrm{L}(0), \ldots, \mathrm{L}(\lceil \frac{l}{2} \rceil)\} \quad \text{and}$$
$$\mathrm{H} = \{\mathrm{H}(0), \ldots, \mathrm{H}(\lfloor \frac{l}{2} \rfloor)\}$$

We are most interested in the Daubechies 9/7 wavelet transform, the real valued filter specified in part 1 of the JPEG2000 standard, so we will consider reconstruction for this transform; the procedure is identical for other transforms but the required coefficients will vary. The equations for the lifting implementation of the Daubechies 9/7 IDWT for JPEG2000[4] are as follows (where $m \in \mathbb{Z}$)

$$Y_a(2m) = K\mathrm{L}(m) \qquad [STEP1]$$
$$Y_a(2m+1) = \frac{-1}{K}\mathrm{H}(m) \qquad [STEP2]$$
$$Y_b(2m) = Y_a(2m) - \delta\big(Y_a(2m-1) + Y_a(2m+1)\big) \qquad [STEP3]$$
$$Y_b(2m+1) = Y_a(2m+1) - \gamma\big(Y_b(2m) + Y_b(2m+2)\big) \qquad [STEP4]$$
$$Y(2m) = Y_b(2m) - \beta\big(Y_b(2m-1) + Y_b(2m+1)\big) \qquad [STEP5]$$
$$Y(2m+1) = Y_b(2m+1) - \alpha\big(Y(2m) + Y(2m+2)\big). \qquad [STEP6]$$

Thus the coefficients required to reconstruct pixel $Y(x)$ depend on whether its coordinate is even or odd (see figure D.144). If $x = 2m$ is even, we require high-pass coefficients $\mathrm{H}(m-2)$ to $\mathrm{H}(m+1)$ (inclusive) and low-pass coefficients $\mathrm{L}(m-1)$ to $\mathrm{L}(m+1)$. If $x = 2m+1$ is odd, we require high-pass coefficients $\mathrm{H}(m-2)$ to $\mathrm{H}(m+2)$ and low-pass coefficients $\mathrm{L}(m-1)$ to $\mathrm{L}(m+2)$.

D.5.2 Extension to 2D reconstruction

Because the 2D wavelet transform is simply the same 1D transform applied in both the row and column directions, extension to two dimensions is trivial. To reconstruct pixel

[4]The equations presented were obtained from [78] where K, α, β, γ and δ are constants with values

$\alpha = -1.586134342$ $\quad \beta = -0.052980118$ $\quad K = 1.230174105$
$\gamma = 0.882911075$ $\quad \delta = 0.443506852$

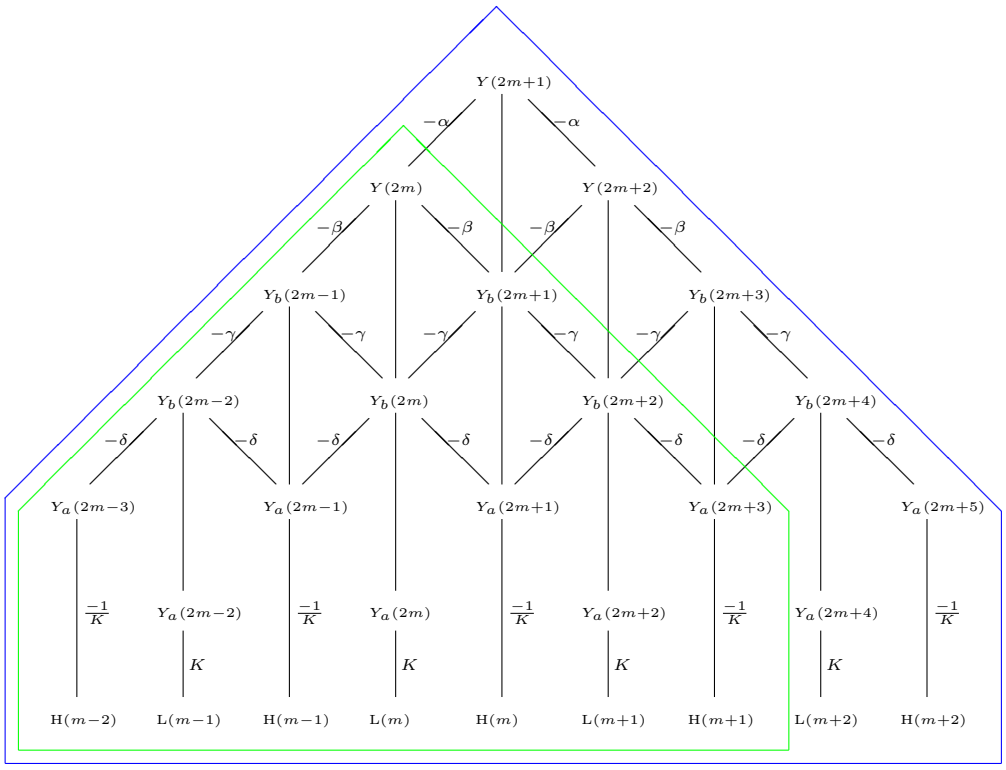


Figure D.144: Reconstruction of pixels, $Y(2m+1)$ outlined in blue and $Y(2m)$ outlined in green, from the high-pass (H($m-2$) to H($m+2$)) and low-pass (L($m-1$) to L($m+2$)) coefficients after a single 1D decomposition using a Daubechies 9/7 DWT. Intermediate lifting steps are shown. Pixels with even coordinates require two fewer coefficients for reconstruction.

$I(x, y)$ from coefficients in the LL, HL, LH and HH subbands, the number of coefficients required from each subband will depend on the oddness or evenness of x and y.

If $x = 2m + b$ and $y = 2n + c$ where $m, n \in \mathbb{Z}, b, c \in \{0, 1\}$ we require coefficients in the rectangular regions

$$\begin{array}{ll} \text{LL} & (m-1, n-1) \text{ to } (m+1+b, n+1+c), \\ \text{HL} & (m-2, n-1) \text{ to } (m+1+b, n+1+c), \\ \text{LH} & (m-1, n-2) \text{ to } (m+1+b, n+1+c) \qquad \text{and} \\ \text{HH} & (m-2, n-2) \text{ to } (m+1+b, n+1+c). \end{array}$$

Rewriting this in terms of the coordinates x and y gives

$$\begin{array}{ll} \text{LL} & (\lfloor \frac{x-2}{2} \rfloor \quad , \lfloor \frac{y-2}{2} \rfloor \quad) \text{ to } (\lceil \frac{x+2}{2} \rceil, \lceil \frac{y+2}{2} \rceil), \\ \text{HL} & (\lfloor \frac{x-2}{2} \rfloor - 1, \lfloor \frac{y-2}{2} \rfloor \quad) \text{ to } (\lceil \frac{x+2}{2} \rceil, \lceil \frac{y+2}{2} \rceil), \\ \text{LH} & (\lfloor \frac{x-2}{2} \rfloor \quad , \lfloor \frac{y-2}{2} \rfloor - 1) \text{ to } (\lceil \frac{x+2}{2} \rceil, \lceil \frac{y+2}{2} \rceil) \qquad \text{and} \\ \text{HH} & (\lfloor \frac{x-2}{2} \rfloor - 1, \lfloor \frac{y-2}{2} \rfloor - 1) \text{ to } (\lceil \frac{x+2}{2} \rceil, \lceil \frac{y+2}{2} \rceil). \end{array}$$

D.5.3 Extension to multiple decomposition levels

The preceding coefficients are sufficient for pixel reconstruction if the transformed image was the result of a single 2D wavelet decomposition (using Daubechies 9/7 filters). If the transformed image was constructed by R applications of the 2D wavelet transform, to reconstruct pixel $I(x, y) = R\text{LL}(x, y)$ we require coefficients with those coordinates from the $(R-1)$LL, $(R-1)$HL, $(R-1)$LH and $(R-1)$HH subbands at the previous resolution level.

If $R > 1$, the $(R-1)$LL subband will not be immediately available so each required coefficient in $(R-1)$LL must be reconstructed from coefficients in the $(R-2)$LL, $(R-2)$HL, $(R-2)$LH and $(R-2)$HH subbands. Because $(R-1)$LL was decomposed in the same manner as RLL, the reconstruction procedure is the same as for reconstructing the pixel RLL(x,y) except we must now reconstruct a rectangular region of coefficients: $(R–1)\text{LL}(\lfloor \frac{x-2}{2} \rfloor, \lfloor \frac{y-2}{2} \rfloor)$ to $(R–1)\text{LL}(\lceil \frac{x+2}{2} \rceil, \lceil \frac{y+2}{2} \rceil)$ If $R > 2$, the $(R–2)$LL subband will not be immediately available so each required coefficient in $(R-2)$LL must be reconstructed from coefficients in the previous resolution ($(R-3)$LL, $(R-3)$HL, $(R-3)$LH, $(R-3)$HH), and so on.

For simplicity, we will examine the procedure for a single dimension; to reconstruct pixel $Y(x) = R\mathrm{L}(x)$ we will need to reconstruct coefficients $(R-1)\mathrm{L}(\lfloor\frac{x-2}{2}\rfloor)$ to $(R-1)\mathrm{L}(\lceil\frac{x+2}{2}\rceil)$ in the $(R-1)$L subband from the $(R-2)$L and $(R-2)$H subbands. Because the reconstruction procedure is identical for all resolution levels, this will require

$$\begin{array}{ll} (R-2)\mathrm{L} & \lfloor\frac{t-2}{2}\rfloor \quad \text{to } \lceil\frac{t+2}{2}\rceil \\ (R-2)\mathrm{H} & \lfloor\frac{t-2}{2}\rfloor - 1 \text{ to } \lceil\frac{t+2}{2}\rceil, \end{array}$$

where t represents the coordinate of the required coefficients, and thus ranges from $\lfloor\frac{x-2}{2}\rfloor$ to $\lceil\frac{x+2}{2}\rceil$. Substituting for t, we require

$$\begin{array}{ll} (R-2)\mathrm{L} & \lfloor\frac{(\lfloor\frac{x-2}{2}\rfloor)-2}{2}\rfloor \quad \text{to } \lceil\frac{(\lceil\frac{x+2}{2}\rceil)+2}{2}\rceil \\ (R-2)\mathrm{H} & \lfloor\frac{(\lfloor\frac{x-2}{2}\rfloor)-2}{2}\rfloor - 1 \text{ to } \lceil\frac{(\lceil\frac{x+2}{2}\rceil)+2}{2}\rceil \end{array}$$

to reconstruct the necessary coefficients in the the $(R-1)$L subband.

If we define $f_f(x) = \lfloor\frac{x-2}{2}\rfloor$ and $f_c(x) = \lceil\frac{x+2}{2}\rceil$ we can express the required coefficients at each resolution in terms of functional powers of f_f and f_c. Thus to reconstruct $R\mathrm{L}(x)$ we require

$$\begin{array}{ll} (R-1)\mathrm{L} & f_f(x) \quad \text{to } f_c(x) \\ (R-1)\mathrm{H} & f_f(x) - 1 \text{ to } f_c(x), \end{array}$$

which, if $R > 1$, in turn requires

$$\begin{array}{ll} (R-2)\mathrm{L} & f_f^2(x) \quad \text{to } f_c^2(x) \\ (R-2)\mathrm{H} & f_f^2(x) - 1 \text{ to } f_c^2(x) \end{array}$$

and so on, until we reach

$$\begin{array}{ll} 0\mathrm{L} & f_f^R(x) \quad \text{to } f_c^R(x) \\ 0\mathrm{H} & f_f^R(x) - 1 \text{ to } f_c^R(x), \end{array}$$

which are available to us.

To obtain a closed form expression, we show that

$$\begin{aligned} f_f^n(x) &= f_f(f_f^{n-1}(x)) \\ &= \lfloor\frac{x-(2^{n+1}-2)}{2^n}\rfloor. \end{aligned}$$

For $n = 1$

$$\begin{aligned} f_f^1(x) &= \lfloor \frac{x - (2^{1+1} - 2)}{2^1} \rfloor \\ &= \lfloor \frac{x - 2}{2} \rfloor \end{aligned}$$

Assume that $\forall n <= k$ for some $k \in \mathbb{N}$ we have

$$f_f^n(x) = \lfloor \frac{x - (2^{n+1} - 2)}{2^n} \rfloor$$

then if $n = k + 1$

$$\begin{aligned} f_f^{k+1}(x) &= f_f(f_f^k(x)) \\ &= \lfloor \frac{f_f^k(x) - 2}{2} \rfloor \\ &= \lfloor \frac{\lfloor \frac{x-(2^{k+1}-2)}{2^k} \rfloor - 2}{2} \rfloor \\ &= \lfloor \frac{\lfloor \frac{x-(2^{k+1}-2)}{2^k} - 2 \rfloor}{2} \rfloor \\ &= \lfloor \frac{\lfloor \frac{x-(2^{k+1}-2)}{2^k} - \frac{2^{k+1}}{2^k} \rfloor}{2} \rfloor \\ &= \lfloor \frac{\lfloor \frac{x-(2^{k+2}-2)}{2^k} \rfloor}{2} \rfloor \\ &= \lfloor \frac{\lfloor \frac{x-(2^{k+2}-2)}{2^k} \rfloor 2^k}{2^{k+1}} \rfloor \\ &= \lfloor \frac{x - (2^{k+2} - 2)}{2^{k+1}} \rfloor \\ &= \lfloor \frac{x - (2^{(k+1)+1} - 2)}{2^{(k+1)}} \rfloor \end{aligned}$$

so, by induction,

$$f_f^n(x) = \lfloor \frac{x - (2^{n+1} - 2)}{2^n} \rfloor \quad \forall n \in \mathbb{N}$$

A similar proof can be used to show that if $f_c(x) = \lceil \frac{x+2}{2} \rceil$ then

$$\begin{aligned} f_c^n(x) &= f_c(f_c^{n-1}(x)) \\ &= \lceil \frac{x + (2^{n+1} - 2)}{2^n} \rceil. \end{aligned}$$

Returning to two dimensions, the full set of subband coefficients required to reconstruct pixel $I(x, y) = R\text{LL}(x, y)$ are

$$0\text{LL} \qquad\qquad (f_f^R(x) - 1, f_f^R(y)) \text{ to } (f_c^R(x), f_c^R(y))$$

at resolution $r = 0$, and

$$\begin{aligned}
&(r-1)\text{HL} \qquad (f_f^{R-(r-1)}(x)-1, f_f^{R-(r-1)}(y) \quad) \text{ to } (f_c^{R-(r-1)}(x), f_c^{R-(r-1)}(y))\\
&(r-1)\text{LH} \qquad (f_f^{R-(r-1)}(x) \quad , f_f^{R-(r-1)}(y)-1) \text{ to } (f_c^{R-(r-1)}(x), f_c^{R-(r-1)}(y))\\
&(r-1)\text{HH} \qquad (f_f^{R-(r-1)}(x)-1, f_f^{R-(r-1)}(y)-1) \text{ to } (f_c^{R-(r-1)}(x), f_c^{R-(r-1)}(y))
\end{aligned}$$

at resolutions $r = 1...R$.

That is, at resolution level r=0

$$\begin{aligned}
\text{LL} \quad &(\lfloor\frac{x-(2^{R+1}-2)}{2^R}\rfloor, \lfloor\frac{y-(2^{R+1}-2)}{2^R}\rfloor)\\
&\text{to } (\lceil\frac{x+(2^{R+1}-2)}{2^R}\rceil, \lceil\frac{y+(2^{R+1}-2)}{2^R}\rceil)
\end{aligned}$$

and at resolution levels $r = 1...R$

$$\begin{aligned}
\text{HL} \quad &(\lfloor\frac{x-(2^{R-r+2}-2)}{2^{R-r+1}}\rfloor - 1, \lfloor\frac{y-(2^{R-r+2}-2)}{2^{R-r+1}}\rfloor)\\
&\text{to } (\lceil\frac{x+(2^{R-r+2}-2)}{2^{R-r+1}}\rceil, \lceil\frac{y+(2^{R-r+2}-2)}{2^{R-r+1}}\rceil),\\
\text{LH} \quad &(\lfloor\frac{x-(2^{R-r+2}-2)}{2^{R-r+1}}\rfloor \quad , \lfloor\frac{y-(2^{R-r+2}-2)}{2^{R-r+1}}\rfloor - 1)\\
&\text{to } (\lceil\frac{x+(2^{R-r+2}-2)}{2^{R-r+1}}\rceil, \lceil\frac{y+(2^{R-r+2}-2)}{2^{R-r+1}}\rceil),\\
\text{HH} \quad &(\lfloor\frac{x-(2^{R-r+2}-2)}{2^{R-r+1}}\rfloor - 1, \lfloor\frac{y-(2^{R-r+2}-2)}{2^{R-r+1}}\rfloor - 1)\\
&\text{to } (\lceil\frac{x+(2^{R-r+2}-2)}{2^{R-r+1}}\rceil, \lceil\frac{y+(2^{R-r+2}-2)}{2^{R-r+1}}\rceil).
\end{aligned}$$

D.5.4 Extension to multiple pixels

For any set of pixels, the union of the sets of required coefficients for the individual pixels will form the required coefficients for the set of pixels. However, because the wavelet transform preserves spatial relationships, when pixels are adjacent there is substantial overlap between the sets of required coefficients.

More specifically, for two horizontally adjacent pixels, the leftmost coefficient required from each subband is defined by the position of the leftmost pixel, the rightmost coefficient required from each subband is defined by the position of the rightmost pixel and the intervening coefficients will overlap[5]. Similarly for vertically adjacent pixels, with the top-

[5]This can be observed for a single dimension and single decomposition level in figure D.144, by examining the required coefficients in the L and H subbands for adjacent pixels $Y(2m)$ and $Y(2m+1)$ and adjacent pixels $Y(2m+1)$ and $Y(2m+2)$

and bottommost coefficients in each subband defined by the top- and bottommost most pixels, respectively, and the intervening coefficients overlapping.

As a result, for a rectangular region of pixels, the required coefficients in each subband will also be a rectangular region with the top left coefficient defined according to the top left pixel and the bottom right coefficient defined according to the bottom right coefficient. Thus to reconstruct a rectangular region of pixels from (x_1, y_1) to (x_2, y_2) we require, at resolution level r=0,

$$\text{LL} \quad (\lfloor \frac{x_1 - (2^{R+1} - 2)}{2^R} \rfloor, \lfloor \frac{y_1 - (2^{R+1} - 2)}{2^R} \rfloor) \text{ to } (\lceil \frac{x_2 + (2^{R+1} - 2)}{2^R} \rceil, \lceil \frac{y_2 + (2^{R+1} - 2)}{2^R} \rceil)$$

and, at resolution levels $r = 1...R$,

$$\text{HL} \quad (\lfloor \frac{x_1 - (2^{R-r+2} - 2)}{2^{R-r+1}} \rfloor - 1, \lfloor \frac{y_1 - (2^{R-r+2} - 2)}{2^{R-r+1}} \rfloor) \text{ to } (\lceil \frac{x_2 + (2^{R-r+2} - 2)}{2^{R-r+1}} \rceil, \lceil \frac{y_2 + (2^{R-r+2} - 2)}{2^{R-r+1}} \rceil),$$

$$\text{LH} \quad (\lfloor \frac{x_1 - (2^{R-r+2} - 2)}{2^{R-r+1}} \rfloor, \lfloor \frac{y_1 - (2^{R-r+2} - 2)}{2^{R-r+1}} \rfloor - 1) \text{ to } (\lceil \frac{x_2 + (2^{R-r+2} - 2)}{2^{R-r+1}} \rceil, \lceil \frac{y_2 + (2^{R-r+2} - 2)}{2^{R-r+1}} \rceil),$$

$$\text{HH} \quad (\lfloor \frac{x_1 - (2^{R-r+2} - 2)}{2^{R-r+1}} \rfloor - 1, \lfloor \frac{y_1 - (2^{R-r+2} - 2)}{2^{R-r+1}} \rfloor - 1) \text{ to } (\lceil \frac{x_2 + (2^{R-r+2} - 2)}{2^{R-r+1}} \rceil, \lceil \frac{y_2 + (2^{R-r+2} - 2)}{2^{R-r+1}} \rceil).$$

In cases where the required coefficients are outside the subband boundaries, periodic symmetric extension is used to generate values for those coefficients, thus the required coefficients will be the intersection of the above coordinates and the subband boundaries.

Appendix E

A Blind Scalable Watermark for JPEG2000: with Improved Security

E.1 Proofs of Design Features for the Secured Algorithm

E.1.1 Proof Watermark Element Generation

We show that the watermark element generation function defined in section 6.1.3 and the corresponding step size exponent satisfy the requirement, for the proof in section D.3.1, that

1. given the original image, where $v_i \in V$ is a selected coefficient, embedding will not disturb the most significant bit of v_i

$$\exists j_i \in \mathbb{N} \text{ s.t. } 0 \leq u_i < 2^{j_i} \leq |v_i|. \tag{5.5}$$

   and ensure that, given the correct parameters $\Lambda^* = \Lambda$ and a potentially scaled but otherwise untampered watermarked image, of which at least the Φ most significant bit planes of the lowest resolution layer have not been lost, the number of watermark bits can be correctly determined for any coefficient

2. for the unscaled watermarked image, where $v'_i \in I'$ is the coefficient corresponding to $v_i \in I$

$$j'_i = j_i \tag{6.4a}$$

3. for the resolution scaled watermarked image, where $v_i^{\mathcal{R}} \in I^{\mathcal{R}}$ is the coefficient corresponding to $v'_i \in I'$ (and has not been completely lost $v_i^{\mathcal{R}} \in I^{\mathcal{R}}$)

$$j_i^{\mathcal{R}} = j_i \tag{6.4b}$$

4. for the quality scaled watermarked image, where $v_i^{\mathcal{Q}} \in I^{\mathcal{Q}}$ is the coefficient corresponding to $v'_i \in I'$ and has not been completely lost $v^{\mathcal{Q}} \neq 0$

$$j_i^{\mathcal{Q}} = j_i. \tag{6.4c}$$

Let our watermark element in an image I^* be defined as in section 6.1.3

$$u^*_{i^*} = \sum_{\kappa^*=0}^{j^*_{i^*}-1} u^*_{i^*,\kappa^*} 2^{\kappa^*}. \tag{6.1}$$

where each watermark bit $u^*_{i^*,\kappa^*} \in \{0,1\}$

and the number of bits $j^*_{i^*}$ is either calculated depending on whether or not the coefficient $v^*_{i^*}$ is in the lowest resolution layer

$$j^*_{i^*} = \begin{cases} \log_2\left(\bar{\mathsf{G}}(v^*_{i^*}, i^*, \Lambda^*, I^*)\right) & \text{for } i^* \geq C^*X^*[0]Y^*[0] \\ \min\left(\log_2\left(\bar{\mathsf{G}}(v^*_{i^*}, i^*, \Lambda^*, I^*)\right), \log_2\left(\max_{x=0}^{C^*X^*[0]Y^*[0]-1} \bar{v}^*_x\right) - \Phi^*\right) & \\ \text{otherwise} & \end{cases} \tag{6.2}$$

whenever $v^*_{i^*} \in V^*$ is a selected coefficient, or is set to zero

$$j^*_{i^*} = 0 \text{ if } v^*_{i^*} \notin V^*. \tag{6.3}$$

The parameter $\Phi^* \in \Lambda^*$ specifies the number of most significant lowest resolution bit planes that will not be disturbed by embedding, and the calculations of $j^*_{i^*}$ are based on the value

$$\mathsf{G}(v^*_{i^*}, i^*, \Lambda^*, I^*) = \lfloor \alpha^* 2^{-(h+1)} \bar{v}^*_{i^*} w^*_{i^*} \rfloor, \tag{5.18}$$

where $w^*_{i^*} = \mathsf{g}(sk^*, i^*)$ (5.15), is in the range $0 \leq w^*_i < 2^h$ for some $h \in \mathbb{N}$, is pseudo-randomly generated using the key $sk^* \in \Lambda^*$ and the index i^*, where $\bar{v}^*_{i^*}$ is the smallest non-negative-integer power of two that exceeds $v^*_{i^*}$ (5.16) and where $\alpha^* \in \Lambda^*$ is a global strength parameter in the range $0 \leq \alpha^* < 1$.

1. Original

If I is the original image and $v_i \in V$ is a selected coefficient then

$$u_i = \sum_{\kappa=0}^{j_i-1} u_{i,\kappa} 2^{\kappa} \tag{6.1}$$

and $u_{i,\kappa} \in \{0,1\}$ so

$$\sum_{\kappa=0}^{j_i-1} 0 \leq u_i \leq \sum_{\kappa=0}^{j_i-1} 2^{\kappa}$$

$$0 \leq u_i \leq 2^{j_i} - 1$$

$$0 \leq u_i < 2^{j_i}.$$

Because $v_i \in V$, the quantization step size exponent is calculated by

$$j_i = \begin{cases} \log_2\left(\bar{\mathsf{G}}(v_i, i, \Lambda, I)\right) & \text{for } i \geq CX[0]Y[0] \\ \min\left(\log_2\left(\bar{\mathsf{G}}(v_i, i, \Lambda, I)\right), \log_2\left(\max_{x=0}^{CX[0]Y[0]-1} \bar{v}_x\right) - \Phi\right) & \\ & \text{otherwise} \end{cases} \tag{6.2}$$

$$\leq \log_2\left(\bar{\mathsf{G}}(v_i, i, \Lambda, I)\right)$$

so

$$2^{j_i} \leq \bar{\mathsf{G}}(v_i, i, \Lambda, I).$$

If we let $\bar{\mathsf{G}}(v_i, i, \Lambda, I) = 2^k$ for some $k \in \mathbb{N}$ then

$$2^{j_i} \leq 2^k$$
$$j_i \leq k$$

but also

$$\lfloor 2^{k-1} \rfloor \leq |\mathsf{G}(v_i, i, \Lambda, I)| < 2^k \tag{5.16}$$

thus

$$\lfloor 2^{j-1} \rfloor \leq |\mathsf{G}(v_i, i, \Lambda, I)|.$$

Now, from the definition of G (5.18)

$$|\mathsf{G}(v_i, i, \Lambda, I)| = |\lfloor \alpha 2^{-(h+1)} \bar{v}_i w_i \rfloor| \qquad \text{where } 0 \leq w_i < 2^h, \text{ and } 0 \leq \alpha < 1$$

which, since $\bar{v}_i > 0$ (5.16), has all non-negative components and is thus

$$
\begin{aligned}
&= \lfloor \alpha 2^{-(h+1)} \bar{v}_i w_i \rfloor \\
&\leq \alpha 2^{-(h+1)} \bar{v}_i w_i \\
&< 2^{-(h+1)} \bar{v}_i w_i && 0 \leq \alpha < 1 \\
&< 2^{-1} \bar{v}_i && 0 \leq w_i < 2^h \\
\lfloor 2^{j_i - 1} \rfloor &< 2^{-1} \bar{v}_i && \lfloor 2^{j_i-1} \rfloor \leq |\mathsf{G}(v_i, i, \Lambda, I)| \\
\left\lfloor \frac{\lfloor 2^{j_i-1} \rfloor 2}{1} \right\rfloor &< \bar{v}_i \\
\left\lfloor \frac{2^{j_i}}{1} \right\rfloor &< \bar{v}_i && \text{Lemma D.2.2} \\
\lfloor 2^{j_i} \rfloor &< \bar{v}_i \\
2^{j_i} &< \bar{v}_i && j_i \in \mathbb{N} \implies 2^{j_i} \in \mathbb{N} \\
2^{j_i} &< 2^{k_i} && \text{let } \bar{v}_i = 2^{k_i} \text{ for some } k_i \in \mathbb{N} \\
2^{j_i} &\leq 2^{k_i - 1} && j_i \in \mathbb{N}, k_i \in \mathbb{N} \\
2^{j_i} &\leq \lfloor 2^{k_i - 1} \rfloor && v_i \in V \implies v_i \geq 1 \ \text{(5.2b)} \implies k_i \geq 1 \\
2^{j_i} &\leq |v_i| && \lfloor 2^{k_i-1} \rfloor \leq |v_i| \ \text{(5.16)}.
\end{aligned}
$$

Therefore

$$j_i \in \mathbb{N} \text{ s.t. } 0 \leq u_i < 2^{j_i} \leq |v_i|$$

whenever $v_i \in V$, and property (5.5) (which is required for the proof in section D.3.1) is proven.

2. Unscaled

If $I' = \mathsf{Embed}(I, \Lambda)$ is the unscaled watermarked image, $\Lambda' = \Lambda$, and $v'_i \in I'$ is the coefficient corresponding to $v_i \in I$ then v'_i will either be selected or not selected. In either case we show that $j'_i = j_i$.

If $v'_i \in V'$ then, because the definition of G is the same as our previous design (5.18), the properties of G in section 5.1.4.2 (and corresponding proofs in section D.3.4) still hold,

$$\mathsf{G}(v'_i, i, \Lambda', I') = \mathsf{G}(v_i, i, \Lambda, I) \tag{5.19a}$$

Let $v'_x \in I$, if $v'_x \in V'$ then, from proof of (5.19a) in section D.3.4,

$$\bar{v}'_x = \bar{v}_x;$$

alternatively, if $v'_x \notin V'$ then $v_x \notin V$ (5.4a) and no watermark is embedded so

$$\bar{v}'_x = \bar{v}_x.$$

Because watermark embedding leaves the number of components and image dimensions unchanged, $C' = C$, $X'[0] = X[0]$ and $Y'[0] = Y[0]$, thus

$$\begin{aligned}
\log_2\left(\max_{x=0}^{C'X'[0]Y'[0]-1} \bar{v}'_x\right) - \Phi' &= \log_2\left(\max_{x=0}^{CX[0]Y[0]-1} \bar{v}'_x\right) - \Phi' \\
&= \log_2\left(\max_{x=0}^{CX[0]Y[0]-1} \bar{v}_x\right) - \Phi' && \bar{v}'_x = \bar{v}_x \\
&= \log_2\left(\max_{x=0}^{CX[0]Y[0]-1} \bar{v}_x\right) - \Phi && \Lambda' = \Lambda
\end{aligned}$$

Substituting for $\mathsf{G}(v'_i, i, \Lambda', I')$ and $\log_2\left(\max_{x=0}^{C'X'[0]Y'[0]-1} \bar{v}'_x\right) - \Phi'$ in the definition of $j^*_{i^*}$ when $v^*_{i^*} \in V^*$, we obtain

$$\begin{aligned}
j'_i &= \begin{cases} \log_2\left(\bar{\mathsf{G}}(v'_i, i, \Lambda', I)\right) & \text{for } i \geq C'X'[0]Y'[0] \\ \min\left(\log_2\left(\bar{\mathsf{G}}(v'_i, i, \Lambda', I')\right), \log_2\left(\max_{x=0}^{C'X'[0]Y'[0]-1} \bar{v}'_x\right) - \Phi\right) & \\ & \text{otherwise} \end{cases} \qquad (6.2) \\
&= \begin{cases} \log_2\left(\bar{\mathsf{G}}(v_i, i, \Lambda, I)\right) & \text{for } i \geq CX[0]Y[0] \\ \min\left(\log_2\left(\bar{\mathsf{G}}(v_i, i, \Lambda, I)\right), \log_2\left(\max_{x=0}^{CX[0]Y[0]-1} \bar{v}_x\right) - \Phi\right) & \\ & \text{otherwise} \end{cases} \\
&= j_i
\end{aligned}$$

If, $v'_i \notin V'$ then

$$v'_i \notin V' \wedge v_i \notin V \qquad (5.4a)$$

so, from the definition of $j^*_{i^*}$ when $v^*_{i^*} \notin V^*$,

$$\begin{aligned}
j'_i &= 0 \wedge j_i = 0 \qquad (6.3) \\
j'_i &= j_i
\end{aligned}$$

Thus $\forall v'_i \in I'$, $j'_i = j_i$ and property (6.4a) is proven.

3. Resolution Scaled

If $I^{\mathcal{R}} = \mathcal{R}(I')$ resolution scaled image, $\Lambda^{\mathcal{R}} = \Lambda$ and $v^{\mathcal{R}} \in I^{\mathcal{R}}$ is the corresponding coefficient to $v'_i \in I'$ then $v^{\mathcal{R}}_i$ will either be selected or not selected. In either case we show that $j^{\mathcal{R}}_i = j_i$.

If $v^{\mathcal{R}} \in V^{\mathcal{R}}$ then, because the definition of G is the same as our previous design (5.18), the properties of G in section 5.1.4.2 (and corresponding proofs in section D.3.4) still hold,

$$\mathsf{G}(v^{\mathcal{R}}_i, i, \Lambda^{\mathcal{R}}, I^{\mathcal{R}}) = \mathsf{G}(v_i, i, \Lambda, I) \qquad (5.19b)$$

For all coefficients $v_x^{\mathcal{R}} = I^{\mathcal{R}}(c^{\mathcal{R}}, r^{\mathcal{R}}, o^{\mathcal{R}}, x^{\mathcal{R}}, y^{\mathcal{R}})$ where $x \in \mathbb{Z}$, $0 \le x \le C^{\mathcal{R}}X^{\mathcal{R}}[0]Y^{\mathcal{R}}[0]-1$, from the indexing formula we have

$$\begin{aligned} x = {} & c^{\mathcal{R}}X^{\mathcal{R}}[r^{\mathcal{R}}]Y^{\mathcal{R}}[r^{\mathcal{R}}] + (C^{\mathcal{R}} - c^{\mathcal{R}})X^{\mathcal{R}}[r^{\mathcal{R}}-1]Y^{\mathcal{R}}[r^{\mathcal{R}}-1] \\ & + \sum_{a=0}^{a=o^{\mathcal{R}}-1} X^{\mathcal{R}}[r^{\mathcal{R}}, a]Y^{\mathcal{R}}[r^{\mathcal{R}}, a] + y^{\mathcal{R}}X^{\mathcal{R}}[r^{\mathcal{R}}, o^{\mathcal{R}}] + x^{\mathcal{R}} \end{aligned} \tag{5.12}$$

Now if $v_x^{\mathcal{R}}$ does not belong to the lowest resolution layer, $r^{\mathcal{R}} \ge 1$ and

$$\begin{aligned} x \ge {} & c^{\mathcal{R}}X^{\mathcal{R}}[1]Y^{\mathcal{R}}[1] + (C^{\mathcal{R}} - c^{\mathcal{R}})X^{\mathcal{R}}[0]Y^{\mathcal{R}}[0] \\ & + \sum_{a=0}^{a=o^{\mathcal{R}}-1} X^{\mathcal{R}}[1, a]Y^{\mathcal{R}}[1, a] + y^{\mathcal{R}}X^{\mathcal{R}}[1, o^{\mathcal{R}}] + x^{\mathcal{R}} \end{aligned}$$

which, since $X^{\mathcal{R}}[1]Y^{\mathcal{R}}[1] \ge X^{\mathcal{R}}[0]Y^{\mathcal{R}}[0]$ (2.14),

$$\begin{aligned} & \ge c^{\mathcal{R}}X^{\mathcal{R}}[0]Y^{\mathcal{R}}[0] + (C^{\mathcal{R}} - c^{\mathcal{R}})X^{\mathcal{R}}[0]Y^{\mathcal{R}}[0] \\ & \qquad + \sum_{a=0}^{a=o^{\mathcal{R}}-1} X^{\mathcal{R}}[1, a]Y^{\mathcal{R}}[1, a] + y^{\mathcal{R}}X^{\mathcal{R}}[1, o^{\mathcal{R}}] + x^{\mathcal{R}} \\ & \ge C^{\mathcal{R}}X^{\mathcal{R}}[0]Y^{\mathcal{R}}[0] \end{aligned}$$

which is a contradiction, since $x \le C^{\mathcal{R}}X^{\mathcal{R}}[0]Y^{\mathcal{R}}[0] - 1$, therefore $r^{\mathcal{R}} = 0$. Therefore $v_x^{\mathcal{R}}$ lies in the lowest resolution layer, which is unaffected by resolution scaling, so

$$\begin{aligned} v_x^{\mathcal{R}} &= v'_x \\ \bar{v}_x^{\mathcal{R}} &= \bar{v}'_x \\ &= \bar{v}_x \qquad \text{(from the } \textbf{2. Unscaled} \text{ proof).} \end{aligned}$$

Neither the number of components nor the lowest resolution layer is affected by resolution scaling so $C^{\mathcal{R}} = C' = C$, $X^{\mathcal{R}}[0] = X'[0] = X[0]$ and $Y^{\mathcal{R}}[0] = Y'[0] = Y[0]$, thus

$$\begin{aligned} \log_2\left(\max_{x=0}^{C^{\mathcal{R}}X^{\mathcal{R}}[0]Y^{\mathcal{R}}[0]-1} \bar{v}_x^{\mathcal{R}}\right) - \Phi^{\mathcal{R}} &= \log_2\left(\max_{x=0}^{CX[0]Y[0]-1} \bar{v}_x^{\mathcal{R}}\right) - \Phi^{\mathcal{R}} \\ &= \log_2\left(\max_{x=0}^{CX[0]Y[0]-1} \bar{v}_x\right) - \Phi^{\mathcal{R}} \qquad \bar{v}_x^{\mathcal{R}} = \bar{v}_x \\ &= \log_2\left(\max_{x=0}^{CX[0]Y[0]-1} \bar{v}_x\right) - \Phi \qquad \Lambda^{\mathcal{R}} = \Lambda. \end{aligned}$$

Substituting for $\mathsf{G}(v_i^{\mathcal{R}}, i, \Lambda^{\mathcal{R}}, I^{\mathcal{R}})$ and $\log_2\left(\max_{x=0}^{C^{\mathcal{R}}X^{\mathcal{R}}[0]Y^{\mathcal{R}}[0]-1} \bar{v}_x^{\mathcal{R}}\right) - \Phi^{\mathcal{R}}$ in the definition of $j_{i^*}^*$ when $v_{i^*}^* \in V^*$, we obtain

$$
\begin{aligned}
j_i^{\mathcal{R}} &= \begin{cases} \log_2\left(\bar{\mathsf{G}}(v_i^{\mathcal{R}}, i, \Lambda^{\mathcal{R}}, I^{\mathcal{R}})\right) & \text{for } i \geq C^{\mathcal{R}}X^{\mathcal{R}}[0]Y^{\mathcal{R}}[0] \\ \min\left(\log_2\left(\bar{\mathsf{G}}(v_i^{\mathcal{R}}, i, \Lambda^{\mathcal{R}}, I^{\mathcal{R}})\right), \log_2\left(\max_{x=0}^{C^{\mathcal{R}}X^{\mathcal{R}}[0]Y^{\mathcal{R}}[0]-1} \bar{v}_x^{\mathcal{R}}\right) - \Phi\right) & \\ \qquad \text{otherwise} \end{cases} \\
&= \begin{cases} \log_2\left(\bar{\mathsf{G}}(v_i, i, \Lambda, I)\right) & \text{for } i \geq CX[0]Y[0] \\ \min\left(\log_2\left(\bar{\mathsf{G}}(v_i, i, \Lambda, I)\right), \log_2\left(\max_{x=0}^{CX[0]Y[0]-1} \bar{v}_x\right) - \Phi\right) & \\ \qquad \text{otherwise} \end{cases} \\
&= j_i
\end{aligned}
$$

If, $v_i^{\mathcal{R}} \notin V^{\mathcal{R}}$ then

$$v_i^{\mathcal{R}} \notin V^{\mathcal{R}} \wedge v_i' \notin V' \tag{5.4b}$$
$$v_i^{\mathcal{R}} \notin V^{\mathcal{R}} \wedge v_i \notin V \tag{5.4a}$$

so, from the definition of $j_{i^*}^*$ when $v_{i^*}^* \notin V^*$,

$$
\begin{aligned}
&j_i^{\mathcal{R}} = 0 \text{ and } j_i = 0 \qquad (6.3) \\
&j_i^{\mathcal{R}} = j_i
\end{aligned}
$$

Thus $\forall v_i^{\mathcal{R}} \in I^{\mathcal{R}}$, $j_i^{\mathcal{R}} = j_i$ and property (6.4b) is proven.

4. Quality Scaled

If $I^{\mathcal{Q}} = \mathcal{Q}(I')$ is the quality scaled image and $\Lambda^{\mathcal{Q}} = \Lambda$, then any coefficient $v_i^{\mathcal{Q}} \in I^{\mathcal{Q}}$ that has not been completely lost due to quality scaling $v_i^{\mathcal{Q}} \neq 0$, will either be selected or not selected. In either case we show that $j_i^{\mathcal{Q}} = j_i$.

If $v_i^{\mathcal{Q}} \in V^{\mathcal{Q}}$ then, because the definition of G is the same as our previous design, so the properties of G in section 5.1.4.2 (and corresponding proofs in section D.3.4) still hold,

$$\mathsf{G}(v_{i^{\mathcal{Q}}}^{\mathcal{Q}}, i^{\mathcal{Q}}, \Lambda^{\mathcal{Q}}, I^{\mathcal{Q}}) = \mathsf{G}(v_i, i, \Lambda, I) \tag{5.19b}$$

If $v_i^{\mathcal{Q}} \in V^{\mathcal{Q}}$ and $i < C^{\mathcal{Q}}X^{\mathcal{Q}}[0]Y^{\mathcal{Q}}[0]$ then since $v_i^{\mathcal{Q}} \in V^{\mathcal{Q}} \iff |v_i^{\mathcal{Q}}| \geq t = 2^n$ (5.2b) it must be the case that $\forall v_x^{\mathcal{Q}} \notin V^{\mathcal{Q}}$,

$$
\begin{aligned}
|v_x^{\mathcal{Q}}| &< |v_i^{\mathcal{Q}}| \\
\bar{v}_x^{\mathcal{Q}} &< \bar{v}_i^{\mathcal{Q}}
\end{aligned}
$$

so

$$\max_{x=0}^{C^{\mathcal{Q}}X^{\mathcal{Q}}[0]Y^{\mathcal{Q}}[0]-1} \bar{v}_x^{\mathcal{Q}} = \max_{\substack{x=0 \\ v_x^{\mathcal{Q}} \in V^{\mathcal{Q}}}}^{C^{\mathcal{Q}}X^{\mathcal{Q}}[0]Y^{\mathcal{Q}}[0]-1} \bar{v}_x^{\mathcal{Q}}$$

For coefficients $v_x^{\mathcal{Q}} \in V^{\mathcal{Q}}$ from the proof of (5.19c) in section D.3.4, we have

$$\begin{aligned} \bar{v}_x^{\mathcal{Q}} &= \bar{v}'_x \\ &= \bar{v}_x \end{aligned}$$

and, because quality scaling leaves the image dimensions unchanged, we have $C^{\mathcal{Q}} = C' = C$, $X^{\mathcal{Q}}[0] = X'[0] = X[0]$ and $Y^{\mathcal{Q}}[0] = Y'[0] = Y[0]$. Thus

$$\begin{aligned} \log_2\left(\max_{x=0}^{C^{\mathcal{Q}}X^{\mathcal{Q}}[0]Y^{\mathcal{Q}}[0]-1} \bar{v}_x^{\mathcal{Q}}\right) - \Phi^{\mathcal{Q}} &= \log_2\left(\max_{x=0}^{CX[0]Y[0]-1} \bar{v}_x^{\mathcal{Q}}\right) - \Phi^{\mathcal{Q}} \\ &= \log_2\left(\max_{\substack{x=0 \\ v_x^{\mathcal{Q}} \in V^{\mathcal{Q}}}}^{C^{\mathcal{Q}}X^{\mathcal{Q}}[0]Y^{\mathcal{Q}}[0]-1} \bar{v}_x^{\mathcal{Q}}\right) - \Phi^{\mathcal{Q}} \\ &= \log_2\left(\max_{x=0}^{CX[0]Y[0]-1} \bar{v}_x\right) - \Phi^{\mathcal{Q}} \\ &= \log_2\left(\max_{x=0}^{CX[0]Y[0]-1} \bar{v}_x\right) - \Phi \qquad\qquad \Lambda^{\mathcal{Q}} = \Lambda \end{aligned}$$

provided there exists at least one selected coefficient $v_x^{\mathcal{Q}} \in V^{\mathcal{Q}}$ in the lowest resolution layer $0 \le x < C^{\mathcal{Q}}X^{\mathcal{Q}}[0]Y^{\mathcal{Q}}[0]$.

So if $v_i^{\mathcal{Q}} \in V^{\mathcal{Q}}$, then from the definition of $j_i^{\mathcal{Q}}$

$$j_i^{\mathcal{Q}} = \begin{cases} \log_2\left(\bar{\mathsf{G}}(v_{i^{\mathcal{Q}}}^{\mathcal{Q}}, i^{\mathcal{Q}}, \Lambda^{\mathcal{Q}}, I^{\mathcal{Q}})\right) & \text{for } i^{\mathcal{Q}} \ge C^{\mathcal{Q}}X^{\mathcal{Q}}[0]Y^{\mathcal{Q}}[0] \\ \min\left(\log_2\left(\bar{\mathsf{G}}(v_{i^{\mathcal{Q}}}^{\mathcal{Q}}, i^{\mathcal{Q}}, \Lambda^{\mathcal{Q}}, I^{\mathcal{Q}})\right), \log_2\left(\max_{x=0}^{C^{\mathcal{Q}}X^{\mathcal{Q}}[0]Y^{\mathcal{Q}}[0]-1} \bar{v}_x^{\mathcal{Q}}\right) - \Phi^{\mathcal{Q}}\right) & \\ & \text{otherwise} \end{cases}$$

substituting for $\bar{\mathsf{G}}(v_{i^{\mathcal{Q}}}^{\mathcal{Q}}, i^{\mathcal{Q}}, \Lambda^{\mathcal{Q}}, I^{\mathcal{Q}})$,

$$= \begin{cases} \log_2\left(\bar{\mathsf{G}}(v_i, i, \Lambda, I)\right) & \text{for } i \ge C^{\mathcal{Q}}X^{\mathcal{Q}}[0]Y^{\mathcal{Q}}[0] \\ \min\left(\log_2\left(\bar{\mathsf{G}}(v_i, i, \Lambda, I)\right), \log_2\left(\max_{x=0}^{C^{\mathcal{Q}}X^{\mathcal{Q}}[0]Y^{\mathcal{Q}}[0]-1} \bar{v}_x^{\mathcal{Q}}\right) - \Phi^{\mathcal{Q}}\right) & \text{otherwise} \end{cases}$$

and for $\log_2\left(\max_{x=0}^{C^{\mathcal{Q}}X^{\mathcal{Q}}[0]Y^{\mathcal{Q}}[0]-1} \bar{v}_x^{\mathcal{Q}}\right) - \Phi^{\mathcal{Q}}$ when $i < C^{\mathcal{Q}}X^{\mathcal{Q}}[0]Y^{\mathcal{Q}}[0]$

$$= \begin{cases} \log_2\left(\bar{\mathsf{G}}(v_i, i, \Lambda, I)\right) & \text{for } i \ge C^{\mathcal{Q}}X^{\mathcal{Q}}[0]Y^{\mathcal{Q}}[0] \\ \min\left(\log_2\left(\bar{\mathsf{G}}(v_i, i, \Lambda, I)\right), \log_2\left(\max_{x=0}^{CX[0]Y[0]-1} \bar{v}_x\right) - \Phi\right) & \text{otherwise} \end{cases}$$

and finally for $C^{\mathcal{Q}}$, $X^{\mathcal{Q}}[0]$ and $Y^{\mathcal{Q}}[0]$

$$= \begin{cases} \log_2\left(\bar{\mathsf{G}}(v_i, i, \Lambda, I)\right) & \text{for } i \geq CX[0]Y[0] \\ \min\left(\log_2\left(\bar{\mathsf{G}}(v_i, i, \Lambda, I)\right), \log_2\left(\max_{x=0}^{CX[0]Y[0]-1} \bar{v}_x\right) - \Phi\right) & \text{otherwise} \end{cases}$$

$$= j_i$$

If the coefficient has not been selected $v_i^{\mathcal{Q}} \notin V^{\mathcal{Q}}$, then

$$v_i^{\mathcal{Q}} \notin V^{\mathcal{Q}} \wedge v_i' \notin V' \tag{5.4c}$$

$$v_i^{\mathcal{Q}} \notin V^{\mathcal{Q}} \wedge v_i \notin V \tag{5.4a}$$

so, from the definition of $j_{i^*}^*$ when $v_{i^*}^* \notin V^*$,

$$j_i^{\mathcal{Q}} = 0 \text{ and } j_i = 0 \tag{6.3}$$

$$j_i^{\mathcal{Q}} = j_i$$

Thus $\forall v_i^{\mathcal{Q}} \neq 0 \in I^{\mathcal{Q}}$, $j_i^{\mathcal{Q}} = j_i$ and property (6.4c) is proven.

E.1.2 Proof for Feature Sequence Formation

We show that the feature formation procedure described in section 6.1.3.1 satisfies the properties defined in that section. That is, given the correct watermarking parameters and a potentially scaled but otherwise untampered watermarked image, of which at least the Φ most significant bit planes of the lowest resolution layer have not been lost, corresponding feature sets are composed of corresponding coefficients.

- for the unscaled watermarked image, where $v_i' \in V'$ is the coefficient corresponding to $v_i \in V$

$$\mathfrak{f}'(i,\, \kappa, x) = \mathfrak{f}(i,\, \kappa, x) \tag{6.11a}$$

$$v'_{\mathfrak{f}'(i,\,\kappa,x)} \in V'_{i,\,\kappa} \iff v_{\mathfrak{f}(i,\,\kappa,x)} \in V_{i,\,\kappa} \tag{6.11b}$$

- for the resolution scaled watermarked image, where $v_i^{\mathcal{R}} \in V^{\mathcal{R}}$ is the coefficient corresponding to $v_i' \in V'$

$$\mathfrak{f}^{\mathcal{R}}(i,\, \kappa, x) = \mathfrak{f}(i,\, \kappa, x) \tag{6.11c}$$

$$v^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\,\kappa,x)} \in V^{\mathcal{R}}_{i,\,\kappa} \iff v_{\mathfrak{f}(i,\,\kappa,x)} \in V_{i,\,\kappa} \tag{6.11d}$$

- for the quality scaled watermarked image, where $v_i^{\mathcal{Q}} \in V^{\mathcal{Q}}$ is the coefficient corresponding to $v_i' \in V'$

$$\mathfrak{f}^{\mathcal{Q}}(i,\, \kappa, x) = \mathfrak{f}(i,\, \kappa, x) \tag{6.11e}$$

$$v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\,\kappa,x)} \in V^{\mathcal{Q}}_{i,\,\kappa} \iff v_{\mathfrak{f}(i,\,\kappa,x)} \in V_{i,\,\kappa}. \tag{6.11f}$$

Let our feature sequence be defined as in section 6.1.3.1

$$V^*_{i,\kappa}(i,\kappa,\Lambda^*,I^*) = \{v^*_{\mathfrak{f}^*(i,\kappa,0)}, v^*_{\mathfrak{f}^*(i,\kappa,1)}, \ldots, v^*_{\mathfrak{f}^*(i,\kappa,\eta^*-1)}\} \qquad \kappa \in \mathbb{Z}, 0 \leq \kappa < j^*_i \quad (6.5)$$

with

$$\mathfrak{f}^*(i,\ \kappa, x) = \mathfrak{I}^*_{i,\kappa}(\mathfrak{r}^*_{i,\kappa,x}) \quad (6.10)$$

and

$$\mathfrak{r}^*{}_{i,\kappa,x}(i,\kappa,x,\Lambda^*,I^*) = \mathsf{g}(\mathsf{H}(sk^*,\Phi^*,0\mathrm{LL}^*),i,\kappa,x) \quad (6.9)$$

where $\mathsf{H}(sk^*,\Phi^*,0\mathrm{LL}^*)$ is a hash of the secret key sk^* and the first Φ^* significant bit planes of the lowest resolution layer $0\mathrm{LL}^*$ of I^* and their signs (with 0 representing both positively signed coefficients and coefficients which are not significant in the first Φ^* significant bit planes of $0\mathrm{LL}^*$) and where $\mathfrak{I}^*{}_{i,\kappa}(i,\kappa,\Lambda^*,I^*)$ is a monotonically increasing sequence such that

$$\mathfrak{I}^*{}_{i,\kappa}(i,\kappa,\Lambda^*,I^*) = \begin{cases} \left[C^*X^*[r^*_i-1]Y^*[r^*_i-1], C^*X^*[r^*_i]Y^*[r^*_i]\right) & \text{if } \Psi^*_{i,\kappa}=0 \\ \bigcup_{n^*_x,n^*_y} \Big\{ c^*_i X^*[r^*_i]Y^*[r^*_i] & \\ \quad +(c^*_i-c^*_i)X^*[r^*_i-1]Y^*[r^*_i-1] & \\ \quad +\sum_{o=0}^{o=s^*_i-1} X^*[r^*_i,o]Y^*[r^*_i,o] & \\ \quad +(n^*_y-\mathrm{tby0}^*)X[r^*_i,s^*_i] & \\ \quad +n^*_x-\mathrm{tbx0}^* \Big\} & \text{if } \Psi^*_{i,\kappa}=1 \end{cases} \quad (6.7)$$

where $n^*_x, n^*_y \in \mathbb{Z}$ such that

$$\begin{aligned} &\max\left(\left\lfloor\frac{\mathrm{tbx0}^*+x^*_i}{2^{\mathrm{xcb}'^*}}\right\rfloor 2^{\mathrm{xcb}'^*}, \mathrm{tbx0}^*\right) \leq n^*_x < \min\left(\left\lceil\frac{\mathrm{tbx0}^*+x^*_i}{2^{\mathrm{xcb}'^*}}\right\rceil 2^{\mathrm{xcb}'^*}, \mathrm{tbx1}^*\right), \\ &\max\left(\left\lfloor\frac{\mathrm{tby0}^*+y^*_i}{2^{\mathrm{ycb}'^*}}\right\rfloor 2^{\mathrm{ycb}'^*}, \mathrm{tby0}^*\right) \leq n^*_y < \min\left(\left\lceil\frac{\mathrm{tby0}^*+y^*_i}{2^{\mathrm{ycb}'^*}}\right\rceil 2^{\mathrm{ycb}'^*}, \mathrm{tby1}^*\right) \end{aligned} \quad (6.8b)$$

and

$$\Psi^*_{i,\kappa^*}(i,\kappa,\Lambda^*,I^*) = \mathsf{g}(\mathsf{H}(sk^*,\Phi^*,0\mathrm{LL}^*),i)_\kappa. \quad (6.6)$$

1. Unscaled

Let $\mathrm{kmax} = \log_2\left(\max_{x=0}^{CX[0]Y[0]-1} \bar{v}_x\right)$ denote the number of most significant bit planes in LL0. For all coefficients v_i' in the lowest resolution layer,

$$
\begin{aligned}
j_i' &= j_i && \text{(6.4a)}\\
&\leq \begin{cases} \log_2\left(\bar{\mathsf{G}}_i\right) & \text{for } i \geq CX[0]Y[0] \\ \min\left(\log_2\left(\bar{\mathsf{G}}_i\right), \log_2\left(\max_{x=0}^{CX[0]Y[0]-1} \bar{v}_x\right) - \Phi\right) & \text{(6.2) (6.3)} \\ & \text{otherwise} \end{cases}
\end{aligned}
$$

from the proof of (6.4a) we know that for v_i in the lowest resolution layer $i < CX[0]Y[0]$, so this is simply

$$
\begin{aligned}
&= \min\left(\log_2\left(\bar{\mathsf{G}}(v_i, i, \Lambda, I)\right), \log_2\left(\max_{x=0}^{CX[0]Y[0]-1} \bar{v}_x\right) - \Phi\right)\\
&\leq \mathrm{kmax} - \Phi.
\end{aligned}
$$

For any coefficient in v_i' in the lowest resolution layer $0\mathrm{LL}'$ the magnitude and sign computed from the Φ most significant bit planes is obtained by setting the $\mathrm{kmax} - \Phi$ least significant bit planes to zero

$$
\begin{aligned}
\mathrm{sign}(v_i')Q_{2^{\mathrm{kmax}-\Phi}}(v_i') &= \mathrm{sign}(v_i')\left\lfloor \frac{|v_i'|}{2^{\mathrm{kmax}-\Phi}} \right\rfloor 2^{\mathrm{kmax}-\Phi} && \text{(5.3a)}\\
&= \mathrm{sign}(v_i')\left\lfloor \frac{Q_{2^{j_i}}(v_i) + u_i}{2^{\mathrm{kmax}-\Phi}} \right\rfloor 2^{\mathrm{kmax}-\Phi} && \text{(5.3c)}\\
&= \mathrm{sign}(v_i')\left\lfloor \frac{\left\lfloor \frac{|v_i|}{2^{j_i}} \right\rfloor 2^{j_i} + u_i}{2^{\mathrm{kmax}-\Phi}} \right\rfloor 2^{\mathrm{kmax}-\Phi} && \text{(5.3a)}\\
&= \mathrm{sign}(v_i')\left\lfloor \frac{\left\lfloor \frac{|v_i|}{2^{j_i}} \right\rfloor + \frac{u_i}{2^{j_i}}}{2^{\mathrm{kmax}-\Phi-j_i}} \right\rfloor 2^{\mathrm{kmax}-\Phi}
\end{aligned}
$$

but $\left\lfloor \frac{|v_i|}{2^{j_i}} \right\rfloor \in \mathbb{Z}$, and $0 \leq u_i < 2^{j_i}$ so $0 \leq \frac{u_i}{2^{j_i}} < 1$ and $j_i, \text{kmax}, \Phi \in \mathbb{Z}, j_i \leq \text{kmax} - \Phi$ so $2^{\text{kmax}-\Phi-j_i} \in \mathbb{N}$ thus

$$
\begin{aligned}
&= \text{sign}(v_i') \left\lfloor \frac{\left\lfloor \frac{|v_i|}{2^{j_i}} \right\rfloor}{2^{\text{kmax}-\Phi-j_i}} \right\rfloor 2^{\text{kmax}-\Phi} && \text{Lemma D.2.2} \\
&= \text{sign}(v_i') \left\lfloor \frac{\left\lfloor \frac{|v_i|}{2^{j_i}} \right\rfloor 2^{j_i}}{2^{\text{kmax}-\Phi}} \right\rfloor 2^{\text{kmax}-\Phi} \\
&= \text{sign}(v_i') \left\lfloor \frac{|v_i|}{2^{\text{kmax}-\Phi}} \right\rfloor 2^{\text{kmax}-\Phi} && \text{Lemma D.2.3} \\
&= \begin{cases} \text{sign}(v_i') \left\lfloor \frac{|v_i|}{2^{\text{kmax}-\Phi}} \right\rfloor 2^{\text{kmax}-\Phi} & \text{if } |v_i| \geq 2^{\text{kmax}-\Phi} \\ 0 & \text{if } |v_i| < 2^{\text{kmax}-\Phi} \end{cases} \\
&= \begin{cases} \text{sign}(v_i') Q_{2^{\text{kmax}-\Phi}}(v_i) & \text{if } |v_i| \geq 2^{\text{kmax}-\Phi} \\ 0 & \text{if } |v_i| < 2^{\text{kmax}-\Phi} \end{cases} && \text{(5.3a)} \\
&= \begin{cases} \text{sign}(\text{sign}(v_i) Q_{2^{j_i}}(v_i) + u_i) Q_{2^{\text{kmax}-\Phi}}(v_i) & \\ & \text{if } |v_i| \geq 2^{\text{kmax}-\Phi} \\ 0 & \text{if } |v_i| < 2^{\text{kmax}-\Phi} \end{cases} && \text{(5.3c)}
\end{aligned}
$$

but $u_i \geq 0$ and $j_i \leq \text{kmax} - \Phi$, so $Q_{2^{j_i}}(v_i) + u_i > 0$ whenever $|v_i| \geq 2^{\text{kmax}-\Phi}$, thus

$$
\begin{aligned}
&= \begin{cases} \text{sign}(v_i) Q_{2^{\text{kmax}-\Phi}}(v_i) & \text{if } |v_i| \geq 2^{\text{kmax}-\Phi} \\ 0 & \text{if } |v_i| < 2^{\text{kmax}-\Phi} \end{cases} \\
&= \text{sign}(v_i) Q_{2^{\text{kmax}-\Phi}}(v_i),
\end{aligned}
$$

which is equal to the magnitude and sign computed from the Φ most significant bit planes of the corresponding coefficient v_i in the lowest resolution layer 0LL of the original image I.

Thus if $I' = \mathsf{Embed}(I, \Lambda)$ is the unscaled watermarked image, $\Lambda' = \Lambda$, and $v_i' \in V'$ is the coefficient corresponding to $v_i \in V$ then, as $\Lambda' = \Lambda$ $sk' = sk, \Phi' = \Phi$, and $\text{sign}(v_i') Q_{2^{\text{kmax}-\Phi}}(v_i') = \text{sign}(v_i) Q_{2^{\text{kmax}-\Phi}}(v_i) \forall v_i' \in 0\text{LL}'$ the hashes of the Φ (Φ') most significant bit planes of the lowest resolution layers, $LL0$ and $LL0'$, of the original and watermarked images are equal

$$\mathsf{H}(sk', \Phi', 0\text{LL}') = \mathsf{H}(sk, \Phi, 0\text{LL}).$$

Substituting into the definition for $\mathfrak{r}'$ we obtain

$$\begin{aligned}\mathfrak{r}'_{i,\kappa,x}(i,\kappa,x,\Lambda',I') &= \mathsf{g}(\mathsf{H}(sk',\Phi',0\mathrm{LL}'),i,\kappa,x) && (6.9)\\ &= \mathsf{g}(\mathsf{H}(sk,\Phi,0\mathrm{LL}),i,\kappa,x)\\ &= \mathfrak{r}_{i,\kappa,x}(i,\kappa,x,\Lambda,I) && (6.9)\end{aligned}$$

and, similarly, for Ψ'

$$\begin{aligned}\Psi'_{i,\kappa}(i,\kappa,\Lambda',I') &= \mathsf{g}(\mathsf{H}(sk',\Phi',0\mathrm{LL}'),i)_\kappa && (6.6)\\ &= \mathsf{g}(\mathsf{H}(sk,\Phi,0\mathrm{LL}),i)_\kappa\\ &= \Psi_{i,\kappa}(i,\kappa,x,\Lambda,I). && (6.6)\end{aligned}$$

The watermark embedding process causes no change to the number of components, image dimensions or codeblock parameters or sample positions, so $C'=C$ and $X'[r]=X[r]$, $Y'[r]=Y[r]$, $X'[r,s]=X[r,s]$, $Y'[r,s]=Y[r,s],\forall r,s$ and $xcb''=xcb'$, $ycb''=ycb'$ and $tbx0'=tbx0$, $tby0'=tby0$, $tbx1'=tbx1$, $tby1'=tby1$, $x'_i=x_i$ and $y'_i=y_i$. Thus, substituting these and $\Psi'_{i,\kappa}=\Psi_{i,\kappa}$ into the definition for $\mathfrak{I}'_{i,\kappa}$,

$$\mathfrak{I}'_{i,\kappa}(i,\kappa,\Lambda',I') = \begin{cases} \left[C'X'[r'_i-1]Y'[r'_i-1], C'X'[r'_i]Y'[r'_i]\right) & \text{if } \Psi'_{i,\kappa}=0\\ \bigcup_{n'_x,n'_y}\Big\{c'_iX'[r'_i]Y'[r'_i] & \\ \quad +(c'_i-c'_i)X'[r'_i-1]Y'[r'_i-1] & \\ \quad +\sum_{a=0}^{a=o'_i-1}X'[r'_i,a]Y'[r'_i,a] & \\ \quad +(n'_y-\mathrm{tby0}')X[r'_i,o'_i] & \\ \quad +n'_x-\mathrm{tbx0}'\Big\} & \text{if } \Psi'_{i,\kappa}=1 \end{cases} \qquad (6.7)$$

where $n'_x, n'_y \in \mathbb{Z}$ such that

$$\begin{aligned}&\max\left(\left\lfloor\frac{\mathrm{tbx0}'+x'_i}{2^{\mathrm{xcb}''}}\right\rfloor 2^{\mathrm{xcb}''},\mathrm{tbx0}'\right) \le n'_x < \min\left(\left\lceil\frac{\mathrm{tbx0}'+x'_i}{2^{\mathrm{xcb}''}}\right\rceil 2^{\mathrm{xcb}''},\mathrm{tbx1}'\right),\\ &\max\left(\left\lfloor\frac{\mathrm{tby0}'+y'_i}{2^{\mathrm{ycb}''}}\right\rfloor 2^{\mathrm{ycb}''},\mathrm{tby0}'\right) \le n'_y < \min\left(\left\lceil\frac{\mathrm{tby0}'+y'_i}{2^{\mathrm{ycb}''}}\right\rceil 2^{\mathrm{ycb}''},\mathrm{tby1}'\right) \qquad (6.8b)\end{aligned}$$

$$= \begin{cases} \left[CX[r_i - 1]Y[r_i - 1], CX[r_i]Y[r_i] \right) \text{ if } \Psi_{i,\kappa} = 0 \\ \bigcup_{n'_x, n'_y} \Big\{ c_i X[r_i]Y[r_i] \\ \qquad +(c_i - c_i)X[r_i - 1]Y[r_i - 1] \\ \qquad + \sum_{o=0}^{o=s_i-1} X[r_i, o]Y[r_i, o] \\ \qquad +(n'_y - \text{tby0})X[r_i, s_i] \\ \qquad +n'_x - \text{tbx0} \Big\} \qquad \text{if } \Psi_{i,\kappa} = 1 \end{cases}$$

with $n'_x, n'_y \in \mathbb{Z}$ such that

$$\max\left(\left\lfloor \frac{\text{tbx0} + x_i}{2^{\text{xcb}'}} \right\rfloor 2^{\text{xcb}'}, \text{tbx0} \right) \le n_x < \min\left(\left\lceil \frac{\text{tbx0} + x_i}{2^{\text{xcb}'}} \right\rceil 2^{\text{xcb}'}, \text{tbx1} \right),$$
$$\max\left(\left\lfloor \frac{\text{tby0} + y_i}{2^{\text{ycb}'}} \right\rfloor 2^{\text{ycb}'}, \text{tby0} \right) \le n_y < \min\left(\left\lceil \frac{\text{tby0} + y_i}{2^{\text{ycb}'}} \right\rceil 2^{\text{ycb}'}, \text{tby1} \right)$$

$$= \mathfrak{I}_{i,\kappa}(i, \kappa, \Lambda, I), \qquad (6.7)$$

we obtain $\mathfrak{I}'_{i,\kappa} = \mathfrak{I}_{i,\kappa}$. Thus

$$\begin{aligned} \mathfrak{f}'(i, \kappa, x) &= \mathfrak{I}'_{i,\kappa}(\mathfrak{r}'_{i,\kappa,x}) && (6.10) \\ &= \mathfrak{I}'_{i,\kappa}(\mathfrak{r}_{i,\kappa,x}) && \mathfrak{r}'_{i,\kappa,x} = \mathfrak{r}_{i,\kappa,x} \\ &= \mathfrak{I}_{i,\kappa}(\mathfrak{r}_{i,\kappa,x}) && \mathfrak{I}'_{i,\kappa} = \mathfrak{I}_{i,\kappa} \\ &= \mathfrak{f}(i, \kappa, x) && (6.10) \end{aligned}$$

and property (6.11a) is proven.

To prove the remaining property we observe, from definition of the feature sequence (6.5)

$$V'_{i,\kappa}(i, \kappa, \Lambda', I') = \{v'_{\mathfrak{f}'(i,\kappa,0)}, v'_{\mathfrak{f}'(i,\kappa,1)}, \ldots, v'_{\mathfrak{f}'(i,\kappa,\eta'-1)}\} \qquad \kappa \in \mathbb{Z}, 0 \le \kappa < j'_i,$$

that a coefficient $v'_{\mathfrak{f}'(i,\kappa,x)} \in I'$ belongs to the feature sequence $V'_{i,\kappa}$ if and only if its index $\mathfrak{f}'(i, \kappa, x)$ is in the $\mathfrak{r}'_{i,\kappa,x}$th position of the index sequence $\mathfrak{I}'_{i,\kappa}$:

$$v'_{\mathfrak{f}'(i,\kappa,x)} \in V'_{i,\kappa} \iff \mathfrak{f}'(i, \kappa, x) = \mathfrak{I}'_{i,\kappa}(\mathfrak{r}'_{i,\kappa,x}) \quad \kappa, x \in \mathbb{Z}, 0 \le \kappa < j'_i, 0 \le x < \eta'.$$

Substituting $\mathfrak{f}'(i,\,\kappa,x) = \mathfrak{I}'_{i,\,\kappa}(\mathfrak{r}'_{i,\,\kappa,x}) = \mathfrak{I}_{i,\,\kappa}(\mathfrak{r}_{i,\,\kappa,x}) = \mathfrak{f}(i,\,\kappa,x)$,

$$\begin{aligned}
v'_{\mathfrak{f}'(i,\,\kappa,x)} \in V'_{i,\,\kappa} &\iff \mathfrak{f}(i,\,\kappa,x) = \mathfrak{I}_{i,\,\kappa}(\mathfrak{r}_{i,\,\kappa,x}) \quad \kappa,x \in \mathbb{Z}, 0 \le \kappa < j'_i, 0 \le x < \eta' \\
&\iff \mathfrak{f}(i,\,\kappa,x) = \mathfrak{I}_{i,\,\kappa}(\mathfrak{r}_{i,\,\kappa,x}) \quad \kappa,x \in \mathbb{Z}, 0 \le \kappa < j_i, 0 \le x < \eta' \qquad (6.4a) \\
&\iff \mathfrak{f}(i,\,\kappa,x) = \mathfrak{I}_{i,\,\kappa}(\mathfrak{r}_{i,\,\kappa,x}) \quad \kappa,x \in \mathbb{Z}, 0 \le \kappa < j_i, 0 \le x < \eta \qquad \Lambda' = \Lambda \\
&\iff v_{\mathfrak{f}(i,\,\kappa,x)} \in V_{i,\,\kappa}.
\end{aligned}$$

So property (6.11b) is proven.

2. Resolution Scaled

Let $I^{\mathcal{R}} = \mathcal{R}(I')$ be a resolution scaled image, $\Lambda^{\mathcal{R}} = \Lambda$ and $v_i^{\mathcal{R}} \in V^{\mathcal{R}}$ is the coefficient corresponding to $v'_i \in V'$.

Because $I^{\mathcal{R}}$ is only resolution scaled, all coefficients in the lowest resolution layer must have been fully received so $0\mathrm{LL}^{\mathcal{R}} = 0\mathrm{LL}'$ and, because $\Lambda^{\mathcal{R}} = \Lambda$, $sk^{\mathcal{R}} = sk$ and $\Phi^{\mathcal{R}} = \Phi$, thus

$$\begin{aligned}
\mathsf{H}(sk^{\mathcal{R}}, \Phi^{\mathcal{R}}, 0\mathrm{LL}^{\mathcal{R}}) &= \mathsf{H}(sk, \Phi, 0\mathrm{LL}^{\mathcal{R}}) && \Lambda^{\mathcal{R}} = \Lambda \\
&= \mathsf{H}(sk, \Phi, 0\mathrm{LL}') && 0\mathrm{LL}^{\mathcal{R}} = 0\mathrm{LL}' \\
&= \mathsf{H}(sk, \Phi, 0\mathrm{LL}) && \text{from the proof of (6.11a).}
\end{aligned}$$

Substituting into the definition of $\mathfrak{r}^{\mathcal{R}}$

$$\begin{aligned}
\mathfrak{r}^{\mathcal{R}}_{i,\,\kappa,x}(i,\kappa,x,\Lambda^{\mathcal{R}},I^{\mathcal{R}}) &= \mathsf{g}(\mathsf{H}(sk^{\mathcal{R}}, \Phi^{\mathcal{R}}, 0\mathrm{LL}^{\mathcal{R}}), i, \kappa, x) && (6.9) \\
&= \mathsf{g}(\mathsf{H}(sk, \Phi, 0\mathrm{LL}), i, \kappa, x) \\
&= \mathfrak{r}_{i,\,\kappa,x}(i,\kappa,x,\Lambda,I) && (6.9)
\end{aligned}$$

and, similarly for $\Psi^{\mathcal{R}}$

$$\begin{aligned}
\Psi^{\mathcal{R}}_{i,\,\kappa}(i,\kappa,\Lambda^{\mathcal{R}},I^{\mathcal{R}}) &= \mathsf{g}(\mathsf{H}(sk^{\mathcal{R}}, \Phi^{\mathcal{R}}, 0\mathrm{LL}^{\mathcal{R}}), i)_{\kappa} && (6.6) \\
&= \mathsf{g}(\mathsf{H}(sk, \Phi, 0\mathrm{LL}), i)_{\kappa} \\
&= \Psi_{i,\,\kappa}(i,\kappa,x,\Lambda,I) && (6.6).
\end{aligned}$$

Since $v_i^{\mathcal{R}} \in V^{\mathcal{R}}$ it must be the case that $v_i^{\mathcal{R}} \in I^{\mathcal{R}}$ (5.4b), so the image $I^{\mathcal{R}}$ must contain at least $R^{\mathcal{R}} > r_i^{\mathcal{R}} = r_i$ resolution layers. Thus the dimensions of all subbands and resolutions at or below resolution r_i are unchanged (see the proof of (5.13b) in sect. D.3.3)

$$X^{\mathcal{R}}[r] = X[r] \text{ and } Y^{\mathcal{R}}[r] = Y[r] \quad 0 \le r \le r_i^{\mathcal{R}} = r_i$$

and

$$X^{\mathcal{R}}[r,o] = X[r,o] \text{ and } Y^{\mathcal{R}}[r,o] = Y[r,o] \quad 0 \le r \le r_i^{\mathcal{R}} = r_i, 0 \le o \le 2.$$

Also, the coding parameters and the positions of received coefficients for subbands in received resolutions $0 \le r \le r_i$ are unchanged by resolution scaling, so $xcb'^{\mathcal{R}} = xcb'$, $ycb'^{\mathcal{R}} = ycb'$ and $tbx0^{\mathcal{R}} = tbx0$, $tby0^{\mathcal{R}} = tby0$, $tbx1^{\mathcal{R}} = tbx1$, $tby1^{\mathcal{R}} = tby1$, $x_i^{\mathcal{R}} = x_i$ and $y_i^{\mathcal{R}} = y_i$. Finally, the number of components is unchanged by resolution scaling so $C^{\mathcal{R}} = C$.

Thus, substituting these and $\Psi^{\mathcal{R}}_{i,\kappa} = \Psi_{i,\kappa}$ into the definition for $\mathfrak{I}^{\mathcal{R}}{}_{i,\kappa}$,

$$\mathfrak{I}^{\mathcal{R}}{}_{i,\kappa}(i,\kappa,\Lambda^{\mathcal{R}},I^{\mathcal{R}}) = \begin{cases} \left[C^{\mathcal{R}}X^{\mathcal{R}}[r_i^{\mathcal{R}}-1]Y^{\mathcal{R}}[r_i^{\mathcal{R}}-1], C^{\mathcal{R}}X^{\mathcal{R}}[r_i^{\mathcal{R}}]Y^{\mathcal{R}}[r_i^{\mathcal{R}}] \right) & \text{if } \Psi^{\mathcal{R}}_{i,\kappa} = 0 \\ \bigcup_{n_x^{\mathcal{R}}, n_y^{\mathcal{R}}} \Big\{ c_i^{\mathcal{R}} X^{\mathcal{R}}[r_i^{\mathcal{R}}]Y^{\mathcal{R}}[r_i^{\mathcal{R}}] & \\ \quad +(c_i^{\mathcal{R}} - c_i^{\mathcal{R}})X^{\mathcal{R}}[r_i^{\mathcal{R}}-1]Y^{\mathcal{R}}[r_i^{\mathcal{R}}-1] & \\ \quad +\sum_{a=0}^{a=o_i^{\mathcal{R}}-1} X^{\mathcal{R}}[r_i^{\mathcal{R}},a]Y^{\mathcal{R}}[r_i^{\mathcal{R}},a] & \\ \quad +(n_y^{\mathcal{R}} - \text{tby0}^{\mathcal{R}})X[r_i^{\mathcal{R}},o_i^{\mathcal{R}}] & \\ \quad +n_x^{\mathcal{R}} - \text{tbx0}^{\mathcal{R}} \Big\} & \text{if } \Psi^{\mathcal{R}}_{i,\kappa} = 1 \end{cases} \tag{6.7}$$

where $n_x^{\mathcal{R}}, n_y^{\mathcal{R}} \in \mathbb{Z}$ such that

$$\max\left(\left\lfloor \frac{\text{tbx0}^{\mathcal{R}} + x_i^{\mathcal{R}}}{2^{\text{xcb}'^{\mathcal{R}}}} \right\rfloor 2^{\text{xcb}'^{\mathcal{R}}}, \text{tbx0}^{\mathcal{R}} \right) \le n_x^{\mathcal{R}} < \min\left(\left\lceil \frac{\text{tbx0}^{\mathcal{R}} + x_i^{\mathcal{R}}}{2^{\text{xcb}'^{\mathcal{R}}}} \right\rceil 2^{\text{xcb}'^{\mathcal{R}}}, \text{tbx1}^{\mathcal{R}} \right),$$
$$\max\left(\left\lfloor \frac{\text{tby0}^{\mathcal{R}} + y_i^{\mathcal{R}}}{2^{\text{ycb}'^{\mathcal{R}}}} \right\rfloor 2^{\text{ycb}'^{\mathcal{R}}}, \text{tby0}^{\mathcal{R}} \right) \le n_y^{\mathcal{R}} < \min\left(\left\lceil \frac{\text{tby0}^{\mathcal{R}} + y_i^{\mathcal{R}}}{2^{\text{ycb}'^{\mathcal{R}}}} \right\rceil 2^{\text{ycb}'^{\mathcal{R}}}, \text{tby1}^{\mathcal{R}} \right)$$

$$= \begin{cases} \left[CX[r_i-1]Y[r_i-1], CX[r_i]Y[r_i] \right) & \text{if } \Psi_{i,\kappa} = 0 \\ \bigcup_{n'_x, n'_y} \Big\{ c_i X[r_i]Y[r_i] & \\ \quad +(c_i - c_i)X[r_i-1]Y[r_i-1] & \\ \quad +\sum_{a=0}^{a=o_i-1} X[r_i,a]Y[r_i,a] & \\ \quad +(n_y^{\mathcal{R}} - \text{tby0})X[r_i,o_i] & \\ \quad +n_x^{\mathcal{R}} - \text{tbx0} \Big\} & \text{if } \Psi_{i,\kappa} = 1 \end{cases}$$

with $n_x^{\mathcal{R}}, n_y^{\mathcal{R}} \in \mathbb{Z}$ such that

$$\max\left(\left\lfloor \frac{\mathrm{tbx0}+x_i}{2^{\mathrm{xcb}'}} \right\rfloor 2^{\mathrm{xcb}'}, \mathrm{tbx0}\right) \le n_x < \min\left(\left\lceil \frac{\mathrm{tbx0}+x_i}{2^{\mathrm{xcb}'}} \right\rceil 2^{\mathrm{xcb}'}, \mathrm{tbx1}\right),$$
$$\max\left(\left\lfloor \frac{\mathrm{tby0}+y_i}{2^{\mathrm{ycb}'}} \right\rfloor 2^{\mathrm{ycb}'}, \mathrm{tby0}\right) \le n_y < \min\left(\left\lceil \frac{\mathrm{tby0}+y_i}{2^{\mathrm{ycb}'}} \right\rceil 2^{\mathrm{ycb}'}, \mathrm{tby1}\right)$$

$$= \mathfrak{I}_{i,\,\kappa}(i, \kappa, \Lambda, I) \qquad (6.7)$$

we obtain $\mathfrak{I}^{\mathcal{R}}{}_{i,\,\kappa} = \mathfrak{I}_{i,\,\kappa}$. Thus

$$\begin{aligned}
\mathfrak{f}^{\mathcal{R}}(i,\,\kappa,x) &= \mathfrak{I}^{\mathcal{R}}_{i,\,\kappa}(\mathfrak{r}^{\mathcal{R}}{}_{i,\,\kappa,x}) && (6.10)\\
&= \mathfrak{I}^{\mathcal{R}}_{i,\,\kappa}(\mathfrak{r}_{i,\,\kappa,x}) && \mathfrak{r}^{\mathcal{R}}{}_{i,\,\kappa,x} = \mathfrak{r}_{i,\,\kappa,x}\\
&= \mathfrak{I}_{i,\,\kappa}(\mathfrak{r}_{i,\,\kappa,x}) && \mathfrak{I}^{\mathcal{R}}{}_{i,\,\kappa} = \mathfrak{I}_{i,\,\kappa}\\
&= \mathfrak{f}(i,\,\kappa,x) && (6.10)
\end{aligned}$$

and property (6.11c) is proven.

To prove the remaining property we observe, from definition of the feature sequence (6.5)

$$V^{\mathcal{R}}_{i,\,\kappa}(i,\kappa,\Lambda^{\mathcal{R}},I^{\mathcal{R}}) = \{v^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\,\kappa,0)}, v^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\,\kappa,1)}, \ldots, v^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\,\kappa,\eta^{\mathcal{R}}-1)}\} \qquad \kappa \in \mathbb{Z}, 0 \le \kappa < j_i^{\mathcal{R}},$$

that a coefficient $v^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\,\kappa,x)} \in I^{\mathcal{R}}$ belongs to the feature sequence $V^{\mathcal{R}}_{i,\,\kappa}$ if and only if its index $\mathfrak{f}^{\mathcal{R}}(i,\,\kappa,x)$ is in the $\mathfrak{r}^{\mathcal{R}}{}_{i,\,\kappa,x}$th position of the index sequence $\mathfrak{I}^{\mathcal{R}}_{i,\,\kappa}$:[1]

$$v^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\,\kappa,x)} \in V^{\mathcal{R}}_{i,\,\kappa} \iff \mathfrak{f}^{\mathcal{R}}(i,\,\kappa,x) = \mathfrak{I}^{\mathcal{R}}_{i,\,\kappa}(\mathfrak{r}^{\mathcal{R}}{}_{i,\,\kappa,x}) \quad \kappa, x \in \mathbb{Z}, 0 \le \kappa < j_i^{\mathcal{R}}, 0 \le x < \eta^{\mathcal{R}}$$

so, since $\mathfrak{f}^{\mathcal{R}}(i,\,\kappa,x) = \mathfrak{I}^{\mathcal{R}}_{i,\,\kappa}(\mathfrak{r}^{\mathcal{R}}{}_{i,\,\kappa,x}) = \mathfrak{I}_{i,\,\kappa}(\mathfrak{r}_{i,\,\kappa,x}) = \mathfrak{f}(i,\,\kappa,x)$,

$$\begin{aligned}
v^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\,\kappa,x)} \in V^{\mathcal{R}}_{i,\,\kappa} &\iff \mathfrak{f}(i,\,\kappa,x) = \mathfrak{I}_{i,\,\kappa}(\mathfrak{r}_{i,\,\kappa,x}) \quad \kappa, x \in \mathbb{Z}, 0 \le \kappa < j_i^{\mathcal{R}}, 0 \le x < \eta^{\mathcal{R}}\\
&\iff \mathfrak{f}(i,\,\kappa,x) = \mathfrak{I}_{i,\,\kappa}(\mathfrak{r}_{i,\,\kappa,x}) \quad \kappa, x \in \mathbb{Z}, 0 \le \kappa < j_i, 0 \le x < \eta^{\mathcal{R}} && (6.4\mathrm{b})\\
&\iff \mathfrak{f}(i,\,\kappa,x) = \mathfrak{I}_{i,\,\kappa}(\mathfrak{r}_{i,\,\kappa,x}) \quad \kappa, x \in \mathbb{Z}, 0 \le \kappa < j_i, 0 \le x < \eta && \Lambda^{\mathcal{R}} = \Lambda\\
&\iff v_{\mathfrak{f}(i,\,\kappa,x)} \in V_{i,\,\kappa}.
\end{aligned}$$

so property (6.11d) is proven.

[1] Note that since $v_i^{\mathcal{R}} \in I^{\mathcal{R}}$ has been received and any feature coefficient $v^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\,\kappa,x)}$ belongs to the same resolution layer as $v_i^{\mathcal{R}} \in I^{\mathcal{R}}$ so it must also have been received, $v^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\,\kappa,x)} \in I^{\mathcal{R}}$

3. Quality Scaled

Let $\text{kmax} = \log_2\left(\max_{x=0}^{CX[0]Y[0]-1} \bar{v}_x\right)$ denote the number of most significant bit planes in LL0'. If $I^{\mathcal{Q}} = \mathcal{Q}(I')$ is a quality scaled image for which the Φ most significant bit planes of the lowest resolution layer have not been lost, then for all coefficients $v_i^{\mathcal{Q}}$ in $LL0^{\mathcal{Q}}$, the number of bits $m_i^{\mathcal{Q}}$ missing from $v_i^{\mathcal{Q}}$ can be no more than $\text{kmax} - \Phi$

$$m_i^{\mathcal{Q}} \leq \text{kmax} - \Phi \qquad \forall v_i^{\mathcal{Q}} \in \text{LL0}^{\mathcal{Q}}.$$

Thus for any coefficient $v_i^{\mathcal{Q}}$ in the lowest resolution layer $0\text{LL}^{\mathcal{Q}}$ the magnitude and sign computed from the Φ most significant bit planes is

$$\begin{aligned}
&sign(v_i^{\mathcal{Q}})Q_{2^{\text{kmax}-\Phi}}(v_i^{\mathcal{Q}}) \\
&\quad = \text{sign}(v_i^{\mathcal{Q}}) \left\lfloor \frac{|v_i^{\mathcal{Q}}|}{2^{\text{kmax}-\Phi}} \right\rfloor 2^{\text{kmax}-\Phi} && \text{(5.3a)} \\
&\quad = \text{sign}(v_i^{\mathcal{Q}}) \left\lfloor \frac{\left\lfloor \frac{|v_i'|}{2^{m_i^{\mathcal{Q}}}} \right\rfloor 2^{m_i^{\mathcal{Q}}} + \left\lfloor 2^{m_i^{\mathcal{Q}}} \text{r} \right\rfloor}{2^{\text{kmax}-\Phi}} \right\rfloor 2^{\text{kmax}-\Phi} && \text{(2.20a)} \\
&\quad = \text{sign}(v_i^{\mathcal{Q}}) \left\lfloor \frac{\left\lfloor \frac{|v_i'|}{2^{m_i^{\mathcal{Q}}}} \right\rfloor + \frac{\left\lfloor 2^{m_i^{\mathcal{Q}}} \text{r} \right\rfloor}{2^{m_i^{\mathcal{Q}}}}}{2^{\text{kmax}-\Phi-m_i^{\mathcal{Q}}}} \right\rfloor 2^{\text{kmax}-\Phi}
\end{aligned}$$

but $\left\lfloor \frac{|v_i'|}{2^{m_i^{\mathcal{Q}}}} \right\rfloor \in \mathbb{Z}$, and $0 \leq \mathrm{r} < 1$ so $0 \leq \frac{\left\lfloor 2^{m_i^{\mathcal{Q}}} \mathrm{r} \right\rfloor}{2^{m_i^{\mathcal{Q}}}} < 1$ and $m_i^{\mathcal{Q}}, \mathrm{kmax}, \Phi \in \mathbb{Z}, m_i^{\mathcal{Q}} \leq \mathrm{kmax} - \Phi$ so $2^{\mathrm{kmax}-\Phi-m_i^{\mathcal{Q}}} \in \mathbb{N}$ thus

$$
\begin{aligned}
&= \mathrm{sign}(v_i^{\mathcal{Q}}) \left\lfloor \frac{\left\lfloor \frac{|v_i'|}{2^{m_i^{\mathcal{Q}}}} \right\rfloor}{2^{\mathrm{kmax}-\Phi-m_i^{\mathcal{Q}}}} \right\rfloor 2^{\mathrm{kmax}-\Phi} && \text{Lemma D.2.2} \\
&= \mathrm{sign}(v_i^{\mathcal{Q}}) \left\lfloor \frac{\left\lfloor \frac{|v_i'|}{2^{m_i^{\mathcal{Q}}}} \right\rfloor 2^{m_i^{\mathcal{Q}}}}{2^{\mathrm{kmax}-\Phi}} \right\rfloor 2^{\mathrm{kmax}-\Phi} \\
&= \mathrm{sign}(v_i^{\mathcal{Q}}) \left\lfloor \frac{|v_i'|}{2^{\mathrm{kmax}-\Phi}} \right\rfloor 2^{\mathrm{kmax}-\Phi} && \text{Lemma D.2.3} \\
&= \begin{cases} \mathrm{sign}(v_i^{\mathcal{Q}}) \left\lfloor \frac{|v_i'|}{2^{\mathrm{kmax}-\Phi}} \right\rfloor 2^{\mathrm{kmax}-\Phi} & \text{if } |v_i'| \geq 2^{\mathrm{kmax}-\Phi} \\ 0 & \text{if } |v_i'| < 2^{\mathrm{kmax}-\Phi} \end{cases} \\
&= \begin{cases} \mathrm{sign}(v_i^{\mathcal{Q}}) Q_{2^{\mathrm{kmax}-\Phi}}(v_i') & \text{if } |v_i'| \geq 2^{\mathrm{kmax}-\Phi} \\ 0 & \text{if } |v_i'| < 2^{\mathrm{kmax}-\Phi} \end{cases} && \text{(5.3a)} \\
&= \begin{cases} \mathrm{sign}(\mathrm{sign}(v_i') Q_{2^{m_i^{\mathcal{Q}}}}(v_i') + \left\lfloor 2^{m_i^{\mathcal{Q}}} \mathrm{r} \right\rfloor) Q_{2^{\mathrm{kmax}-\Phi}}(v_i) & \\ \qquad\qquad \text{if } |v_i'| \geq 2^{\mathrm{kmax}-\Phi} && \text{(2.20a)} \\ 0 \qquad\qquad \text{if } |v_i'| < 2^{\mathrm{kmax}-\Phi} \end{cases}
\end{aligned}
$$

but $2^{m_i^{\mathcal{Q}}} \mathrm{r} \geq 0$ and $m_i^{\mathcal{Q}} \leq \mathrm{kmax} - \Phi$, so $Q_{2^{m_i^{\mathcal{Q}}}}(v_i') + \left\lfloor 2^{m_i^{\mathcal{Q}}} \mathrm{r} \right\rfloor > 0$ whenever $|v_i'| \geq 2^{\mathrm{kmax}-\Phi}$, so this simplifies to

$$
\begin{aligned}
&= \begin{cases} \mathrm{sign}(v_i') Q_{2^{\mathrm{kmax}-\Phi}}(v_i') & \text{if } |v_i'| \geq 2^{\mathrm{kmax}-\Phi} \\ 0 & \text{if } |v_i'| < 2^{\mathrm{kmax}-\Phi} \end{cases} \\
&= \mathrm{sign}(v_i) Q_{2^{\mathrm{kmax}-\Phi}}(v_i) && \text{from the } \textbf{1. Unscaled} \text{ proof,}
\end{aligned}
$$

which is equal to the magnitude and sign computed from the Φ most significant bit planes of the corresponding coefficient in 0LL. Thus the hashes of the Φ most significant bit planes of $LL0$ and $LL^{\mathcal{Q}}$ are equal

$$
\begin{aligned}
\mathsf{H}(sk^{\mathcal{Q}}, \Phi^{\mathcal{Q}}, 0\mathrm{LL}^{\mathcal{Q}}) &= \mathsf{H}(sk, \Phi, 0\mathrm{LL}^{\mathcal{Q}}) && \Lambda^{\mathcal{Q}} = \Lambda \\
&= \mathsf{H}(sk, \Phi, 0\mathrm{LL}') && 0\mathrm{LL}^{\mathcal{Q}} = 0\mathrm{LL}' \\
&= \mathsf{H}(sk, \Phi, 0\mathrm{LL}) && \text{from the } \textbf{1. Unscaled} \text{ proof.}
\end{aligned}
$$

If $I^{\mathcal{Q}} = \mathcal{Q}(I')$ is a quality scaled image, for which the Φ most significant bit planes of the lowest resolution layer have not been lost, and $\Lambda^{\mathcal{Q}} = \Lambda$, and $v_i^{\mathcal{Q}} \in V^{\mathcal{Q}}$ is the coefficient

corresponding to $v'_i \in V'$ then substituting into the definition for $\mathfrak{r}^{\mathcal{Q}}$ we obtain

$$\begin{aligned}\mathfrak{r}^{\mathcal{Q}}_{i,\,\kappa,x}(i,\kappa,x,\Lambda^{\mathcal{Q}},I^{\mathcal{Q}}) &= \mathsf{g}(\mathsf{H}(sk^{\mathcal{Q}},\Phi^{\mathcal{Q}},0\mathrm{LL}^{\mathcal{Q}}),i,\kappa,x) && (6.9)\\ &= \mathsf{g}(\mathsf{H}(sk,\Phi,0\mathrm{LL}),i,\kappa,x)\\ &= \mathfrak{r}_{i,\,\kappa,x}(i,\kappa,x,\Lambda,I) && (6.9)\end{aligned}$$

and, similarly, for $\Psi^{\mathcal{Q}}$

$$\begin{aligned}\Psi^{\mathcal{Q}}_{i,\,\kappa}(i,\kappa,\Lambda^{\mathcal{Q}},I^{\mathcal{Q}}) &= \mathsf{g}(\mathsf{H}(sk^{\mathcal{Q}},\Phi^{\mathcal{Q}},0\mathrm{LL}^{\mathcal{Q}}),i)_\kappa && (6.6)\\ &= \mathsf{g}(\mathsf{H}(sk,\Phi,0\mathrm{LL}),i)_\kappa\\ &= \Psi_{i,\,\kappa}(i,\kappa,x,\Lambda,I). && (6.6)\end{aligned}$$

Because the watermark embedding process causes no change to the number of components, image dimensions or codeblock parameters or sample positions, $C^{\mathcal{Q}} = C$ and $X^{\mathcal{Q}}[r] = X[r]$, $Y^{\mathcal{Q}}[r] = Y[r]$, $X^{\mathcal{Q}}[r,s] = X[r,s]$, $Y^{\mathcal{Q}}[r,s] = Y[r,s], \forall r,s$ and $xcb'^{\mathcal{Q}} = xcb'$, $ycb'^{\mathcal{Q}} = ycb'$ and $tbx0^{\mathcal{Q}} = tbx0$, $tby0^{\mathcal{Q}} = tby0$, $tbx1^{\mathcal{Q}} = tbx1$, $tby1^{\mathcal{Q}} = tby1$, $x_i^{\mathcal{Q}} = x_i$ and $y_i^{\mathcal{Q}} = y_i$. Thus, substituting these and $\Psi^{\mathcal{Q}}_{i,\,\kappa} = \Psi_{i,\,\kappa}$ into the definition for $\mathfrak{I}^{\mathcal{Q}}{}_{i,\,\kappa}$,

$$\mathfrak{I}^{\mathcal{Q}}{}_{i,\,\kappa}(i,\kappa,\Lambda^{\mathcal{Q}},I^{\mathcal{Q}}) = \begin{cases} \left[C^{\mathcal{Q}}X^{\mathcal{Q}}[r_i^{\mathcal{Q}}-1]Y^{\mathcal{Q}}[r_i^{\mathcal{Q}}-1], C^{\mathcal{Q}}X^{\mathcal{Q}}[r_i^{\mathcal{Q}}]Y^{\mathcal{Q}}[r_i^{\mathcal{Q}}]\right) & \text{if } \Psi^{\mathcal{Q}}_{i,\,\kappa} = 0 \\ \bigcup_{n_x^{\mathcal{Q}},n_y^{\mathcal{Q}}} \Big\{ c_i^{\mathcal{Q}}X^{\mathcal{Q}}[r_i^{\mathcal{Q}}]Y^{\mathcal{Q}}[r_i^{\mathcal{Q}}] & \\ \quad +(c_i^{\mathcal{Q}}-c_i^{\mathcal{Q}})X^{\mathcal{Q}}[r_i^{\mathcal{Q}}-1]Y^{\mathcal{Q}}[r_i^{\mathcal{Q}}-1] & \\ \quad +\textstyle\sum_{o=0}^{o=s_i^{\mathcal{Q}}-1} X^{\mathcal{Q}}[r_i^{\mathcal{Q}},o]Y^{\mathcal{Q}}[r_i^{\mathcal{Q}},o] & \\ \quad +(n_y^{\mathcal{Q}}-\mathrm{tby0}^{\mathcal{Q}})X[r_i^{\mathcal{Q}},s_i^{\mathcal{Q}}] & \\ \quad +n_x^{\mathcal{Q}}-\mathrm{tbx0}^{\mathcal{Q}} \Big\} & \text{if } \Psi^{\mathcal{Q}}_{i,\,\kappa} = 1 \end{cases} \qquad (6.7)$$

where $n_x^{\mathcal{Q}}, n_y^{\mathcal{Q}} \in \mathbb{Z}$ such that

$$\begin{aligned}&\max\left(\left\lfloor\frac{\mathrm{tbx0}^{\mathcal{Q}}+x_i^{\mathcal{Q}}}{2^{\mathrm{xcb}'^{\mathcal{Q}}}}\right\rfloor 2^{\mathrm{xcb}'^{\mathcal{Q}}}, \mathrm{tbx0}^{\mathcal{Q}}\right) \le n_x^{\mathcal{Q}} < \min\left(\left\lceil\frac{\mathrm{tbx0}^{\mathcal{Q}}+x_i^{\mathcal{Q}}}{2^{\mathrm{xcb}'^{\mathcal{Q}}}}\right\rceil 2^{\mathrm{xcb}'^{\mathcal{Q}}}, \mathrm{tbx1}^{\mathcal{Q}}\right),\\ &\max\left(\left\lfloor\frac{\mathrm{tby0}^{\mathcal{Q}}+y_i^{\mathcal{Q}}}{2^{\mathrm{ycb}'^{\mathcal{Q}}}}\right\rfloor 2^{\mathrm{ycb}'^{\mathcal{Q}}}, \mathrm{tby0}^{\mathcal{Q}}\right) \le n_y^{\mathcal{Q}} < \min\left(\left\lceil\frac{\mathrm{tby0}^{\mathcal{Q}}+y_i^{\mathcal{Q}}}{2^{\mathrm{ycb}'^{\mathcal{Q}}}}\right\rceil 2^{\mathrm{ycb}'^{\mathcal{Q}}}, \mathrm{tby1}^{\mathcal{Q}}\right) \quad (6.8b)\end{aligned}$$

$$= \begin{cases} \left[CX[r_i-1]Y[r_i-1], CX[r_i]Y[r_i] \right) & \text{if } \Psi_{i,\kappa} = 0 \\ \bigcup_{n'_x, n'_y} \Big\{ c_i X[r_i]Y[r_i] & \\ \qquad +(c_i - c_i)X[r_i-1]Y[r_i-1] & \\ \qquad +\sum_{o=0}^{o=s_i-1} X[r_i, o]Y[r_i, o] & \\ \qquad +(n_y^{\mathcal{Q}} - \mathrm{tby0})X[r_i, s_i] & \\ \qquad +n_x^{\mathcal{Q}} - \mathrm{tbx0} \Big\} & \text{if } \Psi_{i,\kappa} = 1 \end{cases}$$

with $n_x^{\mathcal{Q}}, n_y^{\mathcal{Q}} \in \mathbb{Z}$ such that

$$\max\left(\left\lfloor \frac{\mathrm{tbx0} + x_i}{2^{\mathrm{xcb}'}} \right\rfloor 2^{\mathrm{xcb}'}, \mathrm{tbx0} \right) \le n_x < \min\left(\left\lceil \frac{\mathrm{tbx0} + x_i}{2^{\mathrm{xcb}'}} \right\rceil 2^{\mathrm{xcb}'}, \mathrm{tbx1} \right),$$
$$\max\left(\left\lfloor \frac{\mathrm{tby0} + y_i}{2^{\mathrm{ycb}'}} \right\rfloor 2^{\mathrm{ycb}'}, \mathrm{tby0} \right) \le n_y < \min\left(\left\lceil \frac{\mathrm{tby0} + y_i}{2^{\mathrm{ycb}'}} \right\rceil 2^{\mathrm{ycb}'}, \mathrm{tby1} \right)$$

$$= \mathfrak{I}_{i,\kappa}(i, \kappa, \Lambda, I), \tag{6.7}$$

we obtain $\mathfrak{I}^{\mathcal{Q}}{}_{i,\kappa}(i, \kappa, \Lambda^{\mathcal{Q}}, I^{\mathcal{Q}}) = \mathfrak{I}_{i,\kappa}(i, \kappa, \Lambda, I)$. Thus

$$\begin{aligned} \mathfrak{f}^{\mathcal{Q}}(i,\, \kappa, x) &= \mathfrak{I}^{\mathcal{Q}}_{i,\kappa}(\mathfrak{r}^{\mathcal{Q}}{}_{i,\kappa,x}) && (6.10) \\ &= \mathfrak{I}^{\mathcal{Q}}_{i,\kappa}(\mathfrak{r}_{i,\kappa,x}) && \mathfrak{r}^{\mathcal{Q}}{}_{i,\kappa,x} = \mathfrak{r}_{i,\kappa,x} \\ &= \mathfrak{I}_{i,\kappa}(\mathfrak{r}_{i,\kappa,x}) && \mathfrak{I}^{\mathcal{Q}}{}_{i,\kappa} = \mathfrak{I}_{i,\kappa} \\ &= \mathfrak{f}(i,\, \kappa, x) && (6.10) \end{aligned}$$

and the property (6.11e) is proven.

To prove remaining property we observe, from the definition of the feature sequence (6.5)

$$V^{\mathcal{Q}}_{i,\kappa}(i, \kappa, \Lambda^{\mathcal{Q}}, I^{\mathcal{Q}}) = \{v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,0)}, v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,1)}, \ldots, v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,\eta^{\mathcal{Q}}-1)}\} \qquad \kappa \in \mathbb{Z}, 0 \le \kappa < j_i^{\mathcal{Q}},$$

that a coefficient $v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)} \in I^{\mathcal{Q}}$ belongs to the feature sequence $V^{\mathcal{Q}}_{i,\kappa}$ if and only if its index $\mathfrak{f}^{\mathcal{Q}}(i,\, \kappa, x)$ has position $\mathfrak{r}^{\mathcal{Q}}{}_{i,\kappa,x}$ in the index sequence $\mathfrak{I}^{\mathcal{Q}}_{i,\kappa}$:

$$v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)} \in V^{\mathcal{Q}}_{i,\kappa} \iff \mathfrak{f}^{\mathcal{Q}}(i,\, \kappa, x) = \mathfrak{I}^{\mathcal{Q}}_{i,\kappa}(\mathfrak{r}^{\mathcal{Q}}{}_{i,\kappa,x}) \quad \kappa, x \in \mathbb{Z}, 0 \le \kappa < j_i^{\mathcal{Q}}, 0 \le x < \eta^{\mathcal{Q}}$$

so, since $\mathfrak{f}^{\mathcal{Q}}(i,\,\kappa,x) = \mathfrak{I}^{\mathcal{Q}}_{i,\,\kappa}(\mathfrak{r}^{\mathcal{Q}}{}_{i,\,\kappa,x}) = \mathfrak{I}_{i,\,\kappa}(\mathfrak{r}_{i,\,\kappa,x}) = \mathfrak{f}(i,\,\kappa,x)$,

$$\begin{aligned} v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\,\kappa,x)} \in V^{\mathcal{Q}}_{i,\,\kappa} &\iff \mathfrak{f}(i,\,\kappa,x) = \mathfrak{I}_{i,\,\kappa}(\mathfrak{r}_{i,\,\kappa,x}) \quad \kappa,x \in \mathbb{Z}, 0 \le \kappa < j^{\mathcal{Q}}_i, 0 \le x < \eta^{\mathcal{Q}} \\ &\iff \mathfrak{f}(i,\,\kappa,x) = \mathfrak{I}_{i,\,\kappa}(\mathfrak{r}_{i,\,\kappa,x}) \quad \kappa,x \in \mathbb{Z}, 0 \le \kappa < j_i, 0 \le x < \eta^{\mathcal{Q}} \qquad (6.4c) \\ &\iff \mathfrak{f}(i,\,\kappa,x) = \mathfrak{I}_{i,\,\kappa}(\mathfrak{r}_{i,\,\kappa,x}) \quad \kappa,x \in \mathbb{Z}, 0 \le \kappa < j_i, 0 \le x < \eta \qquad \Lambda^{\mathcal{Q}} = \Lambda \\ &\iff v_{\mathfrak{f}(i,\,\kappa,x)} \in V_{i,\,\kappa}. \end{aligned}$$

So property (6.11f) is proven.

E.1.3 Proof for Watermark Bit Construction

We show that the procedure for determining the feature quantization step size described in section 6.1.3.2 ensures that given the correct watermarking parameters $\Lambda^* = \Lambda$ and a potentially scaled but otherwise untampered watermarked image, of which at least the Φ most significant bit planes of the lowest resolution layer have not been lost, quantization step size exponents for corresponding feature coefficients are identical provided the feature coefficient has not been completely lost due to scaling

- for the unscaled watermarked image, where $v'_i \in V'$ is the coefficient corresponding to $v_i \in V$

$$\forall v'_{\mathfrak{f}'(i,\,\kappa,x)} \in V'_{i,\,\kappa}, q'_{i,\,\kappa,x} = q_{i,\,\kappa,x} \qquad (6.14a)$$

- for the resolution scaled watermarked image, where $v^{\mathcal{R}}_i \in V^{\mathcal{R}}$ is the coefficient corresponding to $v'_i \in V'$

$$\forall v^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\,\kappa,x)} \in V^{\mathcal{R}}_{i,\,\kappa}, q^{\mathcal{R}}_{i,\,\kappa,x} = q_{i,\,\kappa,x} \qquad (6.14b)$$

- for the quality scaled watermarked image, where $v^{\mathcal{Q}}_i \in V^{\mathcal{Q}}$ is the coefficient corresponding to $v'_i \in V'$

$$\forall v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\,\kappa,x)} \neq 0 \in V^{\mathcal{Q}}_{i,\,\kappa}, q^{\mathcal{Q}}_{i,\,\kappa,x} = q_{i,\,\kappa,x}. \qquad (6.14c)$$

Let the quantization step size for the feature coefficient $v^*_{\mathfrak{f}^*(i^*,\,\kappa^*,x^*)}$ be

$$q^*_{i^*,\,\kappa^*,x^*} = \begin{cases} \max(M^*_{s^*_{\mathfrak{f}^*(i^*,\,\kappa^*,x^*)}} - (1 + \lfloor \frac{j^*_{i^*} - 1 - \kappa^*}{a^*} \rfloor), j^*_{\mathfrak{f}^*(i^*,\,\kappa^*,x^*)}) & \text{if } \Psi_{i^*,\,\kappa^*} = 0 \\ \max(\kappa^* + 1, j^*_{\mathfrak{f}^*(i^*,\,\kappa^*,x^*)}) & \text{if } \Psi_{i^*,\,\kappa^*} = 1 \end{cases} \qquad (6.13)$$

where $j^*_{\mathfrak{f}^*(i^*,\,\kappa^*,x^*)}$ is the number of embedded bits for $v^*_{\mathfrak{f}^*(i^*,\,\kappa^*,x^*)}$, $M_{s_{\mathfrak{f}^*(i^*,\,\kappa^*,x^*)}}$ is the maximum number of significant bit planes in the subband $s_{\mathfrak{f}^*(i^*,\,\kappa^*,x^*)}$ of resolution $r_{\mathfrak{f}^*(i^*,\,\kappa^*,x^*)}$, $\Psi_{i^*,\,\kappa^*}$ is the sequence type, and $a^* \in \Lambda^*$ is a given feature robustness parameter.

1. Unscaled

If $I' = \mathsf{Embed}(I, \Lambda)$ is the unscaled watermarked image, $\Lambda' = \Lambda$, and $v'_i \in V'$ is the coefficient corresponding to $v_i \in V$ then, $\forall v'_{\mathfrak{f}'(i,\,\kappa,x)} \in V'_{i,\,\kappa}$,

$$q'_{i,\,\kappa,x} = \begin{cases} \max(M'_{s'_{\mathfrak{f}'(i,\,\kappa,x)}} - (1 + \lfloor \frac{j'_i - 1 - \kappa}{a'} \rfloor), j'_{\mathfrak{f}'(i,\,\kappa,x)}) & \text{if } \Psi'_{i,\,\kappa} = 0 \\ \max(\kappa + 1, j'_{\mathfrak{f}'(i,\,\kappa,x)}) & \text{if } \Psi'_{i,\,\kappa} = 1 \end{cases} \tag{6.13}$$

which, since $\mathfrak{f}'(i,\,\kappa,x) = \mathfrak{f}(i,\,\kappa,x)$ (6.11a)

$$= \begin{cases} \max(M'_{s'_{\mathfrak{f}(i,\,\kappa,x)}} - (1 + \lfloor \frac{j'_i - 1 - \kappa}{a'} \rfloor), j'_{\mathfrak{f}(i,\,\kappa,x)}) & \text{if } \Psi'_{i,\,\kappa} = 0 \\ \max(\kappa + 1, j'_{\mathfrak{f}(i,\,\kappa,x)}) & \text{if } \Psi'_{i,\,\kappa} = 1 \end{cases}$$

but, $\forall v'_x \in I'$, $j'_x = j_x$ (6.4a), so this becomes

$$= \begin{cases} \max(M'_{s'_{\mathfrak{f}(i,\,\kappa,x)}} - (1 + \lfloor \frac{j_i - 1 - \kappa}{a'} \rfloor), j_{\mathfrak{f}(i,\,\kappa,x)}) & \text{if } \Psi'_{i,\,\kappa} = 0 \\ \max(\kappa + 1, j_{\mathfrak{f}(i,\,\kappa,x)}) & \text{if } \Psi'_{i,\,\kappa} = 1 \end{cases}$$

which, as $\Lambda' = \Lambda$ gives $\Psi'_{i,\,\kappa} = \Psi_{i,\,\kappa}$ and $a' = a$,

$$= \begin{cases} \max(M'_{s'_{\mathfrak{f}(i,\,\kappa,x)}} - (1 + \lfloor \frac{j_i - 1 - \kappa}{a} \rfloor), j_{\mathfrak{f}(i,\,\kappa,x)}) & \text{if } \Psi_{i,\,\kappa} = 0 \\ \max(\kappa + 1, j_{\mathfrak{f}(i,\,\kappa,x)}) & \text{if } \Psi_{i,\,\kappa} = 1. \end{cases}$$

Because watermarking does not affect the positions of coefficients or the locations of subbands, $s'_{\mathfrak{f}(i,\,\kappa,x)} = s_{\mathfrak{f}(i,\,\kappa,x)}$. Recall, from section 5.1.6, that $M_{s_{\mathfrak{f}(i,\,\kappa,x)}}$ is derived from the QCC or QCD marker segments, which are also unchanged by watermarking so $M'_{s_{\mathfrak{f}(i,\,\kappa,x)}} = M_{s_{\mathfrak{f}(i,\,\kappa,x)}}$. Thus

$$q'_{i,\,\kappa,x} = \begin{cases} \max(M_{s_{\mathfrak{f}(i,\,\kappa,x)}} - (1 + \lfloor \frac{j_i - 1 - \kappa}{a} \rfloor), j_{\mathfrak{f}(i,\,\kappa,x)}) & \text{if } \Psi_{i,\,\kappa} = 0 \\ \max(\kappa + 1, j_{\mathfrak{f}(i,\,\kappa,x)}) & \text{if } \Psi_{i,\,\kappa} = 1. \end{cases}$$

$$= q_{i,\,\kappa,x}$$

and property (6.14a) is proven.

2. Resolution Scaled

If $I^{\mathcal{R}} = \mathcal{R}(I')$ is a resolution scaled image, of which at least the Φ most significant bit planes of the lowest resolution layer have not been lost, $\Lambda^{\mathcal{R}} = \Lambda$ and $v_i^{\mathcal{R}} \in V^{\mathcal{R}}$ is the coefficient corresponding to $v'_i \in V'$ then $\forall v^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\,\kappa,x)} \in V^{\mathcal{R}}_{i,\,\kappa}$,

$$q^{\mathcal{R}}_{i,\,\kappa,x} = \begin{cases} \max(M^{\mathcal{R}}_{s^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\,\kappa,x)}} - (1 + \lfloor \frac{j_i^{\mathcal{R}} - 1 - \kappa}{a^{\mathcal{R}}} \rfloor), j^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\,\kappa,x)}) & \text{if } \Psi^{\mathcal{R}}_{i,\,\kappa} = 0 \\ \max(\kappa + 1, j^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\,\kappa,x)}) & \text{if } \Psi^{\mathcal{R}}_{i,\,\kappa} = 1 \end{cases} \tag{6.13}$$

which, since $\mathfrak{f}^{\mathcal{R}}(i,\,\kappa,x) = \mathfrak{f}(i,\,\kappa,x)$ (6.11c)

$$= \begin{cases} \max(M^{\mathcal{R}}_{s^{\mathcal{R}}_{\mathfrak{f}(i,\,\kappa,x)}} - (1 + \lfloor \frac{j_i^{\mathcal{R}}-1-\kappa}{a^{\mathcal{R}}} \rfloor), j^{\mathcal{R}}_{\mathfrak{f}(i,\,\kappa,x)}) & \text{if } \Psi^{\mathcal{R}}_{i,\,\kappa} = 0 \\ \max(\kappa+1, j^{\mathcal{R}}_{\mathfrak{f}(i,\,\kappa,x)}) & \text{if } \Psi^{\mathcal{R}}_{i,\,\kappa} = 1. \end{cases}$$

Now $v^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\,\kappa,x)} \in V^{\mathcal{R}}_{i,\,\kappa}$ and is thus in the same resolution layer as $v^{\mathcal{R}}_i \in I^{\mathcal{R}}$. Therefore the subband $s^{\mathcal{R}}_{\mathfrak{f}(i,\,\kappa,x)}$, and all other subbands in resolutions $r \leq r^{\mathcal{R}}_{\mathfrak{f}(i,\,\kappa,x)}$ have been received unchanged, so $s^{\mathcal{R}}_{\mathfrak{f}(i,\,\kappa,x)} = s_{\mathfrak{f}(i,\,\kappa,x)}$ and $v^{\mathcal{R}}_{\mathfrak{f}(i,\,\kappa,x)} \in I^{\mathcal{R}}$ so $j^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\,\kappa,x)} = j_{\mathfrak{f}(i,\,\kappa,x)}$ (6.4b). Thus

$$q^{\mathcal{R}}_{i,\,\kappa,x} = \begin{cases} \max(M^{\mathcal{R}}_{s_{\mathfrak{f}(i,\,\kappa,x)}} - (1 + \lfloor \frac{j_i-1-\kappa}{a^{\mathcal{R}}} \rfloor), j_{\mathfrak{f}(i,\,\kappa,x)}) & \text{if } \Psi^{\mathcal{R}}_{i,\,\kappa} = 0 \\ \max(\kappa+1, j_{\mathfrak{f}(i,\,\kappa,x)}) & \text{if } \Psi^{\mathcal{R}}_{i,\,\kappa} = 1 \end{cases}$$

which, as $\Lambda^{\mathcal{R}} = \Lambda$ gives $\Psi^{\mathcal{R}}_{i,\,\kappa} = \Psi_{i,\,\kappa}$ and $a^{\mathcal{R}} = a$,

$$= \begin{cases} \max(M^{\mathcal{R}}_{s_{\mathfrak{f}(i,\,\kappa,x)}} - (1 + \lfloor \frac{j_i-1-\kappa}{a} \rfloor), j_{\mathfrak{f}(i,\,\kappa,x)}) & \text{if } \Psi_{i,\,\kappa} = 0 \\ \max(\kappa+1, j_{\mathfrak{f}(i,\,\kappa,x)}) & \text{if } \Psi_{i,\,\kappa} = 1. \end{cases}$$

Finally, $M_{s_{\mathfrak{f}(i,\,\kappa,x)}}$ is derived from the QCC or QCD marker segments, which are unchanged by resolution scaling[2] so $M^{\mathcal{R}}_{s_{\mathfrak{f}(i,\,\kappa,x)}} = M_{s_{\mathfrak{f}(i,\,\kappa,x)}}$, thus

$$q^{\mathcal{R}}_{i,\,\kappa,x} = \begin{cases} \max(M_{s_{\mathfrak{f}(i,\,\kappa,x)}} - (1 + \lfloor \frac{j_i-1-\kappa}{a} \rfloor), j_{\mathfrak{f}(i,\,\kappa,x)}) & \text{if } \Psi_{i,\,\kappa} = 0 \\ \max(\kappa+1, j_{\mathfrak{f}(i,\,\kappa,x)}) & \text{if } \Psi_{i,\,\kappa} = 1. \end{cases}$$

$$= q_{i,\,\kappa,x}$$

and property (6.14b) is proven.

3. Quality Scaled

If $I^{\mathcal{Q}} = \mathcal{Q}(I')$ is a quality scaled image, of which at least the Φ most significant bit planes of the lowest resolution layer have not been lost, $\Lambda^{\mathcal{Q}} = \Lambda$ and $v^{\mathcal{Q}}_i \in V^{\mathcal{Q}}$ is the coefficient corresponding to $v'_i \in V'$ then $\forall v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\,\kappa,x)} \neq 0 \in V^{\mathcal{Q}}_{i,\,\kappa}$,

$$q^{\mathcal{Q}}_{i,\,\kappa,x} = \begin{cases} \max(M^{\mathcal{Q}}_{s^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\,\kappa,x)}} - (1 + \lfloor \frac{j_i^{\mathcal{Q}}-1-\kappa}{a^{\mathcal{Q}}} \rfloor), j^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\,\kappa,x)}) & \text{if } \Psi^{\mathcal{Q}}_{i,\,\kappa} = 0 \\ \max(\kappa+1, j^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\,\kappa,x)}) & \text{if } \Psi^{\mathcal{Q}}_{i,\,\kappa} = 1 \end{cases} \tag{6.13}$$

which, since $\mathfrak{f}^{\mathcal{Q}}(i,\,\kappa,x) = \mathfrak{f}(i,\,\kappa,x)$ (6.11e) and $j^{\mathcal{Q}}_{\mathfrak{f}(i,\,\kappa,x)} = j_{\mathfrak{f}(i,\,\kappa,x)}$ (6.4c),

$$= \begin{cases} \max(M^{\mathcal{Q}}_{s_{\mathfrak{f}(i,\,\kappa,x)}} - (1 + \lfloor \frac{j_i-1-\kappa}{a^{\mathcal{Q}}} \rfloor), j_{\mathfrak{f}(i,\,\kappa,x)}) & \text{if } \Psi^{\mathcal{Q}}_{i,\,\kappa} = 0 \\ \max(\kappa+1, j_{\mathfrak{f}(i,\,\kappa,x)}) & \text{if } \Psi^{\mathcal{Q}}_{i,\,\kappa} = 1 \end{cases}$$

[2] At least, the values E_{s_i} and G_{c_i} used to compute M_{s_i} are unchanged by resolution scaling for any received subband s_i in component c_i, and the index $\mathfrak{f}(i,\,\kappa,x)$ corresponds to a received subband.

and, as $\Lambda^{\mathcal{Q}} = \Lambda$ gives $\Psi^{\mathcal{Q}}_{i,\kappa} = \Psi_{i,\kappa}$ and $a^{\mathcal{Q}} = a$,

$$= \begin{cases} \max(M^{\mathcal{Q}}_{s_{\mathfrak{f}(i,\kappa,x)}} - (1 + \lfloor \frac{j_i - 1 - \kappa}{a} \rfloor), j_{\mathfrak{f}(i,\kappa,x)}) & \text{if } \Psi_{i,\kappa} = 0 \\ \max(\kappa + 1, j_{\mathfrak{f}(i,\kappa,x)}) & \text{if } \Psi_{i,\kappa} = 1. \end{cases}$$

Because quality scaling does not affect the positions of coefficients or the locations of subbands or the QCC or QCD marker segments, $s^{\mathcal{Q}}_{\mathfrak{f}(i,\kappa,x)} = s_{\mathfrak{f}(i,\kappa,x)}$ and $M^{\mathcal{Q}}_{s_{\mathfrak{f}(i,\kappa,x)}} = M_{s_{\mathfrak{f}(i,\kappa,x)}}$. Thus

$$q^{\mathcal{Q}}_{i,\kappa,x} = \begin{cases} \max(M_{s_{\mathfrak{f}(i,\kappa,x)}} - (1 + \lfloor \frac{j_i - 1 - \kappa}{a} \rfloor), j_{\mathfrak{f}(i,\kappa,x)}) & \text{if } \Psi_{i,\kappa} = 0 \\ \max(\kappa + 1, j_{\mathfrak{f}(i,\kappa,x)}) & \text{if } \Psi_{i,\kappa} = 1. \end{cases}$$

$$= q_{i,\kappa,x}$$

and property (6.14c) is proven.

E.1.4 Proof for Candidate Generation

We show that the candidate generation procedure ensures that, given the correct watermarking parameters and a potentially scaled but otherwise untampered watermarked image, of which at least the Φ most significant bit planes of the lowest resolution layer have not been lost, if a candidate watermark bit exists then it will be identical to the corresponding embedded watermark bit

- for the unscaled watermarked image, where $v'_i \in V'$ is the coefficient corresponding to $v_i \in V$

$$\text{if } \exists u^c_{i,\kappa} \text{ then } u^c_{i,\kappa} = u_{i,\kappa} \tag{6.17a}$$

- for the resolution scaled watermarked image, where $v^{\mathcal{R}}_i \in V^{\mathcal{R}}$ is the coefficient corresponding to $v'_i \in V'$

$$\text{if } \exists u^c_{i,\kappa} \text{ then } u^c_{i,\kappa} = u_{i,\kappa} \tag{6.17b}$$

- for the quality scaled watermarked image, where $v^{\mathcal{Q}}_i \in V^{\mathcal{Q}}$ is the coefficient corresponding to $v'_i \in V'$

$$\text{if } \exists u^c_{i,\kappa} \text{ then } u^c_{i,\kappa} = u_{i,\kappa} \tag{6.17c}$$

Let the candidate watermark bit be as defined in section 6.1.5

$$u^c_{i^*,\kappa^*} = \begin{cases} u^*_{i^*,\kappa^*} & \text{if } \exists v^*_{\mathfrak{f}^*(i^*,\kappa^*,x^*)} \wedge m^*_{\mathfrak{f}^*(i^*,\kappa^*,x^*)} \leq q^*_{i^*,\kappa^*,x^*},\ \forall x^* \in \mathbb{Z} : 0 \leq x^* < \eta^* \\ \nexists & \text{if } \exists x^* \in \mathbb{Z}, 0 \leq x^* < \eta^*, \text{ s.t. } \nexists v^*_{\mathfrak{f}^*(i^*,\kappa^*,x^*)} \vee m^*_{\mathfrak{f}^*(i^*,\kappa^*,x^*)} > q^*_{i^*,\kappa^*,x^*} \end{cases} \tag{6.16}$$

for $\kappa^* \in \mathbb{Z}, 0 \leq \kappa^* < j^*_{i^*}$, where $v^*_{\mathfrak{f}^*(i^*,\kappa^*,x^*)}$ is a feature coefficient with $m^*_{\mathfrak{f}^*(i^*,\kappa^*,x^*)}$ calculated missing bits and quantization step size $q^*_{i^*,\kappa^*,x^*}$, and where $u^*_{i^*,\kappa^*}$ is computed using the watermark bit construction formula

$$u^*_{i^*,\kappa^*} = \bigoplus_{x^*=0}^{\eta^*-1} \mathfrak{S}(v^*_{\mathfrak{f}^*(i^*,\kappa^*,x^*)}) \oplus \mathfrak{M}(v^*_{\mathfrak{f}^*(i,\kappa,x^*-1 \mod \eta^*)}, v^*_{\mathfrak{f}^*(i^*,\kappa^*,x^*)}) \quad (6.12a)$$

where

$$\mathfrak{S}(v^*_{\mathfrak{f}^*(i^*,\kappa^*,x^*)}) = \begin{cases} 0 & \text{if } \operatorname{sign}(v^*_{\mathfrak{f}^*(i^*,\kappa^*,x^*)}) Q_{2^{q^*_{i^*,\kappa^*,x^*}}}(v^*_{\mathfrak{f}^*(i^*,\kappa^*,x^*)})) \geq 0 \\ 1 & \text{if } \operatorname{sign}(v^*_{\mathfrak{f}^*(i^*,\kappa^*,x^*)}) Q_{2^{q^*_{i^*,\kappa^*,x^*}}}(v^*_{\mathfrak{f}^*(i^*,\kappa^*,x^*)})) < 0 \end{cases} \quad (6.12b)$$

represents the sign information, and

$$\mathfrak{M}(v^*_{\mathfrak{f}^*(i^*,\kappa^*,t^*)}, v^*_{\mathfrak{f}^*(i^*,\kappa^*,x^*)}) = \begin{cases} 0 & \text{if } Q_{2^{q^*_{i^*,\kappa^*,t^*}}}(v^*_{\mathfrak{f}^*(i^*,\kappa^*,t^*)}) \leq Q_{2^{q^*_{i^*,\kappa^*,x^*}}}(v^*_{\mathfrak{f}^*(i^*,\kappa^*,x^*)}) \\ 1 & \text{if } Q_{2^{q^*_{i^*,\kappa^*,t^*}}}(v^*_{\mathfrak{f}^*(i^*,\kappa^*,t^*)}) > Q_{2^{q^*_{i^*,\kappa^*,x^*}}}(v^*_{\mathfrak{f}^*(i^*,\kappa^*,x^*)}) \end{cases} \quad (6.12c)$$

1. Unscaled

If $I' = \mathsf{Embed}(I, \Lambda)$ is the unscaled watermarked image, $\Lambda' = \Lambda$, and $v'_i \in V'$ is the coefficient corresponding to $v_i \in V$ then, if the candidate watermark exists $\exists u^c_{i,\kappa}$, from the definition of u^c,

$$u^c_{i,\kappa} = u'_{i,\kappa}, \text{ and } \exists v'_{\mathfrak{f}'(i,\kappa,x)} \wedge m'_{\mathfrak{f}'(i,\kappa,x)} \leq q'_{i,\kappa,x}, \forall x \in \mathbb{Z} : 0 \leq x < \eta'. \quad (6.16)$$

Consider the value $\operatorname{sign}(v'_{\mathfrak{f}'(i,\kappa,x)}) Q_{2^{q'_{i,\kappa,x}}}(v'_{\mathfrak{f}'(i,\kappa,x)})$ used in the definition of $\mathfrak{S}$. There are two possible cases, depending on whether or not the feature coefficient $v'_{\mathfrak{f}'(i,\kappa,x)} \in V'_{i,\kappa}$ also belongs to the set of selected coefficients V'. If $v'_{\mathfrak{f}'(i,\kappa,x)} \notin V'$, then $v_{\mathfrak{f}'(i,\kappa,x)} \notin V$ (5.4a) so no watermarking occurred and $v'_{\mathfrak{f}'(i,\kappa,x)} = v_{\mathfrak{f}'(i,\kappa,x)}$ thus

$$\begin{aligned} \operatorname{sign}(v'_{\mathfrak{f}'(i,\kappa,x)}) Q_{2^{q'_{i,\kappa,x}}}(v'_{\mathfrak{f}'(i,\kappa,x)}) &= \operatorname{sign}(v_{\mathfrak{f}'(i,\kappa,x)}) Q_{2^{q'_{i,\kappa,x}}}(v_{\mathfrak{f}'(i,\kappa,x)}) \\ &= \operatorname{sign}(v_{\mathfrak{f}'(i,\kappa,x)}) Q_{2^{q_{i,\kappa,x}}}(v_{\mathfrak{f}'(i,\kappa,x)}) \quad (6.14a) \\ &= \operatorname{sign}(v_{\mathfrak{f}(i,\kappa,x)}) Q_{2^{q_{i,\kappa,x}}}(v_{\mathfrak{f}(i,\kappa,x)}). \quad (6.11a) \end{aligned}$$

Alternatively, if $v'_{\mathfrak{f}'(i,\,\kappa,x)} \in V'$,

$$\begin{aligned}
\operatorname{sign}(v'_{\mathfrak{f}'(i,\,\kappa,x)})Q_{2^{q'_{i,\,\kappa,x}}}(v'_{\mathfrak{f}'(i,\,\kappa,x)}) &= \operatorname{sign}(v'_{\mathfrak{f}'(i,\,\kappa,x)}) \left\lfloor \frac{|v'_{\mathfrak{f}'(i,\,\kappa,x)}|}{2^{q'_{i,\,\kappa,x}}} \right\rfloor 2^{q'_{i,\,\kappa,x}} && \text{(5.3a)} \\
&= \operatorname{sign}(v'_{\mathfrak{f}'(i,\,\kappa,x)}) \left\lfloor \frac{Q_{2^{j_{\mathfrak{f}'(i,\,\kappa,x)}}}(v_{\mathfrak{f}'(i,\,\kappa,x)}) + u_{\mathfrak{f}'(i,\,\kappa,x)}}{2^{q'_{i,\,\kappa,x}}} \right\rfloor 2^{q'_{i,\,\kappa,x}} && \text{(5.3c)} \\
&= \operatorname{sign}(v'_{\mathfrak{f}'(i,\,\kappa,x)}) \left\lfloor \frac{\left\lfloor \frac{|v_{\mathfrak{f}'(i,\,\kappa,x)}|}{2^{j_{\mathfrak{f}'(i,\,\kappa,x)}}} \right\rfloor 2^{j_{\mathfrak{f}'(i,\,\kappa,x)}} + u_{\mathfrak{f}'(i,\,\kappa,x)}}{2^{q'_{i,\,\kappa,x}}} \right\rfloor 2^{q'_{i,\,\kappa,x}} && \text{(5.3a)} \\
&= \operatorname{sign}(v'_{\mathfrak{f}'(i,\,\kappa,x)}) \left\lfloor \frac{\left\lfloor \frac{|v_{\mathfrak{f}'(i,\,\kappa,x)}|}{2^{j_{\mathfrak{f}'(i,\,\kappa,x)}}} \right\rfloor + \frac{u_{\mathfrak{f}'(i,\,\kappa,x)}}{2^{j_{\mathfrak{f}'(i,\,\kappa,x)}}}}{2^{q'_{i,\,\kappa,x} - j_{\mathfrak{f}'(i,\,\kappa,x)}}} \right\rfloor 2^{q'_{i,\,\kappa,x}}.
\end{aligned}$$

Now, $\forall x, \lfloor x \rfloor \in \mathbb{Z}$ so $\left\lfloor \frac{|v_{\mathfrak{f}'(i,\,\kappa,x)}|}{2^{j_{\mathfrak{f}'(i,\,\kappa,x)}}} \right\rfloor \in \mathbb{Z}$. Since $v'_{\mathfrak{f}'(i,\,\kappa,x)} \in V'$, then $v_{\mathfrak{f}'(i,\,\kappa,x)} \in V$ (5.4a) so $j_{\mathfrak{f}'(i,\,\kappa,x)} \in \mathbb{N}$, $0 \leq u_{\mathfrak{f}'(i,\,\kappa,x)} < 2^{j_{\mathfrak{f}'(i,\,\kappa,x)}} \leq |v_{\mathfrak{f}'(i,\,\kappa,x)}|$ (5.5). Lastly, from the definition in section 6.1.3.2 for the quantization step size we have

$$q'_{i,\,\kappa,x} = \begin{cases} \max(M'_{s'_{\mathfrak{f}'(i,\,\kappa,x)}} - (1 + \lfloor \frac{j'_i - 1 - \kappa}{a} \rfloor), j'_{\mathfrak{f}'(i,\,\kappa,x)}) & \\ & \text{if } \Psi_{i^*,\,\kappa^*} = 0 \\ \max(\kappa + 1, j'_{\mathfrak{f}'(i,\,\kappa,x)}) & \\ & \text{if } \Psi_{i^*,\,\kappa^*} = 1 \end{cases} \tag{6.13}$$

where $M'_{s'_{\mathfrak{f}'(i,\,\kappa,x)}}, j'_{\mathfrak{f}'(i,\,\kappa,x)}$ and κ are all integers, so $q'_{i,\,\kappa,x} \in \mathbb{Z}$ and $q'_{i,\,\kappa,x} \geq j'_{\mathfrak{f}'(i,\,\kappa,x)}$, so $2^{q'_{i,\,\kappa,x} - j'_{\mathfrak{f}'(i,\,\kappa,x)}} \in \mathbb{N}$. Thus, applying lemma D.2.2

$$\begin{aligned}
&sign(v'_{\mathfrak{f}'(i,\,\kappa,x)})Q_{2^{q'_{i,\,\kappa,x}}}(v'_{\mathfrak{f}'(i,\,\kappa,x)}) \\
&\quad = \operatorname{sign}(v'_{\mathfrak{f}'(i,\,\kappa,x)}) \left\lfloor \frac{\left\lfloor \frac{|v_{\mathfrak{f}'(i,\,\kappa,x)}|}{2^{j_{\mathfrak{f}'(i,\,\kappa,x)}}} \right\rfloor}{2^{q'_{i,\,\kappa,x} - j_{\mathfrak{f}'(i,\,\kappa,x)}}} \right\rfloor 2^{q'_{i,\,\kappa,x}} \\
&\quad = \operatorname{sign}(v'_{\mathfrak{f}'(i,\,\kappa,x)}) \left\lfloor \frac{\left\lfloor \frac{|v_{\mathfrak{f}'(i,\,\kappa,x)}|}{2^{j_{\mathfrak{f}'(i,\,\kappa,x)}}} \right\rfloor 2^{j}_{\mathfrak{f}'(i,\,\kappa,x)}}{2^{q'_{i,\,\kappa,x}}} \right\rfloor 2^{q'_{i,\,\kappa,x}}
\end{aligned}$$

and, applying lemma D.2.3

$$= \operatorname{sign}(v'_{\mathfrak{f}'(i,\kappa,x)}) \left\lfloor \frac{|v_{\mathfrak{f}'(i,\kappa,x)}|}{2^{q'_{i,\kappa,x}}} \right\rfloor 2^{q'_{i,\kappa,x}}$$

$$= \operatorname{sign}(v'_{\mathfrak{f}'(i,\kappa,x)}) Q_{2^{q'_{i,\kappa,x}}}(v_{\mathfrak{f}'(i,\kappa,x)}) \tag{5.3a}$$

$$= \begin{cases} \operatorname{sign}(v'_{\mathfrak{f}'(i,\kappa,x)}) Q_{2^{q'_{i,\kappa,x}}}(v_{\mathfrak{f}'(i,\kappa,x)}) & \text{if } |v_{\mathfrak{f}'(i,\kappa,x)}| \geq 2^{q'_{i,\kappa,x}} \\ 0 & \text{if } |v_{\mathfrak{f}'(i,\kappa,x)}| < 2^{q'_{i,\kappa,x}} \end{cases}$$

$$= \begin{cases} \operatorname{sign}\left(\operatorname{sign}(v_{\mathfrak{f}'(i,\kappa,x)}) \left(Q_{2^{j_{\mathfrak{f}'(i,\kappa,x)}}}(v_{\mathfrak{f}'(i,\kappa,x)}) + u_{\mathfrak{f}'(i,\kappa,x)}\right)\right) & \text{if } |v_{\mathfrak{f}'(i,\kappa,x)}| \geq 2^{q'_{i,\kappa,x}} \\ 0 & \text{if } |v_{\mathfrak{f}'(i,\kappa,x)}| < 2^{q'_{i,\kappa,x}} \end{cases} \tag{5.3c}$$

but $u_{\mathfrak{f}'(i,\kappa,x)} \geq 0$ and $j_{\mathfrak{f}'(i,\kappa,x)} \leq q'_{i,\kappa,x}$, so $Q_{2^{j_{\mathfrak{f}'(i,\kappa,x)}}}(v_{\mathfrak{f}'(i,\kappa,x)}) + u_{\mathfrak{f}'(i,\kappa,x)} > 0$ whenever $|v_{\mathfrak{f}'(i,\kappa,x)}| \geq 2^{q'_{i,\kappa,x}}$, so this is simply

$$= \begin{cases} \operatorname{sign}(v_{\mathfrak{f}'(i,\kappa,x)}) Q_{2^{q'_{i,\kappa,x}}}(v_{\mathfrak{f}'(i,\kappa,x)}) & \text{if } |v_{\mathfrak{f}'(i,\kappa,x)}| \geq 2^{q'_{i,\kappa,x}} \\ 0 & \text{if } |v_{\mathfrak{f}'(i,\kappa,x)}| < 2^{q'_{i,\kappa,x}} \end{cases}$$

$$= \operatorname{sign}(v_{\mathfrak{f}'(i,\kappa,x)}) Q_{2^{q'_{i,\kappa,x}}}(v_{\mathfrak{f}'(i,\kappa,x)})$$

$$= \operatorname{sign}(v_{\mathfrak{f}'(i,\kappa,x)}) Q_{2^{q_{i,\kappa,x}}}(v_{\mathfrak{f}'(i,\kappa,x)}) \tag{6.14a}$$

$$= \operatorname{sign}(v_{\mathfrak{f}(i,\kappa,x)}) Q_{2^{q_{i,\kappa,x}}}(v_{\mathfrak{f}(i,\kappa,x)}) \tag{6.11a}$$

Thus, in both possible cases, $\operatorname{sign}(v'_{\mathfrak{f}'(i,\kappa,x)}) Q_{2^{q'_{i,\kappa,x}}}(v'_{\mathfrak{f}'(i,\kappa,x)}) = \operatorname{sign}(v_{\mathfrak{f}(i,\kappa,x)}) Q_{2^{q_{i,\kappa,x}}}(v_{\mathfrak{f}(i,\kappa,x)})$. Using this and $\Lambda' = \Lambda \implies \eta' = \eta$, it can be easily seen that $\forall x \in \mathbb{Z} 0, \leq x < \eta' = \eta$

$$\mathfrak{S}(v'_{\mathfrak{f}'(i,\kappa,x)}) = \begin{cases} 0 & \text{if } \operatorname{sign}(v'_{\mathfrak{f}'(i,\kappa,x)}) Q_{2^{q'_{i,\kappa,x}}}(v'_{\mathfrak{f}'(i,\kappa,x)})) \geq 0 \\ 1 & \text{if } \operatorname{sign}(v'_{\mathfrak{f}'(i,\kappa,x)}) Q_{2^{q'_{i,\kappa,x}}}(v'_{\mathfrak{f}'(i,\kappa,x)})) < 0 \end{cases} \tag{6.12b}$$

$$= \begin{cases} 0 & \text{if } \operatorname{sign}(v_{\mathfrak{f}(i,\kappa,x)}) Q_{2^{q_{i,\kappa,x}}}(v_{\mathfrak{f}(i,\kappa,x)})) \geq 0 \\ 1 & \text{if } \operatorname{sign}(v_{\mathfrak{f}(i,\kappa,x)}) Q_{2^{q_{i,\kappa,x}}}(v_{\mathfrak{f}(i,\kappa,x)})) < 0 \end{cases}$$

$$= \mathfrak{S}(v_{\mathfrak{f}(i,\kappa,x)})$$

Since $\operatorname{sign}(v'_{\mathfrak{f}'(i,\kappa,x)}) Q_{2^{q'_{i,\kappa,x}}}(v'_{\mathfrak{f}'(i,\kappa,x)}) = \operatorname{sign}(v_{\mathfrak{f}(i,\kappa,x)}) Q_{2^{q_{i,\kappa,x}}}(v_{\mathfrak{f}(i,\kappa,x)})$ and $Q(v) > 0 \, \forall v$, it must be the case that $Q_{2^{q'_{i,\kappa,x}}}(v'_{\mathfrak{f}'(i,\kappa,x)}) = Q_{2^{q_{i,\kappa,x}}}(v_{\mathfrak{f}(i,\kappa,x)})$, and therefore that $\forall x, t \in \mathbb{Z}$, $0 \leq x, t < \eta' = \eta$

$$\mathfrak{M}(v'_{\mathfrak{f}'(i,\kappa,t)}, v'_{\mathfrak{f}'(i,\kappa,x)}) = \begin{cases} 0 & \text{if } Q_{2^{q'_{i,\kappa,t}}}(v'_{\mathfrak{f}'(i,\kappa,t)}) \leq Q_{2^{q'_{i,\kappa,x}}}(v'_{\mathfrak{f}'(i,\kappa,x)}) \\ 1 & \text{if } Q_{2^{q'_{i,\kappa,t}}}(v'_{\mathfrak{f}'(i,\kappa,t)}) > Q_{2^{q'_{i,\kappa,x}}}(v'_{\mathfrak{f}'(i,\kappa,x)}) \end{cases} \tag{6.12c}$$

$$= \begin{cases} 0 & \text{if } Q_{2^{q_{i,\kappa,t}}}(v_{\mathfrak{f}'(i,\kappa,t)}) \leq Q_{2^{q_{i,\kappa,x}}}(v_{\mathfrak{f}(i,\kappa,x)}) \\ 1 & \text{if } Q_{2^{q_{i,\kappa,t}}}(v_{\mathfrak{f}'(i,\kappa,t)}) > Q_{2^{q_{i,\kappa,x}}}(v_{\mathfrak{f}(i,\kappa,x)}) \end{cases}$$

$$= \mathfrak{M}(v_{\mathfrak{f}(i,\kappa,t)}, v_{\mathfrak{f}(i,\kappa,x)}).$$

Thus

$$\begin{aligned}
u^c_{i,\kappa} &= u'_{i,\kappa} \\
&= \bigoplus_{x=0}^{\eta'-1} \mathfrak{S}(v'_{\mathfrak{f}'(i,\kappa,x)}) \oplus \mathfrak{M}(v'_{\mathfrak{f}'(i,\kappa,x-1 \mod \eta')}, v'_{\mathfrak{f}'(i,\kappa,x)}) && (6.1)\\
&= \bigoplus_{x=0}^{\eta'-1} \mathfrak{S}(v_{\mathfrak{f}(i,\kappa,x)}) \oplus \mathfrak{M}(v_{\mathfrak{f}(i,\kappa,x-1 \mod \eta)}, v_{\mathfrak{f}(i,\kappa,x)}) \\
&= \bigoplus_{x=0}^{\eta-1} \mathfrak{S}(v_{\mathfrak{f}(i,\kappa,x)}) \oplus \mathfrak{M}(v_{\mathfrak{f}(i,\kappa,x-1 \mod \eta)}, v_{\mathfrak{f}(i,\kappa,x)}) && \Lambda' = \Lambda \\
&= u_{i,\kappa}
\end{aligned}$$

and property (6.17a) is proven.

2. Resolution Scaled

If $I^{\mathcal{R}} = \mathcal{R}(\mathsf{Embed}(I, \Lambda))$ is a resolution scaled watermarked image for which the Φ most significant bit planes of the lowest resolution layer have not been lost, $\Lambda^{\mathcal{R}} = \Lambda$, and $v_i^{\mathcal{R}} \in V^{\mathcal{R}}$ is the coefficient corresponding to $v'_i \in V'$ and $v_i \in V$ then, if the candidate watermark bit exists $\exists u^c_{i,\kappa}$, then from the definition of u^c,

$$u^c_{i,\kappa} = u^{\mathcal{R}}_{i,\kappa} \text{ and } \exists v^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\kappa,x)} \wedge m^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\kappa,x)} \leq q^{\mathcal{R}}_{i,\kappa,x},\ \forall x^{\mathcal{R}} \in \mathbb{Z} : 0 \leq x^{\mathcal{R}} < \eta^{\mathcal{R}}$$

we know that $\forall x \in \mathbb{Z},\ 0 \leq x < \eta^{\mathcal{R}}$, $v^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\kappa,x)}$ exists in the image and is therefore unaffected by resolution scaling so

$$\begin{aligned}
v^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\kappa,x)} &= v'_{\mathfrak{f}^{\mathcal{R}}(i,\kappa,x)} && (5.6b)\\
&= v'_{\mathfrak{f}'(i,\kappa,x)} && (5.13b).
\end{aligned}$$

Thus $\forall x \in \mathbb{Z},\ 0 \leq x < \eta^{\mathcal{R}} = \eta$

$$\begin{aligned}
\operatorname{sign}(v^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\kappa,x)}) Q_{2^{q^{\mathcal{R}}_{i,\kappa,x}}}(v^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\kappa,x)}) &= \operatorname{sign}(v'_{\mathfrak{f}'(i,\kappa,x)}) Q_{2^{q^{\mathcal{R}}_{i,\kappa,x}}}(v'_{\mathfrak{f}'(i,\kappa,x)}) \\
&= \operatorname{sign}(v'_{\mathfrak{f}'(i,\kappa,x)}) Q_{2^{q'_{i,\kappa,x}}}(v'_{\mathfrak{f}'(i,\kappa,x)}) && (6.14b)
\end{aligned}$$

which, from **1. Unscaled** proof,

$$= \operatorname{sign}(v_{\mathfrak{f}(i,\kappa,x)}) Q_{2^{q_{i,\kappa,x}}}(v_{\mathfrak{f}(i,\kappa,x)}).$$

Hence, $\forall x \in \mathbb{Z} 0 \leq x < \eta^{\mathcal{R}} = \eta$

$$\begin{aligned}\mathfrak{S}(v^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\kappa,x)}) &= \begin{cases} 0 & \text{if } \operatorname{sign}(v^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\kappa,x)}) Q_{2^{q^{\mathcal{R}}_{i,\kappa,x}}}(v^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\kappa,x)})) \geq 0 \\ 1 & \text{if } \operatorname{sign}(v^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\kappa,x)}) Q_{2^{q^{\mathcal{R}}_{i,\kappa,x}}}(v^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\kappa,x)})) < 0 \end{cases} \\ &= \begin{cases} 0 & \text{if } \operatorname{sign}(v_{\mathfrak{f}(i,\kappa,x)}) Q_{2^{q_{i,\kappa,x}}}(v_{\mathfrak{f}(i,\kappa,x)})) \geq 0 \\ 1 & \text{if } \operatorname{sign}(v_{\mathfrak{f}(i,\kappa,x)}) Q_{2^{q_{i,\kappa,x}}}(v_{\mathfrak{f}(i,\kappa,x)})) < 0 \end{cases} \\ &= \mathfrak{S}(v_{\mathfrak{f}(i,\kappa,x)}) \end{aligned} \tag{6.12b}$$

and $\forall x, t \in \mathbb{Z} 0 \leq x, t < \eta^{\mathcal{R}} = \eta$

$$\begin{aligned}\mathfrak{M}(v^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\kappa,t)}, v^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\kappa,x)}) &= \begin{cases} 0 & \text{if } Q_{2^{q^{\mathcal{R}}_{i,\kappa,t}}}(v^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\kappa,t)}) \leq Q_{2^{q^{\mathcal{R}}_{i,\kappa,x}}}(v^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\kappa,x)}) \\ 1 & \text{if } Q_{2^{q^{\mathcal{R}}_{i,\kappa,t}}}(v^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\kappa,t)}) > Q_{2^{q^{\mathcal{R}}_{i,\kappa,x}}}(v^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\kappa,x)}) \end{cases} \\ &= \begin{cases} 0 & \text{if } Q_{2^{q_{i,\kappa,t}}}(v_{\mathfrak{f}^{\mathcal{R}}(i,\kappa,t)}) \leq Q_{2^{q_{i,\kappa,x}}}(v_{\mathfrak{f}(i,\kappa,x)}) \\ 1 & \text{if } Q_{2^{q_{i,\kappa,t}}}(v_{\mathfrak{f}^{\mathcal{R}}(i,\kappa,t)}) > Q_{2^{q_{i,\kappa,x}}}(v_{\mathfrak{f}(i,\kappa,x)}) \end{cases} \\ &= \mathfrak{M}(v_{\mathfrak{f}(i,\kappa,t)}, v_{\mathfrak{f}(i,\kappa,x)}). \end{aligned} \tag{6.12c}$$

Thus

$$\begin{aligned} u^{c}_{i,\kappa} &= u^{\mathcal{R}}_{i,\kappa} \\ &= \bigoplus_{x=0}^{\eta^{\mathcal{R}}-1} \mathfrak{S}(v^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\kappa,x)}) \oplus \mathfrak{M}(v^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\kappa,x-1 \mod \eta^{\mathcal{R}})}, v^{\mathcal{R}}_{\mathfrak{f}^{\mathcal{R}}(i,\kappa,x)}) && (6.1) \\ &= \bigoplus_{x=0}^{\eta^{\mathcal{R}}-1} \mathfrak{S}(v_{\mathfrak{f}(i,\kappa,x)}) \oplus \mathfrak{M}(v_{\mathfrak{f}(i,\kappa,x-1 \mod \eta)}, v_{\mathfrak{f}(i,\kappa,x)}) \\ &= \bigoplus_{x=0}^{\eta-1} \mathfrak{S}(v_{\mathfrak{f}(i,\kappa,x)}) \oplus \mathfrak{M}(v_{\mathfrak{f}(i,\kappa,x-1 \mod \eta)}, v_{\mathfrak{f}(i,\kappa,x)}) && \Lambda^{\mathcal{R}} = \Lambda \\ &= u_{i,\kappa} \end{aligned}$$

and property (6.17b) is proven.

3. Quality Scaled

If $I^{\mathcal{Q}} = \mathcal{Q}(\mathsf{Embed}(I, \Lambda))$ is a quality scaled watermarked image for which the Φ most significant bit planes of the lowest resolution layer have not been lost, $\Lambda^{\mathcal{Q}} = \Lambda$, and $v^{\mathcal{Q}}_i \in V^{\mathcal{Q}}$ is the coefficient corresponding to $v'_i \in V'$ and $v_i \in V$ then, if the candidate watermark exists $\exists u^{c}_{i,\kappa}$, then, from the definition of u^c,

$$u^{c}_{i,\kappa} = u^{\mathcal{Q}}_{i,\kappa}, \text{ and } \exists v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)} \wedge m^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)} \leq q^{\mathcal{Q}}_{i,\kappa,x}, \forall x \in \mathbb{Z} : 0 \leq x < \eta^{\mathcal{Q}}. \tag{6.16}$$

Consider the value $\mathrm{sign}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)})Q_{2^{q^{\mathcal{Q}}_{i,\kappa,x}}}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)})$ used in the definition of $\mathfrak{S}$:

$$
\begin{aligned}
&\mathrm{sign}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)})Q_{2^{q^{\mathcal{Q}}_{i,\kappa,x}}}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)})\\
&\quad= \mathrm{sign}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)})\left\lfloor\frac{|v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}|}{2^{q^{\mathcal{Q}}_{i,\kappa,x}}}\right\rfloor 2^{q^{\mathcal{Q}}_{i,\kappa,x}} && \text{(5.3a)}\\
&\quad= \mathrm{sign}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)})\left\lfloor\frac{\left\lfloor\frac{|v'_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}|}{2^{m_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}}}\right\rfloor 2^{m_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}}+\left\lfloor 2^{m_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}}\mathrm{r}\right\rfloor}{2^{q^{\mathcal{Q}}_{i,\kappa,x}}}\right\rfloor 2^{q^{\mathcal{Q}}_{i,\kappa,x}} && \text{(2.20a)}\\
&\quad= \mathrm{sign}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)})\left\lfloor\frac{\left\lfloor\frac{|v'_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}|}{2^{m_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}}}\right\rfloor+\frac{\left\lfloor 2^{m_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}}\mathrm{r}\right\rfloor}{2^{m_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}}}}{2^{q^{\mathcal{Q}}_{i,\kappa,x}-m_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}}}\right\rfloor 2^{q^{\mathcal{Q}}_{i,\kappa,x}}
\end{aligned}
$$

but $\left\lfloor\frac{|v'_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}|}{2^{m_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}}}\right\rfloor\in\mathbb{Z}$, and $0\leq\mathrm{r}<1$ (2.20b) so $0\leq\frac{\left\lfloor 2^{m_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}}\mathrm{r}\right\rfloor}{2^{m_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}}}<1$ and $m_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}, q^{\mathcal{Q}}_{i,\kappa,x}\in\mathbb{Z}, m_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}\leq q^{\mathcal{Q}}_{i,\kappa,x}$ so $2^{q^{\mathcal{Q}}_{i,\kappa,x}-m_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}}\in\mathbb{N}$ thus

$$
\begin{aligned}
&\mathit{sign}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)})Q_{2^{q^{\mathcal{Q}}_{i,\kappa,x}}}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)})\\
&\quad= \mathrm{sign}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)})\left\lfloor\frac{\left\lfloor\frac{|v'_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}|}{2^{m_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}}}\right\rfloor}{2^{q^{\mathcal{Q}}_{i,\kappa,x}-m_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}}}\right\rfloor 2^{q^{\mathcal{Q}}_{i,\kappa,x}} && \text{Lemma D.2.2}\\
&\quad= \mathrm{sign}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)})\left\lfloor\frac{\left\lfloor\frac{|v'_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}|}{2^{m_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}}}\right\rfloor 2^{m_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}}}{2^{q^{\mathcal{Q}}_{i,\kappa,x}}}\right\rfloor 2^{q^{\mathcal{Q}}_{i,\kappa,x}}\\
&\quad= \mathrm{sign}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)})\left\lfloor\frac{|v'_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}|}{2^{q^{\mathcal{Q}}_{i,\kappa,x}}}\right\rfloor 2^{q^{\mathcal{Q}}_{i,\kappa,x}} && \text{Lemma D.2.3}\\
&\quad= \mathrm{sign}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)})Q_{2^{q^{\mathcal{Q}}_{i,\kappa,x}}}(v'_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}) && \text{(5.3a)}\\
&\quad= \begin{cases}\mathrm{sign}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)})Q_{2^{q^{\mathcal{Q}}_{i,\kappa,x}}}(v'_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}) & \text{if } |v'_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}|\geq 2^{q^{\mathcal{Q}}_{i,\kappa,x}}\\ 0 & \text{if } |v'_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}|< 2^{q^{\mathcal{Q}}_{i,\kappa,x}}\end{cases}\\
&\quad= \begin{cases}\mathrm{sign}\left(\mathrm{sign}(v'_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)})\left(Q_{2^{m_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}}}(v'_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)})+\left\lfloor 2^{m_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}}\mathrm{r}\right\rfloor\right)\right)Q_{2^{q^{\mathcal{Q}}_{i,\kappa,x}}}(v'_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}) & \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{if } |v'_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}|\geq 2^{q^{\mathcal{Q}}_{i,\kappa,x}}\\ 0 \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{if } |v'_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}|< 2^{q^{\mathcal{Q}}_{i,\kappa,x}}\end{cases}
\end{aligned}
$$

but, because $u^c_{i,\kappa}$ exists, $m^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)} \le q^{\mathcal{Q}}_{i,\kappa,x}$ thus, since $m^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)} = m_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}$ for a JPEG2000 scaled but otherwise untampered image, we know that $m_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)} \le q^{\mathcal{Q}}_{i,\kappa,x}$ also, because $m_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)} \ge 0$ and $0 \le \mathrm{r} \le 1$, we know that $2^{m^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}}\mathrm{r} \ge 0$, and therefore that $Q_{2^{m^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}}}(v'_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}) + \left\lfloor 2^{m^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}}\mathrm{r} \right\rfloor > 0$ whenever $|v'_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}| \ge 2^{q^{\mathcal{Q}}_{i,\kappa,x}}$, so this is simply

$$= \begin{cases} \operatorname{sign}(v'_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}) Q_{2^{q^{\mathcal{Q}}_{i,\kappa,x}}}(v'_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}) & \text{if } |v'_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}| \ge 2^{q^{\mathcal{Q}}_{i,\kappa,x}} \\ 0 & \text{if } |v'_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}| < 2^{q^{\mathcal{Q}}_{i,\kappa,x}} \end{cases}$$

If $v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)} \ne 0$, then by (6.14c) and (6.14a) we have that $q^{\mathcal{Q}}_{i,\kappa,x} = q_{i,\kappa,x} = q'_{i,\kappa,x}$ so

$$\begin{aligned} &= \begin{cases} \operatorname{sign}(v'_{\mathfrak{f}'(i,\kappa,x)}) Q_{2^{q'_{i,\kappa,x}}}(v'_{\mathfrak{f}'(i,\kappa,x)}) & \text{if } |v'_{\mathfrak{f}'(i,\kappa,x)}| \ge 2^{q'_{i,\kappa,x}} \\ 0 & \text{if } |v'_{\mathfrak{f}'(i,\kappa,x)}| < 2^{q'_{i,\kappa,x}} \end{cases} \\ &= \begin{cases} \operatorname{sign}(v'_{\mathfrak{f}'(i,\kappa,x)}) Q_{2^{q'_{i,\kappa,x}}}(v'_{\mathfrak{f}'(i,\kappa,x)}) & \text{if } |v'_{\mathfrak{f}'(i,\kappa,x)}| \ge 2^{q'_{i,\kappa,x}} \\ 0 & \text{if } |v'_{\mathfrak{f}'(i,\kappa,x)}| < 2^{q'_{i,\kappa,x}} \end{cases} \\ &= \operatorname{sign}(v'_{\mathfrak{f}'(i,\kappa,x)}) Q_{2^{q'_{i,\kappa,x}}}(v'_{\mathfrak{f}'(i,\kappa,x)}) \\ &= \operatorname{sign}(v_{\mathfrak{f}(i,\kappa,x)}) Q_{2^{q_{i,\kappa,x}}}(v_{\mathfrak{f}(i,\kappa,x)}), \end{aligned} \tag{5.13c}$$

that is,

$$\operatorname{sign}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}) Q_{2^{q^{\mathcal{Q}}_{i,\kappa,x}}}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}) = \operatorname{sign}(v_{\mathfrak{f}(i,\kappa,x)}) Q_{2^{q_{i,\kappa,x}}}(v_{\mathfrak{f}(i,\kappa,x)}).$$

If, on the other hand, $v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)} = 0$ we consider the two subcases: $|v'_{\mathfrak{f}'(i,\kappa,x)}| < 2^{q'_{i,\kappa,x}}$ and $|v'_{\mathfrak{f}'(i,\kappa,x)}| \ge 2^{q'_{i,\kappa,x}}$. In the first subcase, $|v'_{\mathfrak{f}'(i,\kappa,x)}| < 2^{q'_{i,\kappa,x}}$ and $v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)} = 0$ so

$$\begin{aligned} \operatorname{sign}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}) Q_{2^{q^{\mathcal{Q}}_{i,\kappa,x}}}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}) &= \operatorname{sign}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}) \left\lfloor \frac{|v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}|}{2^{q^{\mathcal{Q}}_{i,\kappa,x}}} \right\rfloor 2^{q^{\mathcal{Q}}_{i,\kappa,x}} && \text{defn. } Q \\ &= 0 && v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)} = 0 \end{aligned}$$

and

$$\begin{aligned} \operatorname{sign}(v'_{\mathfrak{f}'(i,\kappa,x)}) Q_{2^{q'_{i,\kappa,x}}}(v'_{\mathfrak{f}'(i,\kappa,x)}) &= \operatorname{sign}(v'_{\mathfrak{f}'(i,\kappa,x)}) \left\lfloor \frac{|v'_{\mathfrak{f}'(i,\kappa,x)}|}{2^{q'_{i,\kappa,x}}} \right\rfloor 2^{q'_{i,\kappa,x}} && \text{defn. } Q \\ &= 0 && |v'_{\mathfrak{f}'(i,\kappa,x)}| < 2^{q'_{i,\kappa,x}} \end{aligned}$$

and we again obtain

$$\begin{aligned} \operatorname{sign}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}) Q_{2^{q^{\mathcal{Q}}_{i,\kappa,x}}}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}) &= \operatorname{sign}(v'_{\mathfrak{f}'(i,\kappa,x)}) Q_{2^{q'_{i,\kappa,x}}}(v'_{\mathfrak{f}'(i,\kappa,x)}) \\ &= \operatorname{sign}(v_{\mathfrak{f}(i,\kappa,x)}) Q_{2^{q_{i,\kappa,x}}}(v_{\mathfrak{f}(i,\kappa,x)}). \end{aligned}$$

In the second case, $|v'_{\mathfrak{f}'(i,\kappa,x)}| \geq 2^{q'_{i,\kappa,x}}$ and $v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)} = 0$ so more than $q'_{i,\kappa,x}$ bits must have been lost during scaling

$$m^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)} > q'_{i,\kappa,x}$$

and, because $\exists u^{c}_{i,\kappa}$ we have

$$m^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)} \leq q^{\mathcal{Q}}_{i,\kappa,x}$$

so

$$q^{\mathcal{Q}}_{i,\kappa,x} > q'_{i,\kappa,x}.$$

But the definition of $q'_{i,\kappa,x}$ gives us

$$q'_{i,\kappa,x} = \begin{cases} \max(M'_{s'_{\mathfrak{f}'(i,\kappa,x)}} - (1 + \lfloor \frac{j'_i - 1 - \kappa}{a'} \rfloor), j'_{\mathfrak{f}'(i,\kappa,x)}) & \text{if } \Psi'_{i,\kappa} = 0 \\ \max(\kappa + 1, j'_{\mathfrak{f}'(i,\kappa,x)}) & \text{if } \Psi'_{i,\kappa} = 1 \end{cases} \tag{6.13}$$

which, since, from equations (6.4c) and (6.4a), $j^{\mathcal{Q}}_i = j_i = j'_i$ and, from the proof of (6.11f) in section E.1.2, $\Psi^{\mathcal{Q}}_{i,\kappa} = \Psi'_{i,\kappa}$ and, because quality scaling does not affect the image header information, $M^{\mathcal{Q}}_{s^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}} = M'_{s'_{\mathfrak{f}'(i,\kappa,x)}}$, becomes

$$= \begin{cases} \max(M^{\mathcal{Q}}_{s^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}} - (1 + \lfloor \frac{j^{\mathcal{Q}}_i - 1 - \kappa}{a'} \rfloor), j'_{\mathfrak{f}'(i,\kappa,x)}) & \text{if } \Psi^{\mathcal{Q}}_{i,\kappa} = 0 \\ \max(\kappa + 1, j'_{\mathfrak{f}'(i,\kappa,x)}) & \text{if } \Psi^{\mathcal{Q}}_{i,\kappa} = 1. \end{cases}$$

Because $v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)} = 0$ so $v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)} \notin V^{\mathcal{Q}}$, from equation (6.3) we have $j^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)} = 0$ and therefore $j'_{\mathfrak{f}'(i,\kappa,x)} > j^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}$. So this is

$$\begin{aligned} &\geq \begin{cases} \max(M^{\mathcal{Q}}_{s^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}} - (1 + \lfloor \frac{j^{\mathcal{Q}}_i - 1 - \kappa}{a'} \rfloor), j^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}) & \text{if } \Psi^{\mathcal{Q}}_{i,\kappa} = 0 \\ \max(\kappa + 1, j^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}) & \text{if } \Psi^{\mathcal{Q}}_{i,\kappa} = 1 \end{cases} \\ &= q^{\mathcal{Q}}_{i,\kappa,x} \end{aligned}$$

which contradicts $q^{\mathcal{Q}}_{i,\kappa,x} > q'_{i,\kappa,x}$, so it is not possible to the case $|v'_{\mathfrak{f}'(i,\kappa,x)}| \geq 2^{q'_{i,\kappa,x}}$ and $v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)} = 0$ given that the candidate watermark exists. Thus for all *possible* cases, $\exists u^{c}_{i,\kappa}$ implies

$$\operatorname{sign}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}) Q_{2^{q^{\mathcal{Q}}_{i,\kappa,x}}}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}) = \operatorname{sign}(v_{\mathfrak{f}(i,\kappa,x)}) Q_{2^{q_{i,\kappa,x}}}(v_{\mathfrak{f}(i,\kappa,x)}).$$

Hence, $\forall x \in \mathbb{Z}, 0 \leq x < \eta^{\mathcal{Q}} = \eta$,

$$\begin{aligned}
\mathfrak{S}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}) &= \begin{cases} 0 & \text{if } \operatorname{sign}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}) Q_{2^{q^{\mathcal{Q}}_{i,\kappa,x}}}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)})) \geq 0 \\ 1 & \text{if } \operatorname{sign}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}) Q_{2^{q^{\mathcal{Q}}_{i,\kappa,x}}}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)})) < 0 \end{cases} \\
&= \begin{cases} 0 & \text{if } \operatorname{sign}(v_{\mathfrak{f}(i,\kappa,x)}) Q_{2^{q_{i,\kappa,x}}}(v_{\mathfrak{f}(i,\kappa,x)})) \geq 0 \\ 1 & \text{if } \operatorname{sign}(v_{\mathfrak{f}(i,\kappa,x)}) Q_{2^{q_{i,\kappa,x}}}(v_{\mathfrak{f}(i,\kappa,x)})) < 0 \end{cases} \\
&= \mathfrak{S}(v_{\mathfrak{f}(i,\kappa,x)})
\end{aligned}$$

and $\forall x, t \in \mathbb{Z}, 0 \leq x, t < \eta^{\mathcal{Q}} = \eta$

$$\begin{aligned}
\mathfrak{M}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,t)}, v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}) &= \begin{cases} 0 & \text{if } Q_{2^{q^{\mathcal{Q}}_{i,\kappa,t}}}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,t)}) \leq Q_{2^{q^{\mathcal{Q}}_{i,\kappa,x}}}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}) \\ 1 & \text{if } Q_{2^{q^{\mathcal{Q}}_{i,\kappa,t}}}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,t)}) > Q_{2^{q^{\mathcal{Q}}_{i,\kappa,x}}}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}) \end{cases} \\
&= \begin{cases} 0 & \text{if } Q_{2^{q_{i,\kappa,t}}}(v_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,t)}) \leq Q_{2^{q_{i,\kappa,x}}}(v_{\mathfrak{f}(i,\kappa,x)}) \\ 1 & \text{if } Q_{2^{q_{i,\kappa,t}}}(v_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,t)}) > Q_{2^{q_{i,\kappa,x}}}(v_{\mathfrak{f}(i,\kappa,x)}) \end{cases} \\
&= \mathfrak{M}(v_{\mathfrak{f}(i,\kappa,t)}, v_{\mathfrak{f}(i,\kappa,x)}).
\end{aligned}$$

Thus

$$\begin{aligned}
u^{c}_{i,\kappa} &= u^{\mathcal{Q}}_{i,\kappa} \\
&= \bigoplus_{x=0}^{\eta^{\mathcal{Q}}-1} \mathfrak{S}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}) \oplus \mathfrak{M}(v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x-1 \mod \eta^{\mathcal{Q}})}, v^{\mathcal{Q}}_{\mathfrak{f}^{\mathcal{Q}}(i,\kappa,x)}) && \text{defn. } u^{*}_{i^*,\kappa^*} \\
&= \bigoplus_{x=0}^{\eta^{\mathcal{Q}}-1} \mathfrak{S}(v_{\mathfrak{f}(i,\kappa,x)}) \oplus \mathfrak{M}(v_{\mathfrak{f}(i,\kappa,x-1 \mod \eta)}, v_{\mathfrak{f}(i,\kappa,x)}) && 0 \leq x < \eta^{\mathcal{Q}} = \eta \\
&= \bigoplus_{x=0}^{\eta-1} \mathfrak{S}(v_{\mathfrak{f}(i,\kappa,x)}) \oplus \mathfrak{M}(v_{\mathfrak{f}(i,\kappa,x-1 \mod \eta)}, v_{\mathfrak{f}(i,\kappa,x)}) && \eta^{\mathcal{Q}} = \eta \\
&= u_{i,\kappa}
\end{aligned}$$

and property (6.17c) is proven.

E.1.5 Proof for Watermark Extraction

We show that the watermark extraction procedure (section 6.1.6) ensures that, given the correct watermarking parameters and a potentially scaled but otherwise untampered watermarked image, of which at least the Φ most significant bit planes of the lowest resolution layer have not been lost, if an extracted watermark bit exists then it will be identical to the corresponding embedded watermark bit.

- for the unscaled watermarked image, where $v'_i \in V'$ is the coefficient corresponding to $v_i \in V$

$$\text{if } \exists u^d_{i,\kappa} \text{ then } u^d_{i,\kappa} = u_{i,\kappa} \tag{6.19a}$$

- for the resolution scaled watermarked image, where $v^{\mathcal{R}}_i \in V^{\mathcal{R}}$ is the coefficient corresponding to $v'_i \in V'$

$$\text{if } \exists u^d_{i,\kappa} \text{ then } u^d_{i,\kappa} = u_{i,\kappa} \tag{6.19b}$$

- for the quality scaled watermarked image, where $v^{\mathcal{Q}}_i \in V^{\mathcal{Q}}$ is the coefficient corresponding to $v'_i \in V'$

$$\text{if } \exists u^d_{i,\kappa} \text{ then } u^d_{i,\kappa} = u_{i,\kappa} \tag{6.19c}$$

Let the extracted watermark bit be as defined in section 6.1.6

$$u^d_{i^*,\kappa} = \begin{cases} \left\lfloor \frac{|v^*_{i^*}| - Q_{2^{\kappa+1}}(v^*_{i^*}) - \left\lfloor 2^{m^*_{i^*}} \mathrm{r} \right\rfloor}{2^\kappa} \right\rfloor & \text{if } m^*_{i^*} \le \kappa < j^*_{i^*} \\ \nexists & \text{if } \kappa < m^*_{i^*} \text{ or } \kappa \ge j^*_{i^*}. \end{cases} \tag{6.18}$$

1. Unscaled

If $I' = \mathsf{Embed}(I, \Lambda)$ is the unscaled watermarked image, $\Lambda' = \Lambda$, and $v'_i \in V'$ is the coefficient corresponding to $v_i \in V$, then if $\exists u^d_{i,\kappa}$, then from the definition of u^d (6.18) we have that $m'_i \le \kappa < j'_i$ and

$$u^d_{i,\kappa} = \left\lfloor \frac{|v'_i| - Q_{2^{\kappa+1}}(v'_i) - \left\lfloor 2^{m'_i} \mathrm{r} \right\rfloor}{2^\kappa} \right\rfloor$$

since I' is unscaled, no bits are missing $m'_i = 0$ and from equation (6.4a) $j'_i = j_i$, thus $0 \le \kappa < j_i$ and

$$\begin{aligned}
&= \left\lfloor \frac{|v'_i| - Q_{2^{\kappa+1}}(v'_i) - \lfloor \mathrm{r} \rfloor}{2^\kappa} \right\rfloor \\
&= \left\lfloor \frac{|v'_i| - Q_{2^{\kappa+1}}(v'_i)}{2^\kappa} \right\rfloor && \text{(2.20b)} \\
&= \left\lfloor \frac{|v'_i| - \left\lfloor \frac{|v'_i|}{2^{\kappa+1}} \right\rfloor 2^{\kappa+1}}{2^\kappa} \right\rfloor && \text{(5.3a)} \\
&= \left\lfloor \frac{Q_{2^{j_i}}(v_i) + u_i - \left\lfloor \frac{Q_{2^{j_i}}(v_i) + u_i}{2^{\kappa+1}} \right\rfloor 2^{\kappa+1}}{2^\kappa} \right\rfloor && \text{(5.3c)} \\
&= \left\lfloor \frac{\left\lfloor \frac{|v_i|}{2^{j_i}} \right\rfloor 2^{j_i} + u_i - \left\lfloor \frac{\left\lfloor \frac{|v_i|}{2^{j_i}} \right\rfloor 2^{j_i} + u_i}{2^{\kappa+1}} \right\rfloor 2^{\kappa+1}}{2^\kappa} \right\rfloor && \text{(5.3a)}
\end{aligned}$$

$$
\begin{aligned}
&= \left\lfloor \frac{\left\lfloor \frac{|v_i|}{2^{j_i}} \right\rfloor 2^{j_i} + u_i - \left\lfloor \left\lfloor \frac{|v_i|}{2^{j_i}} \right\rfloor 2^{j_i-(\kappa+1)} + \frac{u_i}{2^{\kappa+1}} \right\rfloor 2^{\kappa+1}}{2^{\kappa}} \right\rfloor \\
&= \left\lfloor \frac{\left\lfloor \frac{|v_i|}{2^{j_i}} \right\rfloor 2^{j_i} + u_i - \left\lfloor \frac{|v_i|}{2^{j_i}} \right\rfloor 2^{j_i} - \left\lfloor \frac{u_i}{2^{\kappa+1}} \right\rfloor 2^{\kappa+1}}{2^{\kappa}} \right\rfloor && \kappa < j_i \implies \left\lfloor \frac{|v_i|}{2^{j_i}} \right\rfloor 2^{j_i-(\kappa+1)} \in Z \\
&= \left\lfloor \frac{u_i - \left\lfloor \frac{u_i}{2^{\kappa+1}} \right\rfloor 2^{\kappa+1}}{2^{\kappa}} \right\rfloor \\
&= \left\lfloor \frac{\sum_{x=0}^{j_i-1} 2^x u_{i,x} - \left\lfloor \frac{\sum_{x=0}^{j_i-1} 2^x u_{i,x}}{2^{\kappa+1}} \right\rfloor 2^{\kappa+1}}{2^{\kappa}} \right\rfloor && (6.1) \\
&= \left\lfloor \frac{\sum_{x=0}^{j_i-1} 2^x u_{i,x} - \sum_{x=\kappa+1}^{j_i-1} 2^x u_{i,x}}{2^{\kappa}} \right\rfloor && u_{i,x} \in \{0,1\} \\
&= \left\lfloor \frac{\sum_{x=0}^{\kappa} 2^x u_{i,x}}{2^{\kappa}} \right\rfloor \\
&= u_{i,\kappa} + \left\lfloor \frac{\sum_{x=0}^{\kappa-1} 2^x u_{i,x}}{2^{\kappa}} \right\rfloor && u_{i,\kappa} \in \{0,1\} \\
&= u_{i,\kappa} && 0 \le \sum_{x=0}^{\kappa-1} 2^x u_{i,x} < 2^{\kappa}.
\end{aligned}
$$

So property (6.19a) is proven.

2. Resolution Scaled

If $I^{\mathcal{R}} = \mathcal{R}(\mathsf{Embed}(I, \Lambda))$ is a resolution scaled watermarked image, $\Lambda^{\mathcal{R}} = \Lambda$, and $v_i^{\mathcal{R}} \in V^{\mathcal{R}}$ is the coefficient corresponding to $v_i' \in V'$ and $v_i \in V$ then, if $\exists u_{i,\kappa}^d$ then from the definition of u^d (6.18) we have that $m_i^{\mathcal{R}} \le \kappa < j_i^{\mathcal{R}}$ and

$$u_{i,\kappa}^d = \left\lfloor \frac{|v_i^{\mathcal{R}}| - Q_{2^{\kappa+1}}(v_i^{\mathcal{R}}) - \left\lfloor 2^{m_i^{\mathcal{R}}} \mathrm{r} \right\rfloor}{2^{\kappa}} \right\rfloor .$$

Now $v_i^{\mathcal{R}} \in V^{\mathcal{R}} \implies v^{\mathcal{R}} \in I^{\mathcal{R}}$ (5.4b) and thus $v_i^{\mathcal{R}} = v_i'$ (5.6b) and, since $I^{\mathcal{R}}$ is only resolution scaled, no bits are missing so $m_i^{\mathcal{R}} = 0$ and, from equation (6.4b), $j_i^{\mathcal{R}} = j_i$, thus $0 \le \kappa < j_i$ and

$$
\begin{aligned}
&= \left\lfloor \frac{|v_i'| - Q_{2^{\kappa+1}}(v_i') - \lfloor \mathrm{r} \rfloor}{2^{\kappa}} \right\rfloor \\
&= u_{i,\kappa} && \text{from the } \textbf{1. Unscaled} \text{ proof}
\end{aligned}
$$

So property (6.19b) is proven.

3. Quality Scaled

If $I^{\mathcal{Q}} = \mathcal{Q}(\mathsf{Embed}(I, \Lambda))$ is a quality scaled watermarked image for which the Φ most significant bit planes of the lowest resolution layer have not been lost, $\Lambda^{\mathcal{Q}} = \Lambda$, and $v_i^{\mathcal{Q}} \in V^{\mathcal{Q}}$ is the coefficient corresponding to $v_i' \in V'$ and $v_i \in V$ then, if $\exists u_{i,\kappa}^c$ then from the definition of u^d (6.18) we have that $m_i^{\mathcal{Q}} \leq \kappa < j_i^{\mathcal{Q}}$ and

$$
\begin{aligned}
u_{i,\kappa}^d &= \left\lfloor \frac{|v_i^{\mathcal{Q}}| - Q_{2^{\kappa+1}}(v_i^{\mathcal{Q}}) - \left\lfloor 2^{m_i^{\mathcal{Q}}} \mathrm{r} \right\rfloor}{2^\kappa} \right\rfloor \\
&= \left\lfloor \frac{|v_i^{\mathcal{Q}}| - \left\lfloor \frac{|v_i^{\mathcal{Q}}|}{2^{\kappa+1}} \right\rfloor 2^{\kappa+1} - \left\lfloor 2^{m_i^{\mathcal{Q}}} \mathrm{r} \right\rfloor}{2^\kappa} \right\rfloor && \text{(5.3a)} \\
&= \left\lfloor \frac{\left\lfloor \frac{|v_i'|}{2^{m_i^{\mathcal{Q}}}} \right\rfloor 2^{m_i^{\mathcal{Q}}} + \left\lfloor 2^{m_i^{\mathcal{Q}}} \mathrm{r} \right\rfloor - \left\lfloor \frac{\left\lfloor \frac{|v_i'|}{2^{m_i^{\mathcal{Q}}}} \right\rfloor 2^{m_i^{\mathcal{Q}}} + \left\lfloor 2^{m_i^{\mathcal{Q}}} \mathrm{r} \right\rfloor}{2^{\kappa+1}} \right\rfloor 2^{\kappa+1} - \left\lfloor 2^{m_i^{\mathcal{Q}}} \mathrm{r} \right\rfloor}{2^\kappa} \right\rfloor && \text{(2.20a)} \\
&= \left\lfloor \frac{\left\lfloor \frac{|v_i'|}{2^{m_i^{\mathcal{Q}}}} \right\rfloor 2^{m_i^{\mathcal{Q}}} - \left\lfloor \frac{\left\lfloor \frac{|v_i'|}{2^{m_i^{\mathcal{Q}}}} \right\rfloor 2^{m_i^{\mathcal{Q}}} + \left\lfloor 2^{m_i^{\mathcal{Q}}} \mathrm{r} \right\rfloor}{2^{\kappa+1}} \right\rfloor 2^{\kappa+1}}{2^\kappa} \right\rfloor \\
&= \left\lfloor \frac{\left\lfloor \frac{|v_i'|}{2^{m_i^{\mathcal{Q}}}} \right\rfloor 2^{m_i^{\mathcal{Q}}} - \left\lfloor \frac{\left\lfloor \frac{|v_i'|}{2^{m_i^{\mathcal{Q}}}} \right\rfloor + \frac{\left\lfloor 2^{m_i^{\mathcal{Q}}} \mathrm{r} \right\rfloor}{2^{m_i^{\mathcal{Q}}}}}{2^{\kappa+1-m_i^{\mathcal{Q}}}} \right\rfloor 2^{\kappa+1}}{2^\kappa} \right\rfloor
\end{aligned}
$$

which, since $\left\lfloor \frac{|v_i'|}{2^{m_i^{\mathcal{Q}}}} \right\rfloor \in \mathbb{Z}$ and $0 \leq \frac{\left\lfloor 2^{m_i^{\mathcal{Q}}} \mathrm{r} \right\rfloor}{2^{m_i^{\mathcal{Q}}}} < 1$ (2.20b) and $\kappa, m_i^{\mathcal{Q}} \in \mathbb{Z}, m_i^{\mathcal{Q}} \leq \kappa$ so $2^{\kappa+1-m_i^{\mathcal{Q}}} \in \mathbb{N}$,

$$= \left\lfloor \frac{\left\lfloor \frac{|v_i'|}{2^{m_i^{\mathcal{Q}}}} \right\rfloor 2^{m_i^{\mathcal{Q}}} - \left\lfloor \frac{\left\lfloor \frac{|v_i'|}{2^{m_i^{\mathcal{Q}}}} \right\rfloor}{2^{\kappa+1-m_i^{\mathcal{Q}}}} \right\rfloor 2^{\kappa+1}}{2^{\kappa}} \right\rfloor \qquad \text{Lemma D.2.2}$$

$$= \left\lfloor \frac{\left\lfloor \frac{|v_i'|}{2^{m_i^{\mathcal{Q}}}} \right\rfloor 2^{m_i^{\mathcal{Q}}} - \left\lfloor \frac{\left\lfloor \frac{|v_i'|}{2^{m_i^{\mathcal{Q}}}} \right\rfloor 2^{m_i^{\mathcal{Q}}}}{2^{\kappa+1}} \right\rfloor 2^{\kappa+1}}{2^{\kappa}} \right\rfloor$$

$$= \left\lfloor \frac{\left\lfloor \frac{|v_i'|}{2^{m_i^{\mathcal{Q}}}} \right\rfloor 2^{m_i^{\mathcal{Q}}} - \left\lfloor \frac{|v_i'|}{2^{\kappa+1}} \right\rfloor 2^{\kappa+1}}{2^{\kappa}} \right\rfloor \qquad \text{Lemma D.2.3}$$

$$= \left\lfloor \frac{\left\lfloor \frac{Q_{2^{j_i}}(v_i)+u_i}{2^{m_i^{\mathcal{Q}}}} \right\rfloor 2^{m_i^{\mathcal{Q}}} - \left\lfloor \frac{Q_{2^{j_i}}(v_i)+u_i}{2^{\kappa+1}} \right\rfloor 2^{\kappa+1}}{2^{\kappa}} \right\rfloor \qquad (5.3c)$$

$$= \left\lfloor \frac{\left\lfloor \frac{\left\lfloor \frac{|v_i|}{2^{j_i}} \right\rfloor 2^{j_i}+u_i}{2^{m_i^{\mathcal{Q}}}} \right\rfloor 2^{m_i^{\mathcal{Q}}} - \left\lfloor \frac{\left\lfloor \frac{|v_i|}{2^{j_i}} \right\rfloor 2^{j_i}+u_i}{2^{\kappa+1}} \right\rfloor 2^{\kappa+1}}{2^{\kappa}} \right\rfloor \qquad (5.3a)$$

$$= \left\lfloor \frac{\left\lfloor \left\lfloor \frac{|v_i|}{2^{j_i}} \right\rfloor 2^{j_i-m_i^{\mathcal{Q}}} + \frac{u_i}{2^{m_i^{\mathcal{Q}}}} \right\rfloor 2^{m_i^{\mathcal{Q}}} - \left\lfloor \left\lfloor \frac{|v_i|}{2^{j_i}} \right\rfloor 2^{j_i-\kappa+1} + \frac{u_i}{2^{\kappa+1}} \right\rfloor 2^{\kappa+1}}{2^{\kappa}} \right\rfloor$$

which, as $j_i^{\mathcal{Q}} = j_i$ (6.4c) and $m_i^{\mathcal{Q}} \leq \kappa < j_i^{\mathcal{Q}}$ gives $m_i^{\mathcal{Q}} \leq \kappa < j_i$ so $\left\lfloor \frac{|v_i|}{2^{j_i}} \right\rfloor 2^{j_i-m_i^{\mathcal{Q}}} \in \mathbb{Z}$ and $\left\lfloor \frac{|v_i|}{2^{j_i}} \right\rfloor 2^{j_i-\kappa-1} \in \mathbb{Z}$,

$$= \left\lfloor \frac{\left\lfloor \frac{|v_i|}{2^{j_i}} \right\rfloor 2^{j_i} + \left\lfloor \frac{u_i}{2^{m_i^{\mathcal{Q}}}} \right\rfloor 2^{m_i^{\mathcal{Q}}} - \left\lfloor \frac{|v_i|}{2^{j_i}} \right\rfloor 2^{j_i} - \left\lfloor \frac{u_i}{2^{\kappa+1}} \right\rfloor 2^{\kappa+1}}{2^{\kappa}} \right\rfloor$$

$$
\begin{aligned}
&= \left\lfloor \frac{\left\lfloor \frac{u_i}{2^{m_i^{\mathcal{Q}}}} \right\rfloor 2^{m_i^{\mathcal{Q}}} - \left\lfloor \frac{u_i}{2^{\kappa+1}} \right\rfloor 2^{\kappa+1}}{2^{\kappa}} \right\rfloor \\
&= \left\lfloor \frac{\left\lfloor \frac{\sum_{x=0}^{j_i-1} 2^x u_{i,x}}{2^{m_i^{\mathcal{Q}}}} \right\rfloor 2^{m_i^{\mathcal{Q}}} - \left\lfloor \frac{\sum_{x=0}^{j_i-1} 2^x u_{i,x}}{2^{\kappa+1}} \right\rfloor 2^{\kappa+1}}{2^{\kappa}} \right\rfloor && (6.1) \\
&= \left\lfloor \frac{\sum_{x=m_i^{\mathcal{Q}}}^{j_i-1} 2^x u_{i,x} - \sum_{x=\kappa+1}^{j_i-1} 2^x u_{i,x}}{2^{\kappa}} \right\rfloor \\
&= \left\lfloor \frac{\sum_{x=m_i^{\mathcal{Q}}}^{\kappa} 2^x u_{i,x}}{2^{\kappa}} \right\rfloor && m_i^{\mathcal{Q}} \le \kappa < j_i \\
&= u_{i,\kappa} + \left\lfloor \frac{\sum_{x=m_i^{\mathcal{Q}}}^{\kappa-1} 2^x u_{i,x}}{2^{\kappa}} \right\rfloor && u_{i,\kappa} \in \{0,1\} \\
&= u_{i,\kappa} && 0 \le \sum_{x=0}^{\kappa-1} 2^x u_{i,x} < 2^{\kappa}.
\end{aligned}
$$

So property (6.19c) is proven.

E.1.6 Proof for Output

We show that given the correct watermarking parameters and a potentially scaled but otherwise untampered watermarked image, of which at least the Φ most significant bit planes of the lowest resolution layer have not been lost, then our watermark detector will generate the output 'True'. That is, whenever both the candidate and corresponding extracted watermark bits exist, those bits will match.

Let

$$
\text{Output} = \begin{cases} \text{True} & \text{if } \forall i, \kappa, 0 \le \kappa < j_i^* \text{ s.t. } \exists u_{i,\kappa}^c, u_{i,\kappa}^d, \quad u_{i,\kappa}^c = u_{i,\kappa}^d \\ \text{False} & \text{if } \exists i, \kappa, 0 \le \kappa < j_i^* \text{ s.t. } \exists u_{i,\kappa}^c, u_{i,\kappa}^d, \quad u_{i,\kappa}^c \neq u_{i,\kappa}^d \end{cases}
$$

If $I^* = F(\mathsf{Embed}(I, \Lambda))$ is an unscaled, resolution scaled, or quality scaled watermarked image, $\Lambda' = \Lambda$, then,
$\forall i, \kappa, 0 \le \kappa < j_i^*$ s.t. $\exists u_{i,\kappa}^c, u_{i,\kappa}^d$,

$$
\begin{aligned}
&\exists u_{i,\kappa}^c \wedge \exists u_{i,\kappa}^d \\
u_{i,\kappa}^c = u_{i,\kappa} &\wedge \exists u_{i,\kappa}^d && (6.17) \\
u_{i,\kappa}^c = u_{i,\kappa} \wedge u_{i,\kappa} &= u_{i,\kappa}^d && (6.19) \\
u_{i,\kappa}^c &= u_{i,\kappa}^d
\end{aligned}
$$

Therefore Output = True.

Note that we have proven these properties separately for unscaled (watermarked), resolution scaled and quality scaled images but that because resolution and quality scaling are separable operations, the same properties will hold for images which have been subject to both resolution and quality scaling but are otherwise untampered.

E.2 Additional Details on the Evaluation of the Improved Algorithm

E.2.1 Correctness and Fragility

E.2.1.1 Key Sensitivity

Although there was no trend towards lower BERs as the detection key approached the embedding key (sect. 6.4.2.2) the error rates were not as high as 50%, and lower BERs were obtained with higher resolution layers. This appears to be because the proportion of zero and one bits in the watermarks are unequal.

Watermarking image 1 using key 1 produces a watermark containing roughly 77% zero bits, which is a similar percentage to those of other images. Higher resolution layers have a higher chance of producing a zero watermark bit, as they typically contain a far larger number of zero-valued coefficients.

Lower quality layers also have a higher chance of producing a zero watermark bit from the intra-resolution feature sequences, as a larger quantization step size is used when generating watermark bits in more significant bit-planes. However, this is not apparent from the quality scaled results (table 6.4), presumably because lower quality layers consist primarily codeblocks with lower concentrations of zero coefficients, cancelling out the effect.

The lowest error rates occur in the second image (figs. E.1 and E.2), which consists primarily of smooth regions. Other images with smooth regions (e.g. 8, 16) show the same problem, which is consistent with the observation that large numbers of zero-valued high-resolution coefficients causes a high percentage of zero watermark bits, resulting in BERs below 50%.

E.2.1.2 Recompression

The bit error rates after decompression and recompression are depicted for individual resolution and quality scaled subimages in figures E.3 and E.4.

Many images have lower BERs in both the highest and lowest resolution layers. In the highest resolution layer of most images, many coefficients will be zero, both in the

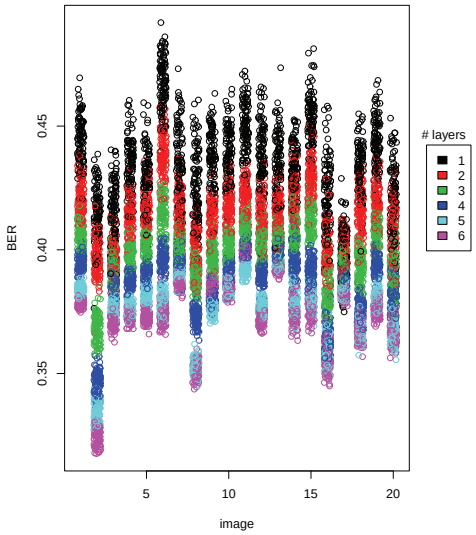


Figure E.1: Bit error rate for resolution scaled subimages: incorrect detection key.

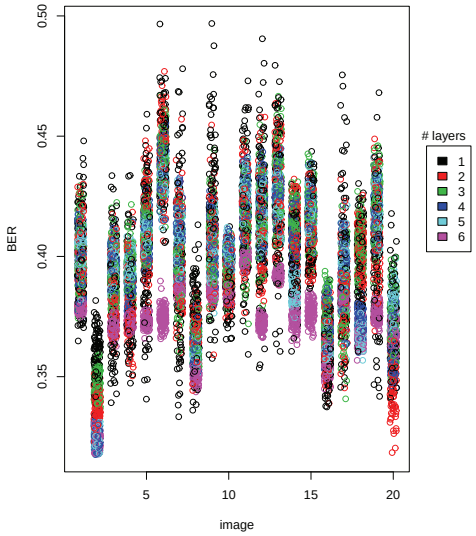


Figure E.2: Bit error rates for quality scaled subimages: incorrect detection key.

watermarked and recompressed images. Thus watermark bits in the lowest resolution layer are more likely to be derived from feature sequences consisting entirely of zero coefficients, and not produce an error. In the lowest resolution layer, coefficients are less likely to be altered by recompression. In particular, the more significant bit planes are likely to remain unchanged, so many intra-resolution sequences, which use a higher quantization step size than the intra-codeblock sequences, are likely to remain unchanged and not produce an error.

The trend of increasing error rate with increasing quality, observed for the unsecured algorithm (sect. D.4.1.3) is still evident in the BER results of individual quality scaled images for this algorithm (fig. E.4). This is because decompression and recompression produces relatively minor changes in the image, which primarily affects the higher quality layers.

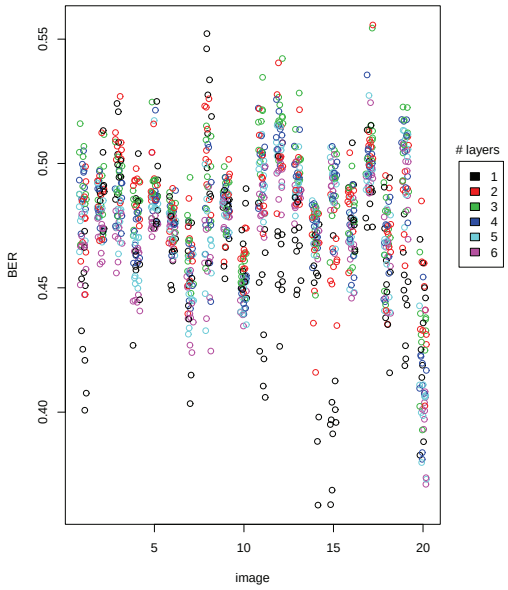


Figure E.3: Bit error rates for resolution scaled subimages: Recompress.

E.2.2 Scalability

E.2.2.1 Detectability

Although all tested images produced detectability values greater than 30 (see section 6.4.3), it is possible to obtain a rough estimate the fraction of images for which the detectability falls below 30 using essentially the same method that of section D.4.1.2. This is done by averaging the detectability values associated with each original image across all keys and using these to estimate the expected detectability of a watermark over the entire population of images.

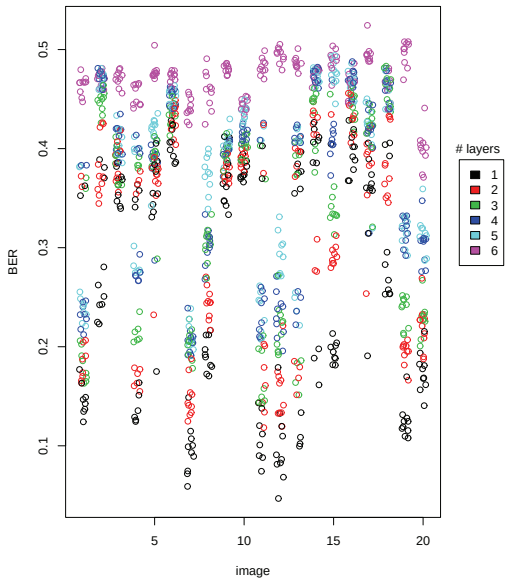


Figure E.4: Bit error rates for quality scaled subimages: Recompress.

Initially, it is assumed that a normal distribution can be used to model both resolution and quality detectability values. Normal quantile plots suggest that this is a reasonable assumption for resolution detectability (fig. E.5) but not for quality detectability (fig. E.6). For quality detectability, the log-normal distribution is used instead, and the log-normal quantile plot (fig. E.7) shows that this is a much better fit.

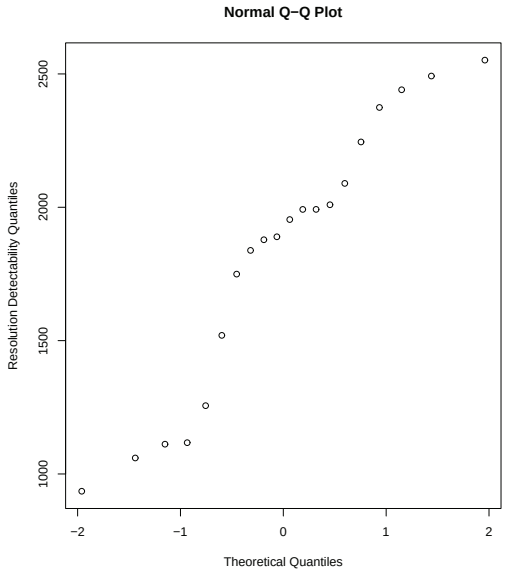


Figure E.5: Normal quantile plot for resolution detectability, averaged across keys.

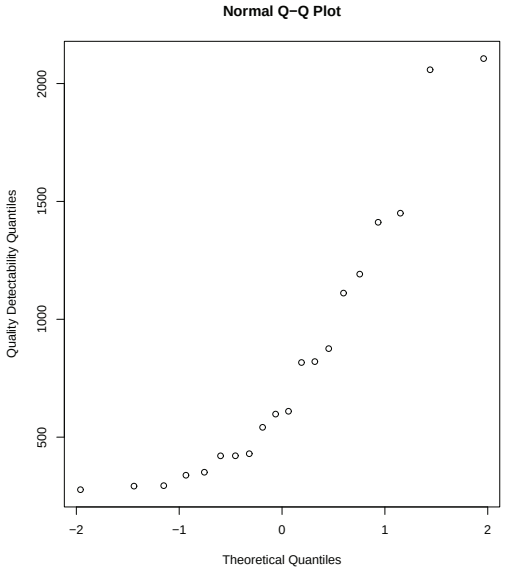


Figure E.6: Normal quantile plot for quality detectability, averaged across keys.

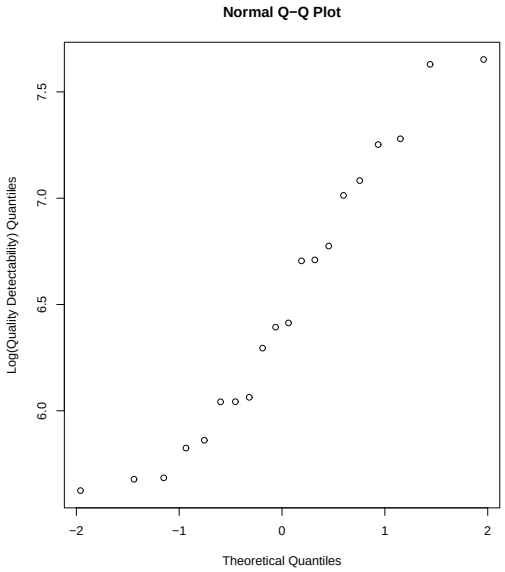


Figure E.7: Log-normal quantile plot for quality detectability, averaged across keys.

The mean and standard deviations for the key-averaged resolution and quality detectability values are used to parameterize normal and log-normal distributions respectively. From these distributions the fraction of images with average detectability below 30 can be easily estimated (tab. E.1) as roughly 0.00018 for resolution scaling and roughly 0.0000012 for quality scaling.

Table E.1: Estimated fraction of images for which the key-averaged detectability will be below thirty $\mathrm{P}(\bar{\mathcal{D}} < 30)$ where the processing type $\mathcal{F}$ is either resolution scaling $\mathcal{R}$ or quality scaling $\mathcal{Q}$.

$\mathcal{F}$	mean($\bar{\mathcal{D}}$)	sd($\bar{\mathcal{D}}$)	Assumed Distribution	P($\bar{\mathcal{D}}$ <30)
$\mathcal{R}$	1824.80	503.8156	Normal	0.0001837299
$\mathcal{Q}$	820.79	566.7723	Log-Normal	0.000001162658

E.2.2.2 Graceful Improvement

The quality graceful improvement results for the secured algorithm were often below 0.9, suggesting that too few watermark bits were extracted from some layers, and too many from others.

By plotting

$$\frac{\epsilon^l - \iota^l}{\iota^l},$$

for each layer l of each image (fig. 6.8) we can identify which layers provided too many or too few watermark bits; a perfect distribution of watermark bits would have zero for all layers, values above zero indicate that too many bits were extracted values below zero indicate that not enough bits were extracted.

From figure 6.8 it is clear that too little of the watermark is recoverable from the lower quality layers, and too much is only recoverable from higher quality layers. This suggests that our algorithm is embedding too little in visually significant areas or that too few feature coefficient bits are being recovered at lower quality layers. The first of these problems could be reduced by better adaptation of the watermark embedding locations to the human visual system. The second problem could be reduced by biasing intra-resolution feature coefficient selection towards more visually significant coefficients, which are more likely to be present at low quality layers, when embedding in more visually significant areas.

E.2.3 Tampering

E.2.3.1 Mark Transfer Attack

The bit error rates for individual subimages that have been subjected to a mark transfer attack are consistently high.

For individual resolution scaled images, the BERs range between 45 and 53 percent. Note that the image index on the x axis of figure E.8 corresponds to the image from which the watermark was transferred. As with the average results (sect. 6.4.4.1), there is a general trend towards slightly lower error rates as the number of resolution layers increases, due to the larger proportion of zero coefficients in the highest resolution layers.

For quality scaled images, the bit error rates are somewhat lower, between 39 and 53 percent for individual images (fig. E.9).

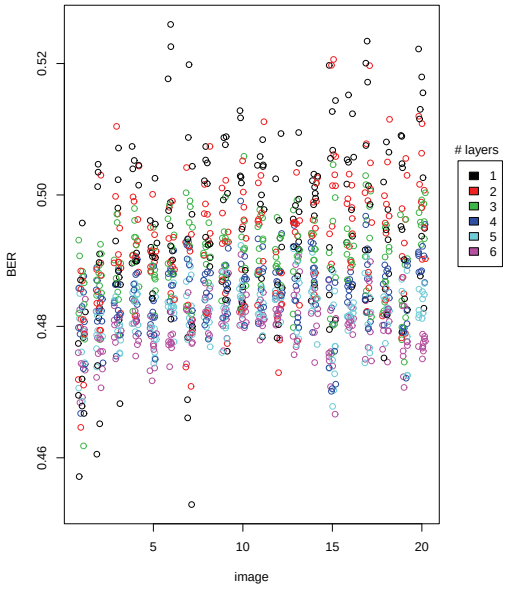


Figure E.8: Bit error rates for resolution scaled subimages: Mark Transfer.

E.2.3.2 Collage Attack

The bit error rates for individual resolution scaled subimages (fig. E.10) vary between 32 and 45 percent. Those for individual quality scaled subimages (fig. E.11) vary between 27 and 43 percent.

The trend towards lower BERs as the number of resolution layers increases, which was present in the average results, is also clear in the individual results.

This is consistent with the observation that a high proportion of zero-valued watermark bits responsible for the less than 50% error rates. Higher resolution layers have a higher chance of producing a zero watermark bit, as they typically contain a far larger number

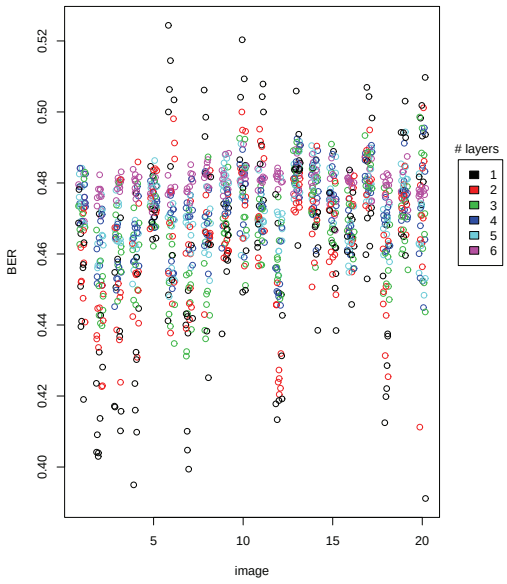


Figure E.9: Bit error rates for quality scaled subimages: Mark Transfer.

of zero-valued coefficients. As was the case with the key sensitivity experiments (sect. E.2.1.1) the smoother images 2, 8 and 16, which have little high-resolution detail, show the poorest detection results.

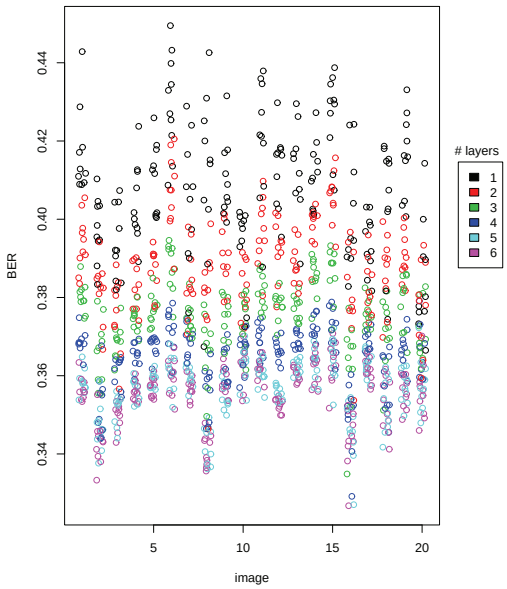


Figure E.10: Bit error rates for resolution scaled subimages: 19-image collage.

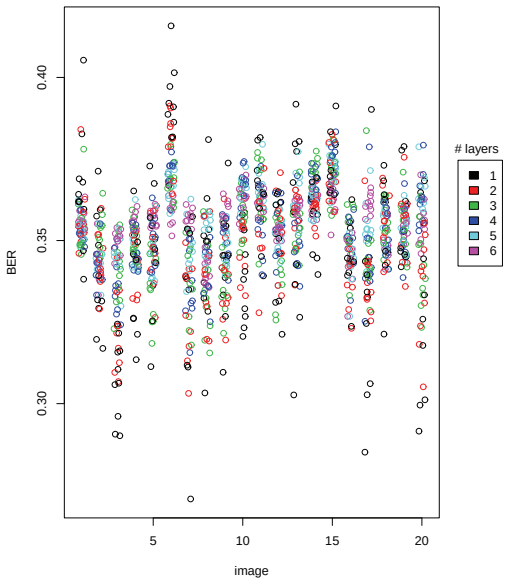


Figure E.11: Bit error rates for quality scaled subimages: 19-image collage.