University of Wollongong

Research Online

Faculty of Engineering and Information
Sciences - Papers: Part A

Faculty of Engineering and Information
Sciences

1-1-2013

Scalable fragile watermarking for image authentication

Angela Piper

University of Wollongong, apiper@uow.edu.au

Reihaneh Safavi-Naini

University of Calgary, rei@uow.edu.au

Follow this and additional works at: <https://ro.uow.edu.au/eispapers>

Part of the [Engineering Commons](#), and the [Science and Technology Studies Commons](#)

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au

Scalable fragile watermarking for image authentication

Abstract

Semi-fragile watermarks are used to detect unauthorised changes to an image, whereas tolerating allowed changes such as compression. Most semi-fragile algorithms that tolerate compression assume that because compression only removes the less visually significant data from an image, tampering with any data that would normally be removed by compression cannot affect a meaningful change to the image. Scalable compression allows a single compressed image to produce a variety of reduced resolution or reduced quality images, termed subimages, to suit the different display or bandwidth requirements of each user. However, highly scaled subimages remove a substantial fraction of the data in the original image, so the assumption used by most semi-fragile algorithms breaks down, as tampering with this data allows meaningful changes to the image content. The authors propose a scalable fragile watermarking algorithm for authentication of scalable JPEG2000 compressed images. It tolerates the loss of large amounts of image data because of resolution or quality scaling, producing no false alarms. Yet, it also protects that data from tampering, detecting even minor manipulations other than scaling, and is secure against mark transfer and collage attacks. Experimental results demonstrate this for scaling down to 1/1024th the area of the original or to 1/100th the file size.

Keywords

era2015, image, watermarking, authentication, fragile, scalable

Disciplines

Engineering | Science and Technology Studies

Publication Details

Piper, A. & Safavi-Naini, R. (2013). Scalable fragile watermarking for image authentication. *IET Information Security*, 7 (4), 300-311.

Scalable Fragile Watermarking for Image Authentication

Angela Piper* and Reihaneh Safavi-Naini†

February 14, 2012

Abstract

Semi-fragile watermarks are used to detect unauthorised changes to an image, while tolerating allowed changes such as compression. Most semi-fragile algorithms that tolerate compression assume that because compression only removes the less visually significant data from an image, tampering with any data that would normally be removed by compression cannot affect a meaningful change to the image.

Scalable compression allows a single compressed image to produce a variety of reduced resolution or reduced quality images, termed subimages, to suit the different display or bandwidth requirements of each user. However, highly scaled subimages remove a substantial fraction of the data in the original image, so the assumption used by most semi-fragile algorithms breaks down, as tampering with this data allows meaningful changes to the image content.

We propose a scalable fragile watermarking algorithm for authentication of scalable JPEG2000 compressed images. It tolerates the loss of large amounts of image data due to resolution or quality scaling, producing no false alarms. Yet it also protects that data from tampering, detecting even minor manipulations other than scaling, and is secure against mark transfer and collage attacks. Experimental results demonstrate this for scaling down to 1/1024 th the area of the original or to 1/100 th the file size.

1 Introduction

In this paper we consider watermarking for image authentication when scaling (resolution and quality) must be tolerated. We assume scaling is performed using JPEG2000: a state-of-the-art scalable compression algorithm.

*Angela Piper is with the School of Computer Science and Software Engineering at the University of Wollongong, Australia. Email: amp08@uow.edu.au

†Reihaneh Safavi-Naini is with the Department of Computer Science at the University of Calgary, Canada. Email: rei@ucalgary.ca Tel: +1 403 2105492 Fax: +1 403 2844707

A common way of providing protection against image tampering is to use watermarks.¹ *Fragile* watermarks are designed to be damaged by any changes to an image and changes to the image can be detected and located. Fragile watermarks do not however tolerate compression and so this severely limits their application, as images in most storage and communication systems are in compressed form.

Semi-fragile watermarks are designed to be damaged by tampering but also to remain unchanged by a set of allowed operations that may include image compression. Compression tolerance is often obtained by excluding all image data which is expected to be lost during compression, from the watermarking process. Although this ensures that the watermark will be correctly recovered from an untampered compressed image, it allows an attacker to tamper freely with and data that has not contributed to the watermark. This means that semi-fragile watermarks that are compression tolerant may not always be usable for image authentication.

Scalable compression systems allow an image to be compressed once but decompressed in a variety of ways, to suit the network and display resources available to the user. Scalability is commonly in the form of *resolution* or *quality* scalability. A fragile watermark that can support these two types of scalability must remain detectable in all scaled versions of the image, but be damaged by any other change to the image.

This paper presents a scalable fragile watermarking system, for use in image authentication in which JPEG2000 [7] scalable compression is used. We use the term ‘scalable fragile’ to emphasise that only valid resolution or quality scaling operations are permitted, and so recompression is not. This allows scalable compression to be used in the distribution of an image that is otherwise intended to remain unmodified (see the motivating scenario in section 1.2). In particular, it ensures that minute tampering in the spatial domain, which necessarily involves decompression and recompression, can be easily detected. In contrast, a semi-fragile algorithm will typically permit recompression and often a variety of other operations, and thus will be much less sensitive to tampering.

1.1 Challenges of Scalable Fragile Watermarking

In scalable compression systems such as JPEG2000, scalability is achieved through the production of a *base (lowest)* layer, containing the data common to all versions, followed by one or more *refinement (higher)* layers, which are decoded as necessary to produce improved versions of the

¹The primary alternative is to use a digital signature, which allows the detection of changes to the image but does not provide tamper location. Digital signatures are typically sensitive to single bit changes, which offers stronger security, but traditionally do not tolerate compression (although cf. [3] and [13]). Furthermore, the separate transmission of the signature data is unattractive in limited bandwidth environments.

image. For images, there are two main types of scalability. In *resolution scalability*, each additional refinement layer provides higher frequency image data, permitting the display of an image with twice the horizontal resolution and twice the vertical resolution of the previous version. This allows a user to download only the image data that is necessary for the display size of their device (fig. 3c). In *quality scalability*, each additional refinement layer provides more finely quantized image data, producing an increase in visual quality. This allows a user to download the version of the image that best fits their available network bandwidth (fig. 3d). Resolution and quality scalability provide the same functionality as resizing and recompression, using minimal storage overhead and only a simple transcoding operation.

The most important requirement for scalable fragile watermarking is to ensure minimal false alarms in a resolution or quality scaled image, i.e. tolerance to scaling. Most semi-fragile algorithms achieve their tolerance to rescaling through tolerance to recompression, and authentication is obtained by only protecting the least acceptable compressed image. However such techniques rely on the assumption that higher layers have negligible effect on image semantics and so tampering with data in those layers cannot affect the ‘meaning’ of an image. This is a valid assumption when only mild compression is tolerated. In many scalable compression scenarios, scalable compression must cater for a wide range of users: those with ‘high-end’ devices and access to high bandwidth, as well as users of ‘low-end’ devices with small display size or bandwidth. To ensure verification succeeds for the latter type of users, the watermark must remain acceptable under severe reductions in resolution or quality. However the lowest resolution or quality layer is likely to represent only a small portion of the full image that will be accessed by the high-end users. This may provide an opportunity to the attacker to make meaningful changes to an image without affecting the base layer, hence violating its authenticity.

Figure 1a shows an authentic image, with 6 resolution and 6 quality layers. Leaving the lowest resolution layer untouched, we are able to change the text on the aeroplane (fig. 1b). Leaving the lowest quality layer untouched, we are able to add a copyright notice (fig. 1c). A *resolution scalable fragile watermark* must allow authentication at low resolutions, but should not permit arbitrary changes to higher resolutions, which frequently contain important edge information. Similarly, a *quality scalable fragile watermark* must not only allow authentication at the lowest quality layer, but should also protect higher quality layers.

1.2 Our Work

Scenario: Consider a company that offers an image collection online via a subscription service. The images are scalably compressed using JPEG2000 so that subscribers may access these images

(a)

(b)

(c)

Figure 1: (a) Zoomed view of the original image, (b) the aeroplane's text is modified to read "CHINA SOUTHERN" without changing the lowest resolution layer, (c) a copyright mark © J. Citizen is inserted without changing the lowest quality layer

at the resolution and quality that best suits their display and bandwidth preferences. The company stores a single copy of each image on the server and each subscriber uses a watermark detector to check the integrity of a received image. The images are not intended to be altered in any way other than by the scaling that is required by users. No operation other than resolution and quality scaling is permitted, and malicious tampering must be detected.

In the following we outline our main contributions that consist of: a framework for the evaluation of scalable watermarking systems used for image authentication in the scenario above, and a scalable watermarking algorithm that can be used for authentication of images in this scenario, tolerating both resolution and quality scaling.

Framework We propose two required properties for scalable fragile watermarks:

Detectability requires that detection algorithm should be able to detect tampering in a highly-scaled image. A fragile watermark that satisfies detectability:

- tolerates scaling as an allowed operation
- protects the base layer from tampering.

Graceful Improvement is intended to ensure that higher resolution and quality layers are also protected by the watermark, so tampering in higher layers, and not just the base layer, must be detected. More specifically, the more important a given layer is to the image, the larger the portion of the watermark that should be devoted to that layer. A fragile watermark with graceful improvement

- protects all higher layers from tampering
- protects each layer according to its contribution to the image

A scalable fragile watermarking algorithm for authentication must provide the above properties against a range of attacks. Attacks include not only simple forms of tampering, but also more sophisticated attacks. In this paper we consider simple tampering in both the spatial and transform domains, as well as *mark transfer*, *Holliman-Memon*, and *collage attacks* (section 2.2.3). These are the most important known attacks.

Algorithm We present a scalable fragile watermarking algorithm specifically for JPEG2000 compressed images. Our algorithm satisfies the important properties of a scalable watermarking algorithm, detectability and graceful improvement, and is secure against sophisticated attacks. To provide *detectability*, features constructed from the lowest resolution and quality layers are embedded in those layers. To allow *graceful improvement* (and prevent the attacks of section 1.1), other

features include (and are embedded in) higher resolution and quality layers. These features are designed to maximize tamper sensitivity while maintaining both scalability and security against attacks. To *eliminate false alarms* we ensure that for any untampered resolution or quality scaled image, there will be an exact match between the candidate watermark bits (that are expected to be present in the image), and the extracted ones. Comparison of the candidate and extracted watermark bits is only performed on bits that can be correctly recovered. Watermark synchronisation is maintained through careful indexing and selection of coefficients. To *prevent mark transfer*, *Holliman-Memon and collage attacks*, watermark construction is based on features from pseudorandomly selected coefficients that are not restricted to within a block. This ensures the watermark is neither image nor blockwise independent, and that the dependencies cannot be easily determined by an attacker. These features are provided even at resolution scaling to $\frac{1}{1024}$ th the area of the original or at quality scaling to compression rate 0.01 ($\frac{1}{100}$ th the file size). We present the design approach and the details of the algorithm in section 3 and show, in the experiments of section 4, that our algorithm achieves all of the design goals.

Importance of our work: Authentication guarantee for a wide range of scaling in the resolution and quality domains is a non-trivial task. The overwhelming majority of previous works focused on scalable watermarking, discussed below, have been robust watermarks. Existing compression-tolerant semi-fragile watermarks protect only the lower layer and so do not satisfy graceful improvement, provide scalability in one domain, or require the domain of scaling to be known before embedding. (More details below.) That is, no existing algorithm can provide resolution and quality scalability simultaneously. Our work gives such an algorithm and evaluates its properties using a well defined set of requirements and an evaluation framework.

1.3 Related Work

Almost all watermarks that specifically consider scalability [21, 19, 1, 10, 16] are robust watermarks that are used for copyright protection. For authentication purposes, fragility is required. The only known algorithm designed for both scalability and fragility is due to Steinder et al. [18]. It allows detection in any scaled image that contains the lowest resolution layer, satisfying the detectability property. However, as its authors note, the algorithm will falsely authenticate a tampered image provided the lowest layer is unmodified.

There are semi-fragile watermarks that allow scaling. Unlike our algorithm, these permit changes such as recompression, blurring, sharpening etc. Sun et al. [20] provide an algorithm which permits resolution or quality scaling. However authentication is only for the *least authenticable bit rate* (LABR), which essentially describes the least acceptable scaled image, so higher

resolution or quality layers may be freely manipulated. Furthermore, the user must specify the type of scaling during embedding or suffer a dramatic drop in tamper sensitivity.

The closest work to our own is the semi-fragile algorithm of Schlauweg et al. [15]. It allows resolution scaling, and like our algorithm, protects all resolution layers from manipulation. It also allows quality scaling (down to a given bit-plane) but, unlike our algorithm, does not protect higher bit planes from manipulation. Also like our algorithm, Schlauweg et al. provide security against mark-transfer and Holliman-Memon style attacks. However, they state that they completely sacrifice tamper localization in order to achieve this.

As noted by Meerwald and Uhl [12], any algorithm that is robust to compression will support some degree of quality scalability. Other semi-fragile watermarking algorithms [4, 8, 11], although not designed to allow scaling, have been designed to tolerate scalable compression. However, like Sun et al., these algorithms also take the approach of authenticating the LABR only. Thus, if the LABR is set sufficiently low as to allow detectability at low quality layers, higher quality layers are not protected.

2 Evaluation Framework

To evaluate a scalable fragile algorithm one must consider scalability, security and tamper location.

The watermarking system that we consider in this paper consists of two algorithms: an *embedding algorithm*, which takes an image and a secret key, and uses them to construct and insert a watermark; and a *detection algorithm*, which takes the secret key and a candidate image, extracts the watermark and compares it to a candidate watermark (what we expect to extract from an untampered image). The candidate image is assumed to be authentic if the extracted watermark correctly matches the candidate watermark. The *bit error rate* (BER) measures the fraction of extracted watermark bits that do not match their counterpart candidate watermark bits

$$\text{BER} = \frac{\text{mismatching extracted bits}}{\text{total extracted bits}}.$$

2.1 Scalability

To assess scalability, we will evaluate both *detectability* and *graceful improvement*, as defined in section 1.2, by applying scaling to a set of scalably compressed watermarked images (see section 4 for experiment details).

Detectability For detectability, an algorithm should tolerate resolution and quality scaling and protect the base layer from tampering.

If the algorithm tolerates scaling, the BER should be zero when only valid scaling operations have been applied. If it protects the base layer from tampering, the BER of a 1-layer scaled image (containing the base layer only) should be non-zero after an attack that affects the base layer.

Graceful Improvement For graceful improvement, an algorithm should protect all higher layers from tampering and protect each layer in proportion to its contribution to the image.

If the algorithm protects higher layers from tampering, the BER of a scaled image with one or more refinement layers should be non-zero after an attack that affects those refinement layers.

Determining whether each layer is protected in proportion to its contribution to the image is more complex. We determine the ideal number of watermark bits appropriate to each (resolution or quality) layer l , and compare them to the numbers of correctly extracted watermark bits in each layer.

The number of correctly extracted watermark bits ϵ^l , in each layer l , is simply counted during watermark detection. The ideal number of bits in layer l is defined as

$$\iota^l = N \frac{P^l - P^{l-1}}{P^f - P^e},$$

where N is the number of embedded bits, and P^f , P^e and P^l represent the PSNR values of the full (all layers), empty (mid-grey), and l -layer images. The fit between the ideal and extracted watermark bits is measured by

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

This is then normalised to the range $[0, 1]$ to produce the final graceful improvement measure $\mathcal{G}$, so that a value of 1 indicates a perfect fit to the ideal, while 0 indicates the worst possible fit

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

where m is the non-empty layer that contributes least to image quality.²

2.2 Security

When we say that an algorithm should have non-zero BER after an attack, we do not simply mean tampering in the spatial domain but also more sophisticated attacks. We describe the attack model by which tamper detection ability is tested.

²See [14] for a more complete discussion of the concept behind this measure in the context of robust watermarking (Ch. 3) or for the details of the normalisation procedure (App. B.1).

2.2.1 Attacker's Goal

An attacker's aim is to alter the image so as to fool both the human receiver and the watermark detector. To constitute a successful attack, the tampered image must be meaningfully different from the watermarked image, be of acceptable visual quality, and produce zero BER during detection.

2.2.2 Attacker's Capabilities

We assume that the attacker has 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) or the secret key.

2.2.3 Attack Strategies

The attacker may try one of several possible strategies (listed in increasing order of sophistication):

Tampering in the spatial domain The watermarked image is decompressed and modified in the spatial domain before recompression. Because our algorithm is designed to be fragile to recompression, detection of spatial domain tampering can be tested using the recompression step alone.

Tampering in the wavelet domain The watermark image is decoded, wavelet coefficients are modified, and the image is re-encoded. This type of tampering is not explicitly evaluated, as both collage and watermark transfer attacks are applied in the wavelet domain.

Watermark transferral The attacker attempts to copy the watermark bits from the watermarked coefficients to the tampered coefficients (see e.g. [9]). The attacker must know the watermarking algorithm.

Holliman-Memon attack The attacker exploits the blockwise independence of the watermarking algorithm to produce a counterfeit image by replacing tampered blocks with the most similar block in the same equivalence class (see [5] for details), from the valid watermarked image.

Collage attack A more sophisticated form of Holliman-Memon attack by Fridrich et al. [2], in which the attacker replaces tampered pixels in the watermarked image with the most similar pixel at the same location from other watermarked images. The attacker must know the watermarking algorithm and have access to multiple images that use the same secret key. We apply the wavelet

domain version of this attack, replacing tampered coefficients (with the most similar coefficient at the same index), rather than tampered pixels.

2.3 Tamper Location

In addition to tamper detection against known attacks, an authentication watermark should provide tamper location. We use tampering in the wavelet domain to alter a single pixel in the watermarked image, and show the estimated tampered region.

3 The Proposed Algorithm

3.1 Objectives

Our goal is to develop a scalable fragile watermarking algorithm, suitable for the authentication of JPEG2000 compressed images.

The watermarking algorithm should be both resolution and quality scalable, satisfying detectability and graceful improvement as described in section 2, without requiring the type of scalability to be chosen in advance. This means that scaling should be tolerated (producing minimal false alarms), that a ‘reasonable’ number of watermark bits should be present in the lowest resolution and quality layers and that *all* layers should be protected, according to their contributions in perceptual quality towards the full image.

It should be otherwise fragile, detecting illicit changes to the watermarked image. In particular, the attacks described in section 2 should not result in a meaningful alteration to the watermarked image that is of acceptable visual quality without causing mismatches between candidate and extracted bits. Furthermore, the approximate locations of alterations should be deducible from the locations of the mismatched watermark bits.

3.2 Design

The watermarking algorithm consists of the algorithms $\text{Embed}(I, sk, \Lambda)$ and $\text{Detect}(I^*, sk, \Lambda)$, where I represents the original image, I^* a potentially altered image, sk a secret key, and Λ a set of additional public parameters (sect. 3.3).

Algorithm Overview

The watermark is embedded in selected wavelet coefficients, during JPEG2000 compression of the image I , directly after the quantization step. Each coefficient in the image is assigned an

index i which is used to generate pseudorandom numbers that are required by the watermarking algorithm. For each selected coefficient v_i , a watermark element u_i is generated and then embedded using quantize-and-replace embedding.

Each watermark element u_i consists of j_i bits. First, the number of watermark bits j_i is determined, according to the index i and value of the coefficient v_i using global and local parameters. Then each watermark bit $u_{i,\kappa}$ is generated as described below, where $0 \leq \kappa < j_i$

Each watermark bit carefully constructed from the image coefficients, to ensure high levels of scalability and security against attack. First a pseudorandom sequence of image coefficients, termed a ‘feature sequence’, is selected. The elements of the feature sequence are then quantized. Finally, the watermark bit is constructed using the signs and relative magnitudes of the quantized feature coefficients. This allows us to maintain good tamper sensitivity at all levels of scaling.

Detection of the watermark, in a possibly modified candidate image I^* , begins with the same selection, indexing and watermark generation procedures as the embedding algorithm. The number of missing bits m_i , lost from each coefficient v_i^* due to scaling, is then determined, using information from the JPEG2000 bitstream and headers. The generated watermark is adjusted to produce the candidate watermark, by marking any watermark bits constructed based on missing coefficient bits as invalid. The embedded watermark is extracted by taking the j_i least significant bits, and any extracted bits which have been identified as missing are also marked as invalid. Finally, the candidate and extracted watermarks are compared. If the image is scaled, but otherwise untampered, then all valid bits of candidate watermark and their corresponding extracted bits will match. If a mismatch of even one bit occurs, then the image is regarded as tampered.

The algorithm steps are given in Figure 2. The individual steps of the algorithms are discussed below.

3.2.1 Watermark Embedding

Select_Coefficients

Significant wavelet coefficients are selected during JPEG2000 compression of the image I , directly after the quantization step. The set of selected coefficients contains all coefficients in I with magnitudes greater than or equal to the threshold 2^n , where $n \in \mathbb{N}$ is a system parameter. The use of a selection threshold that is a power of two ensures that any coefficient v that is not completely lost due to scaling will be selected during detection.

Embed:	Detect:
Input: I, sk, Λ Output: I'	Input: v_i, I^*, sk, Λ Output: True or False
Select_Coefficients Index_Coefficients For each selected coefficient v_i $u_i = \text{Generate_Watermark}(i)$ $v'_i = \text{Embed_Watermark}(u_i, v_i)$	Select_Coefficients Index_Coefficients For each coefficient v_i^* $u_i^* = \text{Generate_Watermark}(v_i^*)$ $m_i = \text{Determine_no_of_Missing_Bits}(I^*, v_i^*)$ $u_i^c = \text{Adjust_Candidate_Watermark}(u_i^*)$ $u_i^d = \text{Extract_Watermark}(v_i^*)$ Compare $u_i^c = u_i^d$
Generate_Watermark:	Generate_Watermark_Bit:
Input: i, I, sk, Λ Output: u_i	Input: $i, \kappa, I, sk, \Lambda$ Output: $u_{i, \kappa}$
$j_i = \text{Determine_no_of_Watermark_Bits}(v_i^*)$ For each watermark bit $u_{i, \kappa} = \text{Generate_Watermark_Bit}(i, \kappa)$	Select_Feature_Sequence Quantize_Feature_Coefficients Compute_Watermark_Bit

Figure 2: Outline of the watermark embedding and detection algorithms $\text{Embed}(I, sk, \Lambda)$ and $\text{Detect}(I^*, sk, \Lambda)$, along with the subroutines $\text{Generate_Watermark}(i)$ and $\text{Generate_Watermark_Bit}(i, \kappa)$.

Index_Coefficients

Each coefficient in the image I is assigned an index i which is used to generate the pseudorandom numbers that are required for watermark generation. Coefficients are indexed using a raster scan of each subband in each component of each resolution layer, from lowest to highest.

For example, given an 80×80 colour image with 4 resolution layers, the lowest resolution layer (10×10) would be indexed using 0 to 99 for the Y component, 100 to 199 for the C_b component, and 200 to 299 for the C_r component; followed by the LH, HL and HH subbands of the Y component of layer 1, with indices 300 to 599; and so on.

Because the indices are based on position within the image grid, changes to the coefficient values due to quality scaling will not affect their indices. Similarly, because lower resolution layers are indexed before higher ones, the loss of resolution layers due to scaling will not affect the indices

of the remaining coefficients (in lower layers).

Embed_Watermark

Once the required watermark bits have been generated (see the following section), each j_i -bit watermark element u_i is embedded using quantization and replacement

$$v'_i = \text{sign}(v_i) \left\lfloor \frac{|v_i|}{2^{j_i}} \right\rfloor 2^{j_i} + u_i \quad (1)$$

where $\text{sign}(v_i) = \pm 1$ and $|v_i|$ is the absolute value of v_i . This is done for all selected coefficients to produce the full (i.e. not scaled) watermarked image I' .

3.2.2 Watermark Generation

Determine_no_of_Watermark_Bits

For each coefficient v_i we determine the number of least significant bits j_i that should be replaced by watermark bits during embedding. A global parameter α is used to control the overall strength of the watermark.

Watermark imperceptibility is improved by using a local embedding strength proportional (Weber's Law) to the coefficient magnitude. However, the magnitude of the original coefficient is not known at the detector, so α_i is set to 2^β , where β is the smallest non-negative integer satisfying $2^\beta > \frac{|v_i|}{2}$. This formulation relies only on knowledge of the most significant bit of the selected coefficient, so even if all less significant bits are lost during scaling, the correct value of α_i can be recovered. Thus α_i serves as an approximation to $|v_i|$ that is obtainable from any selected coefficient in the original image, the watermarked image and any resolution or quality scaled version thereof.

To resist high levels of quality scaling, it would be desirable to embed uniformly strongly in all coefficients, however this would compromise the imperceptibility of the watermark. On the other hand, reducing the watermark strength uniformly across all coefficients would compromise robustness to high levels of scaling. We use a pseudorandom number $g(sk, i) \mapsto [0, 1)$ to reduce the watermark strength non-uniformly. This enables greater resistance to high levels of scaling for a given level of image degradation. Furthermore, since j_i and α_i are natural numbers, the use of a real-valued $g(sk, i)$ allows finer control over the image degradation (via adjustments to the strength parameter α) than would otherwise be possible.

The watermark u_i to be embedded in coefficient v_i will be j_i bits in length, where

$$j_i = \max(0, \lfloor \log_2(\alpha \alpha_i g(sk, i)) \rfloor + 1) \quad (2)$$

3.2.3 Watermark Bit Generation

Select_Feature_Sequence

The κ th watermark bit $u_{i,\kappa}$ of u_i is constructed using a feature sequence

$$V_{i,\kappa} = \{v_{i,\kappa,1}, v_{i,\kappa,2}, \dots, v_{i,\kappa,\eta}\} \quad (3)$$

of length $\eta \in \Lambda$, which consists of pseudorandomly selected image coefficients, and provides some assurance that these coefficients remain untampered.

Each feature sequence is pseudorandomly chosen to be one of two types. An *intra-codeblock* sequence is selected entirely from coefficients in the same codeblock as the watermark bit, while an *intra-resolution* sequence is selected from coefficients anywhere within the same resolution layer as the watermark bit.

Intra-codeblock sequences provide tamper location, as all feature coefficients are in the same spatial neighbourhood. They also allow us to easily identify a quantization stepsize that will ensure excellent tamper sensitivity while maintaining robustness to scaling. However, because they are entirely from within a single spatial region, they offer no security against Holliman-Memon style attacks.

Intra-resolution sequences provide security against Holliman-Memon attacks by creating dependencies which span multiple codeblocks. However this comes at the expense of tamper location, because tampering with the sequence will be indicated at spatially disparate points. Furthermore, because the feature coefficients are not from the same codeblock as the watermark bit, a coarser quantization stepsize is required to improve feature availability after quality scaling, which reduces tamper sensitivity.³ Using a mix of both types of sequences provides security against attack, but retains good tamper sensitivity and some tamper location.

Both the sequence type (intra-codeblock or intra-resolution) and the individual feature coefficients are selected pseudorandomly based on a hash of the key sk and the Φ most significant bit-planes of the lowest resolution layer of the image. This ensures that an attacker, who does not know the key sk , cannot compute the feature sequence that corresponds to a given watermark bit. Furthermore, it ensures that replacement using coefficients from one or more images watermarked with the same key will fail unless all images have the same hash value, thereby defeating the mark transfer and collage attacks. Note that only the Φ most significant bit-planes of the lowest resolution layer are used to compute the hash. This allows the correct hash value to be computed from any resolution scaled image, and any quality scaled image of acceptable quality.

³To ensure that feature coefficients are available after resolution scaling, this type of feature sequence is chosen from anywhere within the same resolution as the watermark bit, rather than from anywhere within any resolution.

Quantize_Feature_Coefficients

The x th feature coefficient, $v_{i,\kappa,x}$ is quantized by $2^{q_{i,\kappa,x}}$ before the watermark bit is computed. This increases feature availability under quality scaling, by ensuring that least significant bits (which are likely to be lost during quality scaling) do not form part of the feature.⁴ This is essential because correct reconstruction of a watermark bit requires that the same features used to compute the watermark bit must be available at the detector, even after some least significant bits have been lost. Ideally, whenever a watermark bit is available at the detector (i.e. has not been lost due to scaling), the features used to compute that bit will also be available so that verification can occur. To help achieve this, the value of the quantization stepsize exponent $q_{i,\kappa,x}$ is determined differently depending upon the sequence type.

For intra-codeblock quantization, a stepsize exponent of

$$q_{i,\kappa,x} = \kappa + 1 \quad (4)$$

is used. This exploits a guarantee of JPEG2000 codestreams that if a bit from a given bit plane in a codeblock is received, then all more significant bit-planes in that codeblock have been received. Thus if the κ th least significant watermark bit $u_{i,\kappa}$ is received, then all bit-planes greater than κ in the same codeblock are recoverable. This allows the maximum possible feature sensitivity at all levels of quality scaling, while still ensuring that all features are available whenever the corresponding watermark bit is available.

For intra-resolution sequences there is no similar guarantee, so a large quantization stepsize is required. A stepsize exponent of

$$q_{i,\kappa,x} = (M_{i,\kappa,x} - \lfloor \frac{j_i - 1 - \kappa}{a} \rfloor - 1) \quad (5)$$

is used, where $M_{i,\kappa,x}$ represents the maximum number of significant bit-planes in the subband containing the feature coefficient $v_{i,\kappa,x}$ that is being quantized. The parameter a controls how quickly feature sensitivity is increased as the plane κ , of the watermark bit $u_{i,\kappa}$, decreases.

This quantization stage should also remove any bits of the feature coefficient which will be altered during watermark embedding. Thus if a feature coefficient $v_{i,\kappa,x}$ belongs to the set of selected coefficients, and the quantization stepsize for its feature type was not sufficiently large to remove all $j_{i,\kappa,x}$ watermark bits, then $v_{i,\kappa,x}$ is instead quantized by $2^{j_{i,\kappa,x}}$.

Compute_Watermark_Bit

For each quantized feature coefficient $Q(v_{i,\kappa,x})$ we construct a sign feature and a magnitude feature.

⁴Selecting the feature sequence from same resolution layer as the watermark bit has already ensured that resolution scaling will not affect feature availability.

The sign feature $\mathfrak{s}_{i, \kappa, x}$ is simply the sign bit of $v_{i, \kappa, x}$.

$$\mathfrak{s}_{i, \kappa, x} = \begin{cases} 0 & \text{if } v_{i, \kappa, x} \geq 0 \\ 1 & \text{if } v_{i, \kappa, x} < 0. \end{cases} \quad (6)$$

The magnitude feature is whether the quantized magnitude of the x th feature coefficient $v_{i, \kappa, x}$ is greater than that of the preceding⁵ feature coefficient $v_{i, \kappa, x-1}$ in the sequence.

$$\mathfrak{m}_{i, \kappa, x} = \begin{cases} 0 & \text{if } |Q(v_{i, \kappa, x-1})| \leq |Q(v_{i, \kappa, x})| \\ 1 & \text{if } |Q(v_{i, \kappa, x-1})| > |Q(v_{i, \kappa, x})| \end{cases} \quad (7)$$

The watermark bit $u_{i, \kappa}$ is formed by XORing all sign and magnitude features from its feature sequence.

$$u_{i, \kappa} = \bigoplus_{x=1}^{\eta} (\mathfrak{s}_{i, \kappa, x} \oplus \mathfrak{m}_{i, \kappa, x}) \quad (8)$$

This essentially establishes a pair of inequalities (one for the sign and one for the magnitude) for each coefficient in the feature sequence. The watermark bit will change in value if an odd number of these inequalities are violated. This avoids the problem that occurs with a simple XOR of all bits in the quantized feature coefficients, which generally allows tampering by flipping pairs of bits within a coefficient.

3.2.4 Watermark Detection

When a possibly modified image I^* is received, the detection algorithm $\text{Detect}(I^*, sk, \Lambda)$ extracts a watermark u^d from I^* and generates a candidate watermark u^c , then compares the candidate and extracted marks. To ensure that any bits which have been affected by scaling are not included in the comparison, the number of bits missing from each coefficient due to scaling are determined, and corresponding watermark bits are marked as invalid.

The procedures to select coefficients, index coefficients and generate candidate watermark elements are the same as those used during embedding, and are described in previous sections. After these procedures have been applied, the number of missing bits is calculated and the candidate and extracted watermark elements are produced accordingly.

Determine_no_of_Missing_Bits

We determine the number of missing bits m_i for each coefficient v_i , in order to correctly adjust the candidate watermark. This requires the numbers M_i , Z_i and P_i , which can easily be obtained from the JPEG2000 headers and bitstream.

⁵For these purposes, the feature sequence is deemed to ‘wrap around’ so that the first coefficient in the feature sequence is preceded by the last coefficient in the sequence.

The QCC or QCD marker segments contain the JPEG2000 quantization exponent E_i and the number of guard bits G_i , which allow us to determine⁶ the maximum number of bit-planes

$$M_i = E_i + G_i - 1 \quad (9)$$

that can be expected from the codestream for the subband containing the received coefficient v_i^* .

If the first Z_i bitplanes of the codeblock containing v_i^* are all zero, these bitplanes are not explicitly encoded in the JPEG2000 bitstream. Each explicitly-encoded bitplane will be present in the bitstream as three coding passes, with the exception of the first bitplane, for which only one coding pass is used. Thus for an unscaled image, we would receive $3(M_i - Z_i - 1) + 1$ coding passes. The number of received passes P_i will often be less than this when quality scaling has occurred.⁷

Provided only JPEG2000 resolution and/or quality scaling has occurred, the number of missing magnitude bits from the coefficient v_i^* can be precisely determined from these values in combination with the value of v_i^* itself. This is done using the formula⁸

$$m_i = M_i - Z_i - 1 - \left\lfloor \frac{P_i}{3} \right\rfloor - \begin{cases} 1 & \text{if } P_i \equiv 2 \pmod{3} \\ & \text{and } \bar{v}_i^* = 2^{(M_i - Z_i - 1 - \lfloor \frac{P_i}{3} \rfloor)} \\ -1 & \text{if } P_i \equiv 0 \pmod{3} \text{ and } v_i^* = 0 \\ 0 & \text{otherwise.} \end{cases} \quad (10)$$

Adjust_Candidate_Watermark

If enough bits of all feature coefficients are present⁹ that it is possible to reconstruct the quantized feature coefficients (and thence the correct sign and magnitude features), then the candidate watermark bit is simply the watermark bit u_i^* generated from the image I^* as for embedding.

If, however, the number of bits missing from any coefficient $v_{i,\kappa,x}^*$ in the feature sequence is sufficiently large as to interfere with the value of the quantized feature coefficient $Q(v_{i,\kappa,x}^*)$, then generation of corresponding candidate watermark bit is compromised. In these circumstances, the candidate watermark bit is deemed invalid (denoted $u_{i,\kappa}^c = \perp$).

⁶See [6] annex E, pg. 105–106.

⁷*Ibid.*, annex B.9, pg. 61 The packet headers list Z_i explicitly, as the ‘number of “insignificant” most significant bit-planes’. P_i is easily constructed from the ‘number of coding passes for each code-block in this packet’ across all received packets.

⁸*Ibid.*, annex D, pg. 93–104, by careful examination.

⁹This is guaranteed to occur with intra-codeblock feature sequences, due to the design of the feature quantization stepsize, but will not necessarily occur for intra-resolution sequences.

That is, if we let $V_{i,\kappa}^*$ be the feature sequence associated with the candidate watermark bit $u_{i,\kappa}^c$, then

$$u_{i,\kappa}^c = \begin{cases} u_i^* & \text{if } \forall v_{i,\kappa,x}^* \in V_{i,\kappa}^* \quad m_{i,\kappa,x} \leq q_{i,\kappa,x}^* \\ \perp & \text{if } \nexists v_{i,\kappa,x}^* \vee m_{i,\kappa,x} > q_{i,\kappa,x}^*. \end{cases} \quad (11)$$

Note that in a scaled but otherwise untampered image, either the coefficient $v_{i,\kappa,x}^*$ will have been completely lost (denoted $\nexists v_{i,\kappa,x}^*$) or the quantization stepsize exponent $q_{i,\kappa,x}^*$ calculated during watermark extraction will equal the one $q_{i,\kappa,x}$ used during embedding. Therefore the watermark bit will match the candidate watermark bit, provided neither is invalid.

By adjusting the candidate watermark at the detector according to the level of scaling that has actually occurred, rather than attempting to adjust the watermark at the embedding stage to allow for all levels of scaling, this algorithm is able to tolerate high levels of scaling yet remain sensitive to illicit changes in less highly-scaled images.

Extract_Watermark

The watermark extraction process is essentially the reverse of watermark embedding, with the exception that any watermark bit that is missing due to scaling is identified and deemed invalid (denoted $u_{i,\kappa}^d = \perp$).

Thus if $v_{i,\kappa}^*$ represents the κ th watermark bit that would be extracted from v_i^* if we did not account for scaling, the extracted watermark bit is

$$v_{i,\kappa}^* = \begin{cases} v_{i,\kappa}^* & \text{if } m_i \leq \kappa < j_i \\ \perp & \text{if } \kappa < m_i \text{ or } \kappa \geq j_i \end{cases} \quad (12)$$

Compare

If both the extracted watermark bit $u_{i,\kappa}^d$ and the candidate watermark bit $u_{i,\kappa}^c$ are deemed valid, they are compared. If every pair of corresponding candidate and extracted watermark bits $u_{i,\kappa}^c$ and $u_{i,\kappa}^d$ that is compared produces a match, we conclude that the image has not been modified. If even a single pair does not match, then we may conclude that the image has been tampered with

$$\text{Output} = \begin{cases} \text{True} & \text{if } \forall i : u_i^c \neq \perp \wedge u_i^d \neq \perp, \quad u_i^c = u_i^d \\ \text{False} & \text{if } \exists i : u_i^c \neq \perp \wedge u_i^d \neq \perp, \quad u_i^c \neq u_i^d. \end{cases} \quad (13)$$

This strictness is possible because in a JPEG2000 scaled, but otherwise untampered image, an exact match between u^d and u^c is guaranteed by the following two points. If a coefficient is

completely lost¹⁰ due to scaling, then it will fail the coefficient selection rule (section 3.2.1) and thus no attempt will be made to extract u_i^d nor calculate u_i^c . If a coefficient is partially lost, the number of bits m_i that are missing will be determined, and the corresponding bits from in both the candidate and extracted watermarks will be marked as invalid, and thus not included in the comparison.

If a mismatch between $u_{i,\kappa}^c$ and $u_{i,\kappa}^d$ occurs, it must be a result of changes to either the coefficient v_i^* and/or one or more of the associated feature coefficients $v_{i,\kappa,x}^* \in V_{i,\kappa}^*$, but it is not possible to determine which coefficients were changed. As a result, all these coefficients are marked as suspect. Because these coefficients may be dispersed spatially across the entire image, tamper location relies on finding a concentration of suspect coefficients in a given spatial region. This is discussed in more detail in section 4.3.

3.3 Parameters

The detection algorithm requires the secret key sk used in embedding, and additional parameters $\Lambda = \{n, \Phi, \eta, a, \alpha\}$. The secret key sk is used in both feature and watermark sequence generation, and its length can be chosen to suit the required level of security. The parameters Λ will be around 48 bits and can be sent using a public channel such as the image header.

The exponent n (4 bits) controls the coefficient selection threshold. Higher values reduce complexity and watermark perceptibility at the cost of tamper sensitivity. The number Φ (4 bits), of most significant bit-planes from the lowest resolution layer to be hashed, controls how similar two images must be before identical secret keys will result in identical feature sequences. Higher values increase security at the cost of reducing quality scalability. The parameter η (5 bits) controls the number of coefficients used in each feature sequence. Higher values increase security against Holliman-Memon attacks, but reduce tamper location ability and may cause reduced sensitivity under quality scaling (because all η coefficients must be present before the watermark bit can be verified). The robustness parameter a (3 bits) controls the tradeoff between quality scalability and feature sensitivity. High values maintain coarse features, low values produce increasingly fine features as the watermark bit approaches the LSB. The parameter α (32 bits) controls the watermark embedding strength. Low values favour imperceptibility, high values favour robustness and tamper sensitivity.

¹⁰We say a coefficient is *completely lost* if no magnitude bits remain, and *partially lost* if some magnitude bits remain.

4 Experimental Evaluation

For this evaluation we use 20 original images and 10 embedding keys, to produce 200 watermarked, scalably compressed images. Watermarking occurs during JPEG2000 compression, with 6 resolution layers and 6 quality layers with compression rates 0.01, 0.02, 0.03, 0.04, 0.05 and 0.9999. At these settings, the base resolution layer has $\frac{1}{1024}$ th the original image area and the base quality layer has $\frac{1}{100}$ th the original file size. The watermarking parameters are $n = 3$, $\eta = 4$, $a = 1$, and $\Phi = 1$ for all images. The value of α is varied according to the image to allow a consistent perceptual quality ¹¹.

Figure 3 shows our original image, and the corresponding image produced using the above settings. The psnr values between the original and the compressed watermarked image are approximately 40, 40 and 38 dB for the red, green and blue channels respectively.

Figure 3: (a) The original image, (b) the JPEG2000 compressed watermarked image, (c) resolution scaled, (d) quality scaled

To demonstrate scalability and security according to the framework in section 2, we report the average BER for untampered and attacked images at all levels of scaling, in tables 1 and 2, and the average graceful improvement values $\mathcal{G}$, in table 3.

¹¹Specifically, α is adjusted so that the 99th percentile of the S-CIELAB [22] CIEDE 2000 [17] error is fixed at $4\Delta E$, for a Dell 1702FP (Analog) monitor, 96dpi, viewed at 46cm.

Average BER after Resolution Scaling				
	No Attack	Spatial	Transfer	Collage
1 layer	0.00	0.46	0.49	0.41
2 layers	0.00	0.48	0.49	0.39
3 layers	0.00	0.49	0.49	0.38
4 layers	0.00	0.48	0.49	0.37
5 layers	0.00	0.48	0.48	0.36
6 layers	0.00	0.47	0.48	0.36

Table 1: Average bit error rate for 200 watermarked images after different attacks and levels of resolution scaling

Average BER after Quality Scaling				
	No Attack	Spatial	Transfer	Collage
1 layer	0.00	0.29	0.46	0.35
2 layers	0.00	0.34	0.47	0.35
3 layers	0.00	0.38	0.47	0.35
4 layers	0.00	0.40	0.47	0.36
5 layers	0.00	0.41	0.48	0.36
6 layers	0.00	0.47	0.48	0.36

Table 2: Average bit error rate for 200 watermarked images after different attacks and levels of quality scaling

Average Graceful Improvement ($\mathcal{G}$)	
Resolution Scaling	0.94
Quality Scaling	0.86

Table 3: Average graceful improvement values for 200 watermarked images under resolution and quality scaling.

4.1 Scalability

4.1.1 Detectability

The algorithm tolerates scaling, with a BER of zero on an untampered image at all levels of resolution and quality scaling (No Attack, tables 1 and 2). It also protects both the base resolution

and base quality layers from tampering (1 layer, tables 1 and 2), detecting spatial, mark transfer and collage attacks. Thus the algorithm satisfies both resolution and quality detectability.

4.1.2 Graceful Improvement

The algorithm not only protects the base layer but also higher layers (2–6 layers, tables 1 and 2), with non-zero BER after spatial, mark transfer and collage attacks. The watermark is fairly well distributed amongst the different image layers, relative to their contributions to perceptual quality, with graceful improvement values (table 3) reasonably close to 1, although quality graceful improvement suffers from the difficulty of imperceptibly embedding in the base quality layer.

This allows the detection of attacks in which only the higher layers are modified, such as those in figure 1, as shown in figures 4a—4d.

4.2 Security

The algorithm is secure against all known attacks. Spatial domain, mark transfer and collage attacks were all tested explicitly and, although the BERs reported in tables 1 and 2 are averages, tampering was detected in all 200 images, for each attack, at all tested levels of resolution and quality scaling.

While not explicitly tested, security against wavelet domain tampering and Holliman-Memon attacks is also provided. The feature sensitivity and image dependence that ensure security against mark transfer attacks also ensures security against less sophisticated tampering in the wavelet domain.¹² Similarly, the block interdependence that ensures security against collage attacks also ensures security against the less sophisticated Holliman-Memon attack.

4.3 Tamper Location

Some sample attacks and their corresponding detection results are shown in figure 4. Figures 4a and 4c show wavelet attacks on the higher resolution layers and higher quality layers, respectively, as described in section 1.1 and shown in figure 1. Figure 4e shows the same change made to the aeroplane as figure 4a, but this time using a 50 image collage attack. In each case, the attack is detected but only its approximate location can be identified.

This occurs because in order to achieve the required security, each watermark bit is computed from several coefficients across different spatial regions of the image rather than from individual coefficients or localised blocks. This sacrifices tamper location, as a mismatched watermark bit

¹²Because we use a wavelet domain watermark, mark transfer attacks are applied in the wavelet domain and thus may be considered as a sophisticated form of tampering in the wavelet domain.

Figure 4: Attacked image and tamper detection result pairs: (a)&(b) wavelet attack of fig. 1b on higher resolution layers, (c)&(d) wavelet attack of fig. 1c on higher quality layers, (e)&(f) collage attack on plane text, (g)&(h) mark transfer to new image

causes all coefficients in the corresponding feature sequence to be marked as ‘tampered’. Tamper location therefore relies on a concentration of ‘tampered’ coefficients in a local area, against a less

concentrated background of ‘tampered’ coefficients spread across the image.

Figures 5a and 5b show a single-pixel modification in the wavelet domain, with the detection result in figure 5c. If mark transfer or collage attacks are applied in a small region, the results are similar. Note that although the attack is concentrated in a very small spatial area, the precision of tamper location is poor. This is because the small spatial change causes changes to surrounding low resolution coefficients, which obscure the concentration of high-resolution coefficients in the tamper detection image. This problem can be eliminated by removing the low resolution tamper data from the tamper detection image 5d. However, small modifications using spatial domain tampering (fig. 5e) cannot be located, though they are detected (fig. 5f), as the recompression step causes widespread changes to the image.

5 Conclusion

We have provided a resolution and quality scalable fragile watermarking algorithm for image authentication.

Unlike most semi-fragile authentication watermarks in the literature, our algorithm generates some image features using layers above the lowest authenticable bit rate. This prevents an attacker from freely modifying higher layers, which is particularly important for highly scalable images where the lowest authenticable bit rate contains only a small fraction of the data contained in the full image.

Our algorithm is robust to resolution and quality scaling. By design, it generates zero false alarms due to JPEG2000 resolution or quality scaling for otherwise untampered images and experimental results confirm this at all tested levels of scaling.

It is fragile to other alterations, including any spatial manipulations, which involve recompression of the watermarked image and collage and mark transfer attacks in the wavelet domain.

Our algorithm also provides some degree of tamper location. However, because multiple pseudorandomly selected feature coefficients are associated with each watermark bit, it is difficult to locate changes precisely using this algorithm. We rely on the assumption that most malicious tampering will alter multiple pixels in a small image region, resulting in a higher concentration of watermark errors in that region relative to the more random errors in surrounding areas of the image.

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 makes it harder for an attacker to use multiple images with the same secret key to determine

Figure 5: (a) A single pixel of the watermarked image is modified in the wavelet domain, (b) zoomed view of the modification, (c) tamper detection result, (d) tamper detection result not including low resolution coefficients (e) single pixel modification in spatial domain and recompression, (f) tamper detection result

coefficient sequences. However this means that these 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 it is unlikely that such extreme scaling would result in an image of acceptable 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 an appropriate value for Φ .

The main concept behind our algorithm is that it is not necessary to treat loss due to scaling as if it were noise, because the locations of the lost bits can be precisely identified at the detector. Thus, rather than restricting the embedding region to a specific level of scaling, it is possible to embed a watermark to protect the full image and then generate an appropriately reduced candidate watermark during detection if scaling has occurred. This concept is not specific to authentication watermarking and, with appropriate error correction measures, might also be adapted to applications such as message embedding.

References

- [1] T. P.-C. Chen and T. Chen. Progressive image watermarking. In *Proc. 2000 IEEE Intl. Conf. Multimedia and Expo (ICME 2000). July 30 – Aug 2*, pages 1025–1028.
- [2] J. Fridrich, M. Goljan, and N. Memon. Cryptanalysis of the Yeung-Mintzer fragile watermarking technique. *Journal of Electronic Imaging*, 11(2):262–274, April 2002.
- [3] Shui-Hua Han and Chao-Hsien Chu. Content-based image authentication: current status, issues, and challenges. *International Journal of Information Security*, 9:19–32, 2010.
- [4] 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*, volume 1, pages 7–11, 2004.
- [5] M. Holliman and N. Memon. Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes. *IEEE Trans. Image Processing*, 9(3):432–441, March 2000.
- [6] 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.
- [7] ISO/IEC JTC 1/SC 29/WG 1. ISO/IEC 15444-1 | ITU-T Rec. T.800 : Information technology – JPEG2000 image coding system: Core coding system. 2004.
- [8] C.-Y. Lin and S.-F. Chang. Semifragile watermarking for authenticating JPEG visual content. In P. W. Wong and E. J. Delp, editors, *Proc. SPIE Security and Watermarking of Multimedia Contents II*, volume 3971, pages 140–151, January 2000.
- [9] E. T. Lin and E. J. Delp. A review of fragile image watermarks. In J. Dittmann, K. Nahrstedt, and P. Wohmacher, editors, *Proc. Multimedia and Security Workshop at ACM Multimedia’99. Orlando, FL, USA, Oct. 30-31*, pages 25–29, 1999.

- [10] 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, Beijing, China, Aug. 31 – Sept. 4*, volume 3, pages 2338–2341.
- [11] 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*, 8(1):32–45, February 2006.
- [12] P. Meerwald and A. Uhl. Toward robust watermarking of scalable video. In E. J. Delp, P. W. Wong, J. Dittmann, and N. D. Memon, editors, *Proc. Security, Forensics, Steganography, and Watermarking of Multimedia Contents X, San Jose, C.A., USA, Jan. 26–31*, volume 6819 of *Proceedings of SPIE*, January 2008.
- [13] Cheng Peng, Robert H. Deng, Yongdong Wu, and Weizhong 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, 2003. ACM Press.
- [14] A. Piper. *Scalable Watermarking for Images*. PhD thesis, School of Computer Science and Software Engineering, University of Wollongong, 2010. <http://ro.uow.edu.au/theses/3171>.
- [15] Mathias Schlaueg, Dima Pröfrock, and Erika Müller. Jpeg2000-based secure image authentication. In *Proc. 8th Workshop on Multimedia & Security (MM&Sec '06), Geneva, Switzerland, Sept. 26–27*, pages 62–67, New York, NY, USA, 2006. ACM Press.
- [16] J.-H. Seo and H.-B. Park. Data protection of multimedia contents using scalable digital watermarking. In Y. K. Yang and K. Akingbehin, editors, *Proc. 4th Annu. ACIS Intl. Conf. Computer and Information Science, (ICIS '05), Jeju Island, South Korea, Jul. 14–16*, pages 376–380. IEEE Computer Society, 2005.
- [17] 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*, 30(1):21–30, February 2005.
- [18] 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*, volume 1, pages 641–645, 1999.
- [19] P.-C. Su, H.-J. M. Wang, and C.-C. J. Kuo. An integrated approach to image watermarking and jpeg2000 compression. *J. of VLSI Signal Processing*, 27:35–53, February 2001.

- [20] 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*, volume 2, pages 921–924. IEEE.
- [21] H.-J. M. Wang and C.-C. J. Kuo. An integrated progressive image coding and watermarking system. In *Proc. 1998 IEEE Conf. Acoustics, Speech and Signal Processing. Seattle, WA, USA, May 12–15*, volume 6, pages 3721– 3724, 1998.
- [22] 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*, volume 27, pages 731–734, 1996.