Multiresolution scalable image and video segmentation

Fardin Akhlaghian Tab

University of Wollongong

Follow this and additional works at: https://ro.uow.edu.au/theses

Recommended Citation

http://ro.uw.edu.au/theses/332

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
Multiresolution Scalable Image and Video Segmentation

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

Doctor of Philosophy

from

THE UNIVERSITY OF WOLLONGONG

by

Fardin Akhlaghian Tab
M.Sc. Electrical Engineering

SCHOOL OF ELECTRICAL, COMPUTER AND TELECOMMUNICATIONS ENGINEERING
2005
Abstract

Due to the popularity of multimedia applications, many efforts have been directed towards presenting new services and functionalities such as interactivity, manipulation, content-based retrieval, scalability, etc. Object-based image/video representation and processing is one of the approaches considered to meet these desired functionalities. However, semantic image and video segmentation is one of the unresolved challenges of this approach. Although many works on segmentation have contributed towards this goal, there are still numerous areas requiring further research.

In this work, a comprehensive range of image and video segmentation algorithms, including low and high level phases, are proposed, tested and analysed. In the low level phase, the image/frame is partitioned into homogeneous regions while in the high level phase, the “objects-of-interest” are extracted.

The proposed algorithms are useful for generic segmentation applications, in particular for scalable coding, which distributes information over heterogeneous networks. One of the requirements of the scalable coding is that the shapes of an object produced at different resolutions should be similar, more precisely, the low resolution objects should be the down sampled version of the higher resolution objects. A multi-dimensional processing integrated with the multiresolution segmentation processing reduces computational complexity and provides a scalability feature for the extracted objects/regions at different resolutions, which is necessary for the scalable coding algorithms. Including smoothness as a visual quality criterion in the segmentation and classification algorithms improves the visual effect of the segmentation results. To meet the scalability and smoothness constraints, a Markov Random Field (MRF) framework with enough flexibility to meet the constraints is utilised. The proposed
algorithm is a reliable and effective low level segmentation which includes the desirable features of both single and multiresolution segmentation algorithms.

At the high level phase of the image segmentation process, a hierarchical searching method for extracting the “object-of-interest” is introduced. The search is based on the concept of the global precedence effect (GPE) of the human visual system (HVS) which searches for the large (global) objects before the small (local) ones. The proposed algorithm compares different combination of regions with the “object-of-interest” template to find the best combination. An irregular pyramid is developed which retains the global objects at the lower levels. A hierarchical search for the “object-of-interest” template starts from the lowest level of this pyramid. This natural priority in searching is very useful when the “object-of-interest” is the main object in the image. The computational complexity of the search is reduced significantly.

In video segmentation, the “object-of-interest” in the first frame is determined either by user’s intervention or the proposed “object-of-interest” extraction algorithm. In the subsequent frames, regions generated by the spatial segmentation are grouped into foreground and background areas by a MRF-based classification algorithm. The objective function of the classification algorithm includes spatial and temporal continuity, motion constraints and smoothness terms. The proposed algorithm tracks the objects detected at the previous frames and extracts the newly appearing objects in the current frame. The algorithm is developed in scalable multiresolution mode where the corresponding regions at the lower and higher resolutions are processed and analysed together. The proposed algorithm extracts moving objects at different resolutions with scalability and visual quality (smoothness) as constraints. It allows larger motion detection, better noise tolerance and less computational complexity.
Statement of Originality

This is to certify that the work described in this thesis is entirely my own, except where due reference is made in the text.

No work in this thesis has been submitted for a degree to any other university or institution.

Signed

Fardin Akhlaghian Tab
November, 2005
Acknowledgments

The four years of hard work that were needed to process this thesis from idea to finished work were not painless, and I really relied on the support, understanding and expertise of those around me.

First of all, I would like to express my deep gratitude to my supervisor Associate Professor Golshah Naghdy for invaluable guidance, encouragement and support through my study. Her direction taught me a lot about research and academic life. I am deeply indebted to her. I’d also like to extend my deep appreciations and thanks to my previous supervisor Professor Alfred Mertins for initiating the topic and for his invaluable supervision during the first two years. Also, after his move to the University of Oldenburg, Germany his support influenced my research very much at various stages. I will never forget his moral support at the points where I needed it. I also sincerely thank my co-supervisor Dr. Prashan Premaratne for his advice, encouragement and time devoted to this work.

And finally I dedicate this thesis to my wife Fariba and two daughters Bahar and Shadi for their support and patience during my study. My wife left her job in my country and sacrificed her life to support me in my study. I will never forget it. I relied on her tremendously, because I could not spend enough time with my children, even though they were far from home and relatives.
# Contents

1 Introduction ................................................................. 1
  1.1 Motivation .............................................................. 1
  1.2 Statement of the Problem ........................................... 3
     1.2.1 Effective, Reliable and Scalable Segmentation .......... 4
     1.2.2 Enhancing the Visual Quality of the Extracted Shapes ... 5
     1.2.3 Effective Hierarchical Semantic Object Segmentation ... 5
  1.3 Research Goals ....................................................... 7
  1.4 Thesis Outline ....................................................... 9
  1.5 Major Contributions ................................................ 11
  1.6 Publications .......................................................... 13

2 Literature Review .......................................................... 15
  2.1 Introduction ............................................................ 15
  2.2 Low Level Image Segmentation ..................................... 16
     2.2.1 Edge-Based Segmentation ................................... 17
     2.2.2 Region-Based Segmentation .................................. 18
  2.3 Major Region-Based Segmentation Approaches ................... 18
     2.3.1 Mathematical Morphology .................................... 19
     2.3.2 Bayesian Inference Theory and Markov Random Fields ... 23
<table>
<thead>
<tr>
<th>CONTENTS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>2.4</strong> Multiresolution Image Segmentation</td>
<td>32</td>
</tr>
<tr>
<td>2.4.1 Hierarchical Multiresolution Image Segmentation (HMIS)</td>
<td>34</td>
</tr>
<tr>
<td>2.4.2 Highly Correlated Multiresolution Image Segmentation (HCMIS)</td>
<td>41</td>
</tr>
<tr>
<td><strong>2.5</strong> Semantic Image Segmentation</td>
<td>45</td>
</tr>
<tr>
<td>2.5.1 “Object-of-Interest” Extraction</td>
<td>46</td>
</tr>
<tr>
<td>2.5.2 Scene Segmentation and Interpretation</td>
<td>52</td>
</tr>
<tr>
<td><strong>2.6</strong> Video Segmentation</td>
<td>57</td>
</tr>
<tr>
<td><strong>2.7</strong> Motion</td>
<td>57</td>
</tr>
<tr>
<td>2.7.1 Motion Estimation</td>
<td>58</td>
</tr>
<tr>
<td>2.7.2 Apparent Motion</td>
<td>59</td>
</tr>
<tr>
<td><strong>2.8</strong> Different Approaches in Video Segmentation</td>
<td>61</td>
</tr>
<tr>
<td><strong>2.9</strong> Region-Based Video Segmentation</td>
<td>62</td>
</tr>
<tr>
<td>2.9.1 3D Motion Segmentation</td>
<td>63</td>
</tr>
<tr>
<td>2.9.2 Motion-Based Segmentation</td>
<td>63</td>
</tr>
<tr>
<td>2.9.3 Spatio-Temporal Video Segmentation</td>
<td>64</td>
</tr>
<tr>
<td><strong>2.10</strong> Semantic Video Segmentation</td>
<td>64</td>
</tr>
<tr>
<td>2.10.1 Video Object Tracking</td>
<td>66</td>
</tr>
<tr>
<td>2.10.2 Change Detection</td>
<td>76</td>
</tr>
<tr>
<td>2.10.3 Hybrid Video Segmentation</td>
<td>78</td>
</tr>
<tr>
<td><strong>2.11</strong> Conclusion and Research Direction</td>
<td>80</td>
</tr>
<tr>
<td><strong>3</strong> Towards Scalable Multiresolution Image Segmentation</td>
<td>84</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>84</td>
</tr>
<tr>
<td>3.2 Object-Based Wavelet Coding Scalability</td>
<td>85</td>
</tr>
<tr>
<td>3.3 Morphology-Based Segmentation</td>
<td>88</td>
</tr>
<tr>
<td>CONTENTS</td>
<td></td>
</tr>
<tr>
<td>-------------------------------------------------------------------------------------------------------------------</td>
<td>-----</td>
</tr>
<tr>
<td>3.3.1 Morphological Single Level Segmentation</td>
<td>91</td>
</tr>
<tr>
<td>3.3.2 Hierarchical Morphology-Based Segmentation</td>
<td>91</td>
</tr>
<tr>
<td>3.3.3 Projection to the Next Level</td>
<td>91</td>
</tr>
<tr>
<td>3.3.4 Projection Complexity Reduction</td>
<td>92</td>
</tr>
<tr>
<td>3.4 Simulation Results</td>
<td>94</td>
</tr>
<tr>
<td>3.5 MRF Based Scalable Multiresolution Image Segmentation</td>
<td>98</td>
</tr>
<tr>
<td>3.5.1 Statistical Image Model</td>
<td>99</td>
</tr>
<tr>
<td>3.5.2 MAP Estimation</td>
<td>102</td>
</tr>
<tr>
<td>3.6 Experimental Results and Discussion</td>
<td>104</td>
</tr>
<tr>
<td>3.7 Conclusions</td>
<td>111</td>
</tr>
</tbody>
</table>

| 4 Further Development of the Scalable Segmentation: Smoothness and Colour                                             | 114 |
| 4.1 Introduction                                                                                                   | 114 |
| 4.2 Smoothness                                                                                                     | 115 |
| 4.3 Colour Image Segmentation Algorithm                                                                          | 118 |
| 4.3.1 Statistical Colour Image Model                                                                              | 118 |
| 4.3.2 MAP Estimation                                                                                               | 122 |
| 4.4 Experimental Results and Discussion                                                                           | 123 |
| 4.5 Conclusions                                                                                                    | 130 |

<p>| 5 Meaningful Image Object Extraction                                                                               | 145 |
| 5.1 Introduction                                                                                                   | 145 |
| 5.2 Template Based Object of Interest Extraction at Single Resolution                                              | 146 |
| 5.2.1 Aspect Ratio Test                                                                                             | 147 |
| 5.2.2 Affine Transform Normalisation                                                                              | 148 |
| 5.2.3 Hausdorff Distance and Computational Complexity Reduction                                                     | 150 |</p>
<table>
<thead>
<tr>
<th>CONTENTS</th>
<th>ix</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.2.4 Single Resolution Search and Computational Complexity Analysis</td>
<td>151</td>
</tr>
<tr>
<td>5.3 Hierarchical Search</td>
<td>153</td>
</tr>
<tr>
<td>5.3.1 Global Precedence Effect</td>
<td>154</td>
</tr>
<tr>
<td>5.3.2 Multiresolution Segmentation and Multi-Level Search</td>
<td>154</td>
</tr>
<tr>
<td>5.4 Experimental Results and Discussion</td>
<td>158</td>
</tr>
<tr>
<td>5.4.1 “Head-and-Shoulders”</td>
<td>159</td>
</tr>
<tr>
<td>5.4.2 Template Design</td>
<td>169</td>
</tr>
<tr>
<td>5.4.3 Natural Objects Extraction Examples</td>
<td>173</td>
</tr>
<tr>
<td>5.5 Conclusion</td>
<td>183</td>
</tr>
<tr>
<td>6 Multiresolution Scalable Video Object Extraction</td>
<td>189</td>
</tr>
<tr>
<td>6.1 Introduction</td>
<td>189</td>
</tr>
<tr>
<td>6.2 Global Motion Estimation</td>
<td>190</td>
</tr>
<tr>
<td>6.3 Single Resolution Markov Random Field Modelling</td>
<td>192</td>
</tr>
<tr>
<td>6.3.1 Smoothness Factor</td>
<td>195</td>
</tr>
<tr>
<td>6.4 Multiresolution Scalable Video Segmentation</td>
<td>197</td>
</tr>
<tr>
<td>6.4.1 Objective Function Optimisation</td>
<td>198</td>
</tr>
<tr>
<td>6.5 Object’s Border Fine Tuning</td>
<td>200</td>
</tr>
<tr>
<td>6.6 Experimental Results and Discussion</td>
<td>202</td>
</tr>
<tr>
<td>6.7 Conclusion</td>
<td>214</td>
</tr>
<tr>
<td>7 Summary, Conclusions and Future Research</td>
<td>216</td>
</tr>
<tr>
<td>7.1 Introduction</td>
<td>216</td>
</tr>
<tr>
<td>7.2 Summary and Conclusions</td>
<td>217</td>
</tr>
<tr>
<td>7.3 Future Research</td>
<td>220</td>
</tr>
</tbody>
</table>
List of Figures

1.1 Hall Monitor sequence object extraction at frame 34: (a) object is extracted with visual quality constraint; (b) object is extracted by a regular algorithm [1]. .................................................. 6

2.1 (a) The Camera Man original image; (b) the extracted canny edges; (c) the extracted sobel edges. .............................................................. 18

2.2 Watershed explanation: catchment basins represent regions and watershed dams create contours. .............................................................. 22

2.3 (a) Camera Man gradient image; (b) Camera Man watershed basins; (c) Camera Man basins shown over the original image. .............. 23

2.4 Normal one and two pixel cliques at 8 neighbourhood system for potential function of Gibbs distribution ............................................. 27

2.5 Aperture problem; the movement of two identical blocks upward has two explanations: (a) both of the blocks moved up in the consecutive frame; (b) or diagonally switched places. Figure from [2]........... 60

2.6 Occlusion problem: incorrect motion vector will be assigned to the covered and uncovered background area. [3]. ............................... 61

3.1 Decoding of an object-based spatially scalable codestream. Depending on the channel, monitor, etc., a resolution is selected and the proper codestream is sent to the decoder. ................................. 87

3.2 Decomposition of a non rectangular object with odd-length filters: (a) the object, shown in dark grey; (b) the decomposed object after horizontal filtering; (c) decomposed object after vertical filtering. The letters “E” and “O” indicate the position (even or odd) of a pixel in the horizontal and vertical dimensions. ................................. 88
3.3 A 4 level pyramid where some of the corresponding pixels are shown by the similar colour and the dashed lines also connect them. Set $V_4$ shows 4 corresponding pixels at different resolutions. $V_3$ shows 3 corresponding pixels, and $V_2$ shows 2 corresponding pixels. $V_1$ is a pixel at highest resolution which has no corresponding pixel at the other resolutions. The number of corresponding pixels depends on the pixels indices.

3.4 Flow chart of the proposed multiresolution segmentation algorithm.

3.5 (a) The projected segmentation at the current resolution; (b) sub-region $D$ in region $A$ is detected; (c) regions $B$, $C$ and $D$ are merged.

3.6 Projection to catchment basins of Lena $256 \times 256$; (a) Lena image segmentation at $128 \times 128$; (b) Catchment basins imposed on the Lena image at $256 \times 256$; (c) The projected basin by the label of corresponding pixels of the lower level. The thick borders are the results of lower level borders projection to higher level.

3.7 (a) The reduced number of catchment basins projected from the lower level segmentation pixels; (b) merge of basins with the same label imposed on the main image, unlabelled basins in projection are shown with white colour; (c) the final projected segmentation to $256 \times 256$ level.

3.8 Segmentation of Lena in three resolutions: (a) $64 \times 64$; (b) $128 \times 128$; (c) $256 \times 256$ (d) The Lena Image at $256 \times 256$ resolution.

3.9 Segmentation of the 5th frame of SIF sequence Table_Tennis in three resolutions: (a) $60 \times 88$; (b) $120 \times 176$ and (c) $240 \times 352$ segmentation, (d) the original image of the 5th frame of SIF sequences table tennis.

3.10 A neighbourhood system in the pyramid: $V_1$ is a vector of corresponding pixels at three resolutions. $V_2$ is a neighbouring vector of $V_1$. Dashed lines connect the corresponding pixels of vectors.

3.11 (a) Normal one and two pixels cliques sets defined at a single resolution; (b) A clique of two vectors with the vector dimensions equal to three.

3.12 Table_Tennis image segmentation with $k = 6$ clusters and $\beta = 100$: (a) the main image; (b) single level segmentation.

3.13 Multiresolution segmentation of Table_Tennis image with $k = 6$ clusters and $\beta = 100$: (a) $240 \times 352$ segmentation; (b) $120 \times 176$ segmentation; (c) $60 \times 88$ segmentation.
3.14 Scalable multiresolution segmentation of Table_Tennis image with $k = 6$ clusters and $\beta = 100$: (a) $240 \times 352$ segmentation; (b) 120 $\times$ 176 segmentation; (c) 60 $\times$ 88 segmentation. ............... 105

3.15 Clair image segmentation with $k = 5$ clusters and $\beta = 50$: (a) the main image; (b) regular single resolution segmentation. .................. 107

3.16 Clair image segmentation by the scalable segmentation algorithm with $k = 5$ cluster and $\beta = 50$: (a) $288 \times 352$ segmentation; (b) 144 $\times$ 176 segmentation; (c) 72 $\times$ 88 segmentation. ............... 107

3.17 Clair image segmentation by the regular multiresolution segmentation algorithm with $k = 5$ cluster and $\beta = 50$: (a) $288 \times 352$ segmentation; (b) 144 $\times$ 176 segmentation; (c) 72 $\times$ 88 segmentation. ............... 108

3.18 (a) Noisy image of Clair, with a uniform noise in the range $(-30, +30)$ added to the images; (b) single resolution image segmentation; (c) scalable multiresolution image segmentation. ............... 108

3.19 Clair object extraction: (a) First selection; (b) object at $288 \times 352$; (c) object at 144 $\times$ 176; (d) object at 72 $\times$ 88. ....................... 110

3.20 Multiresolution Clair image segmentation with $k = 5$ clusters: (a) original Image; (b) regular multiresolution segmentation [4]; (c) proposed scalable multiresolution segmentation. ....................... 111

3.21 Three images segmentation: (a) first frame of QCIF sequence carphone image; (b) carphone segmentation with $k = 7$ and $\beta = 50$; (c) frame 50th of the CIF size sequence Hall_Monitor; (d) Hall_Monitor segmentation with $k = 6$ and $\beta = 50$; (e) 256 $\times$ 256 Barbara image; (f) Barbara image segmentation. ....................... 112

4.1 Circles at different resolutions. Down sampling of pixels at higher resolution with even indexes, creates the shapes at lower resolution: (a) closer approximation of a digital circle at high resolution; (b) down sampling to low resolution; (c) worse approximation of a digital circle at high resolution; (d) down sampling of (c) to low resolution. 132

4.2 Curvature estimation: (a) corner point, $k=90$; (b) same direction $k=0$; (c) change direction point $k=45$. ....................... 133

4.3 Scalable segmentation of a digital circle with an emphasis on low level smoothness: (a) original image; (b) noisy image; (c) segmentation at $20 \times 20$ resolution. (d) segmentation at $10 \times 10$ resolution. ....................... 134
4.4 Frame 5 of the Miss America CIF colour sequence segmentation (a) original image; (b) regular single resolution segmentation; (c) scalable segmentation; (d) scalable segmentation with smoothness constraint. ........................................ 135

4.5 Frame 5 of the Miss America CIF grey sequence segmentation (a) original image; (b) regular single resolution segmentation; (c) single resolution segmentation in RGB colour space; (d) scalable segmentation with smoothness constraint in YUV colour space. ............. 136

4.6 The edges extracted from the segmented image of Miss America by the Canny edge extraction algorithm: (a) edges of single resolution grey segmentation; (b) edges of scalable grey segmentation; (c) edges of single resolution colour segmentation in RGB space; (d) edges of scalable colour segmentation with smoothness constraint in YUV space. ................................................................. 137

4.7 Guitar segmentation: (a) The 256×256 grey image: (b) single resolution segmentation; (c) multiresolution 64×64 segmentation; (d) multiresolution 128×128 segmentation; (e) multiresolution 256×256 segmentation. ................................................................. 138

4.8 Guitar segmentation: (a) the 256×256 grey image of Guitar; (b) single resolution segmentation; (c) multiresolution segmentation at 64×64; (d) multiresolution segmentation at 128×128; (f) scalable segmentation at 64×64; (g) scalable segmentation at 128×128; (h) scalable segmentation at 256×256. ................................................................. 139

4.9 Scalable segmentation of colour Office image: (a) original office grey-level image at 256×256; (b) regular grey-level single resolution segmentation; (c) colour image of Office; (d) single resolution colour segmentation; (e) MRes segmentation at 64×64; (f) MRes segmentation at 128×128; (g) MRes segmentation at 256×256. ................................................................. 140

4.10 Multiresolution Office segmentation by the proposed scalable segmentation: (a) 64×64; (b) 128×128; (c) 256×256. ................................. 141

4.11 Frame 34 of Mother and Daughter QCIF sequence segmentation with \( k = 7, 2, 2 \) clusters and \( \beta = 40 \): (a) original grey-level image; (b) regular grey-level single resolution segmentation; (c) colour image of Mother and Daughter where U and V are in half resolution; (d) proposed scalable segmentation. ................................................................. 141

4.12 Lena colour image at 256×256 resolution. ................................. 142
4.13 Lena image segmentation at $256 \times 256$ with $k = 6, 4, 4$ clusters and $\beta = 100$: (a) regular single resolution segmentation where U and V are projected to higher resolution; (b) proposed scalable segmentation where U and V are in half resolution.

4.14 Segmentation of frame 32 of Foreman QCIF sequence with $k=10,4,2$ clusters at different colour channels: (a) original image; (b) scalable segmentation where U and V are projected to higher resolution; (c) Regular multiresolution segmentation; (d) scalable segmentation where YUV are in full resolution; (e) scalable segmentation in YUV space where U and V are in half resolution; (f) scalable segmentation in RGB space where components are in full resolution.

4.15 Car segmentation: (a) original Car colour image at $256 \times 256$; (b) proposed scalable colour segmentation; (c) original lifting colour image at $288 \times 216$; (d) highest resolution segmentation by scalable algorithm; (e) original House colour image at $192 \times 256$; (f) highest resolution House image segmentation by scalable algorithm.

5.1 (a) a shape with its major axis is shown; (b) another shape which has its major axis in a horizontal direction; (c) a typical shape rotation normalisation. The shape is rotating $\alpha$ degrees to coincide with the direction of the major axis of the second shape. The shape bounding box is also determined; (d) rotation normalised and flipped horizontally; (e) shape c is rotated $180^\circ + \alpha$; (f) shape c is rotated $180^\circ + \alpha$ and then flipped horizontally.

5.2 The shape bounding box is graded to reduce the number of borders pixels: (a) the grid; (b) borders grids are replaced with pixels.

5.3 The image is segmented into four regions. All regions are connected.

5.4 The Global Precedence effect. The ‘E’ character will be seen before the ‘H’ characters.

5.5 The hierarchical stack or irregular pyramid segmentation corresponding to the pyramid segmentation. The lowest resolution segmentation includes 2 regions and the similar segmentation map at the top level of the irregular pyramid can be seen. Similarly, the other corresponding segmentations at different levels of the regular and irregular segmentation pyramids have similar segmentation maps.

5.6 Irregular Pyramid (IRPRMD) creation algorithm. From highest resolution to lowest resolution, different levels of the irregular pyramid are created from the regular pyramid (RPRMD) segmentation.
5.7 Multiresolution search for the "object-of-interest" extraction

5.8 Clair original image, its scalable segmentation (SSeg) and hierarchical segmentation (HSeg) at different resolutions. The hierarchical segmentation images are just after the scalable segmentation at each resolution: (a) the original Clair image at 288 × 352 resolution; (b) 288 × 352 SSeg; (c) 144 × 176 SSeg; (d) HSeg corresponding to 144 × 176; (e) 72 × 88 SSeg; (f) HSeg corresponding to 72 × 88; (g) 36 × 44 SSeg; (h) HSeg corresponding to 36 × 44; (i) 18 × 22 SSeg; (j) HSeg corresponding to 18 × 22; (k) 9 × 11 SSeg; (l) HSeg corresponding to 9 × 11; (m) 5 × 6 SSeg; (n) HSeg corresponding to 5 × 6; (o) 3 × 3 SSeg; (p) HSeg corresponding to 3 × 3; (q) 2 × 2 SSeg; (r) HSeg corresponding to 2 × 2.

5.9 (a) Clair “head-and-shoulders” template; (b) texture of the extracted object; (c) the extracted object’s shape; (d) match between the template and the region, where the candidate region is drawn over the template; (e) template is over candidate region; (f) texture of the second candidate region; (g) candidate region’s shape; (h) match between the template and the candidate region, where the candidate region is drawn over the template; (i) template is over region; (j) match between the rotated template and the candidate region, where the candidate region is drawn over the template; (k) rotated template is over regions.

5.10 Foreman original image, its scalable segmentation (SSeg) and hierarchical segmentation (HSeg) at different resolutions. The hierarchical segmentation images are just after the scalable segmentation at each resolution: (a) the original image at 288 × 352 resolution; (b) 288 × 352 SSeg; (c) 144 × 176 SSeg; (d) HSeg corresponding to 144 × 176; (e) 72 × 88 SSeg; (f) HSeg corresponding to 72 × 88; (g) 36 × 44 SSeg; (h) HSeg corresponding to 36 × 44; (i) 18 × 22 SSeg; (j) HSeg corresponding to 18 × 22; (k) 9 × 11 SSeg; (l) HSeg corresponding to 9 × 11; (m) 5 × 6 SSeg; (n) HSeg corresponding to 5 × 6; (o) 3 × 3 SSeg; (p) HSeg corresponding to 3 × 3; (q) 2 × 2 SSeg; (r) HSeg corresponding to 2 × 2.

5.11 (a) The extracted Foreman “head-and-shoulders” shape; (b) The extracted Foreman "head-and-shoulders" texture; (c) template; (d) match between the template and the region, where the candidate region is drawn over the template; (e) template is over candidate region; (f) a (rejected) candidate region; (g) match between the template and the (rejected) candidate region, where the candidate region is drawn over the template; (h) template is over the (rejected) region.
5.12 Office original image with its scalable segmentation (SSeg) and hierarchical segmentation (HSeg) at different resolutions. The HSeg images are just after the SSeg at any resolution; (a) The original image at $256 \times 256$ resolution; (b) $256 \times 256$ SSeg; (c) $128 \times 128$ SSeg; (d) HSeg corresponding to $128 \times 128$; (e) $64 \times 64$ SSeg; (f) HSeg corresponding to $64 \times 64$; (g) $32 \times 32$ SSeg; (h) HSeg corresponding to $32 \times 32$; (i) $16 \times 16$ SSeg; (j) HSeg corresponding to $16 \times 16$; (k) $8 \times 8$ SSeg; (l) HSeg corresponding to $8 \times 8$; (m) $4 \times 4$ SSeg; (n) HSeg corresponding to $4 \times 4$; (o) $2 \times 2$ SSeg; (p) HSeg corresponding to $2 \times 2$. ................................. 170

5.13 (a) Office original image where the face area is extracted; (b) the bounding box which includes the "head-and-shoulders"; (c) The processed image area which includes the "head-and-shoulders"; (d) The bounding box for the "head-and-shoulders" is drawn where the ratio between the face and the bounding box determines the width and length. ................................. 171

5.14 (a) The extracted (Office) Fardin’s "head-and-shoulders" texture; (b) the shape of Fardin head and shoulders; (c) template; (d) match between the template and the region, where the candidate region is drawn over the template; (e) template is over candidate region; (f) Fardin "head-and-shoulders" mixed with a small part of the background (rough object); (g) shape of the rough object; (h) match between the candidate rough region and the template, where the candidate region is drawn over the template. ................................. 172

5.15 (a) The very short "head-and-shoulders" template; (b) short template; (c) medium template; (d) long template; (e) Clair object match, shown over very Short template; (f) Clair object match shown under very Short template; (g) over short template; (h) under short template; (i) over medium template; (j) under medium template; (k) over long template; (l) under long template; (m) Foreman object match shown over very Short template; (n) under very Short template; (o) over short template; (p) under short template; (q) over medium template; (r) under medium template; (s) over long template; (t) under long template. ................................. 174

5.16 (a) Fardin object match shown over very short template; (b) under very short template; (c) over short template; (d) under short template; (e) over medium template; (f) under medium template; (g) over long template; (h) under long template; ................................. 175
5.17 Car original image with its scalable segmentation (SSeg) and hierarchical segmentation (HSeg) at different resolutions. The HSeg images are just after the SSeg at each resolution: (a) The original image at $256 \times 256$ resolution; (b) $256 \times 256$ SSeg; (c) $128 \times 128$ SSeg; (d) HSeg corresponding to $128 \times 128$; (e) $64 \times 64$ SSeg; (f) HSeg corresponding to $64 \times 64$; (g) $32 \times 32$ SSeg; (h) HSeg corresponding to $32 \times 32$; (i) $16 \times 16$ SSeg; (j) HSeg corresponding to $16 \times 16$; (k) $8 \times 8$ SSeg; (l) HSeg corresponding to $8 \times 8$; (m) $4 \times 4$ SSeg; (n) HSeg corresponding to $4 \times 4$; (o) $2 \times 2$ SSeg; (p) HSeg corresponding to $2 \times 2$.

5.18 (a) The Car template; (b) The extracted car shape at $2 \times 2$ resolution; (c) the extracted car shown with its texture; (d) match between the template and the object, where the candidate region is drawn over the template; (e) template is over candidate region; (f) the extracted car shape at $16 \times 16$ resolution; (g) the extracted car shown with its texture; (h) match between the candidate rough region and the template, where the candidate region is drawn over the template; (i) template is over candidate region; (j) a candidate region at $16 \times 16$ resolution; (k) the candidate region shown with its texture; (l) match between the template and the object, where the candidate region is drawn over the template; (m) template is over candidate region.

5.19 (n) a different car template includes the car’s wheels; (o) the object extracted by the new template at $16 \times 16$ resolution; (p) match between the template and object, where the candidate region is drawn over template; (q) template is over candidate region.

5.20 Table_Tennis original image with its scalable segmentation (SSeg) and hierarchical segmentation (HSeg) at different resolutions. The HSeg images are just after the SSeg at each resolution: (a) the original image at $240 \times 352$ resolution; (b) $240 \times 352$ SSeg; (c) $120 \times 176$ SSeg; (d) HSeg corresponding to $120 \times 176$; (e) $60 \times 88$ SSeg; (f) HSeg corresponding to $60 \times 88$ SSeg; (g) $30 \times 44$ SSeg; (h) HSeg corresponding to $30 \times 44$; (i) $15 \times 22$ SSeg; (j) HSeg corresponding to $15 \times 22$; (k) $8 \times 11$ SSeg; (l) HSeg corresponding to $8 \times 11$; (m) $4 \times 6$ SSeg; (n) HSeg corresponding to $4 \times 6$; (o) $2 \times 3$ SSeg; (p) HSeg corresponding to $2 \times 3$.

5.21 (a) The ball template; (b) The extracted ball shape at $15 \times 22$ resolution; (d) match between the template and the extracted ball, where the candidate region is drawn over the template; (e) template is over candidate region.
5.22 Tools original image with its scalable segmentation (SSeg) and hierarchical segmentation (HSeg) at different resolutions. The HSeg images are just after the SSeg at each resolution: (a) The original image at 256 × 256 resolution; (b) 256 × 256 SSeg; (c) 128 × 128 SSeg; (d) HSeg corresponding to 128 × 128 SSeg; (e) 64 × 64 SSeg; (f) HSeg corresponding to 64 × 64 SSeg; (g) 32 × 32 SSeg; (h) HSeg corresponding to 32 × 32; (i) 16 × 16 SSeg; (j) HSeg corresponding to 16 × 16; (k) 8 × 8 SSeg; (l) HSeg corresponding to 8 × 8; (m) 4 × 4 SSeg; (n) HSeg corresponding to 4 × 4; (o) 2 × 2 SSeg; (p) HSeg corresponding to 2 × 2. 183

5.23 The flexible screwdriver template. 184

5.24 (a) The spanner template; (b) extracted spanner shape; (c) match between template and the extracted spanner, where the candidate region is drawn over the template; (d) template is over candidate spanner region. 184

5.25 (a) The screwdriver long template; (b) normal template; (c) short template; (d) extracted normal screwdriver shape; (e) extracted normal screwdriver texture; (f) match between the long template and the extracted normal screwdriver, where the candidate region is drawn over the template; (g) normal template is over candidate region; (h) match between the normal template and the extracted normal screwdriver, where the candidate region is drawn over the template; (i) normal template is over candidate region; (j) match between the short template and the extracted normal screwdriver, where the candidate region is drawn over the template; (k) short template is over candidate region; (l) extracted short screwdriver shape; (m) extracted short screwdriver texture; (n) match between the short template and the extracted short screwdriver, where the candidate region is drawn over the template; (o) short template is over candidate region. 187

5.26 (p) match between the normal template and the extracted short screwdriver, here the candidate region is drawn over the template; (q) normal template is over candidate region; (r) match between the short template and the extracted short screwdriver, where the candidate region is drawn over the template; (s) short template is over candidate. 188

6.1 Similarity function [5]. 195

6.2 The processed regions are shaded. (a) Adding the shaded region to the object area increases the smoothness; (b) Adding the shaded region to the object area decreases the smoothness. 196

6.3 Detection of uncovered background [6]. 201
6.4 Partitioning the segmentation regions to the basins: (a) The original (scalable) image segmentation; (b) Partitioning the segmentation regions to the basins. Regions $R_2$ and $R_3$ are smaller than the predefined threshold and have not been divided to basins. 202

6.5 Clair sequence object separation: (a) object extracted in frame number 20; (b) object extracted in frame number 45; (c) object extracted in frame number 65. 203

6.6 Clair object 23rd frame: (a1) scalable $288 \times 352$; (b1) scalable $144 \times 176$; (c1) scalable $72 \times 88$; (a2) regular $288 \times 352$; (b2) regular $144 \times 176$; (c2) regular $72 \times 88$. 205

6.7 Table_Tennis object extraction: (a) frame 10; (b) frame 23; (c) frame 32. 206

6.8 Table_Tennis object 10th frame: (a1) scalable $240 \times 352$; (b1) scalable $120 \times 176$; (c1) scalable $60 \times 88$; (a2) regular $240 \times 352$; (b2) regular $120 \times 176$; (c2) regular $60 \times 88$. 206

6.9 Object extraction from noisy Table_Tennis sequence: (a) frame 14 at resolution $240 \times 352$; (b) scalable object extraction at resolution $240 \times 352$; (c) single level object extraction at resolution $240 \times 352$. 208

6.10 Hall_Monitor sequence object extraction at frame 40; (a) resolution $288 \times 352$; (b) resolution $144 \times 176$; (c) resolution $72 \times 86$. 209

6.11 Hall_Monitor sequence object extraction: (a1) scalable extraction at frame 34; (b1) scalable extraction at frame 44; (c1) scalable extraction at frame 60; (a2) regular extraction at frame 34; (b2) regular extraction at frame 44; (c2) regular extraction at frame 60. 209

6.12 Mother & Daughter sequence object extraction at frames (a) 32; (b) 50; (c) 68. 210

6.13 Mother & Daughter sequence object extraction by the proposed scalable algorithm and regular tracking algorithm at different frames: (a) scalable at frame 48; (b) scalable at frame 58; (c) scalable at frame 72; (d) regular tracking algorithm at frame 48; regular tracking algorithm at frame 58; regular tracking algorithm at frame 72. 211

6.14 Foreman image sequence object extraction at frame (a) 5; (b) 20; (c) 30. 211
6.15 Foreman image sequence object extraction at frame 18 by the proposed scalable segmentation algorithm: (a) extracted object at $288 \times 352$ resolution; (b) extracted object at $144 \times 176$ resolution; (c) extracted object at $72 \times 88$ resolution. ........................................... 212

6.16 Foreman image sequence object extraction at frames (a) 8; (b) 28; (c) 42. ................................................................. 212
List of Tables

3.1 Number of regions in Table_Tennis segmentation. .................... 106
3.2 Number of regions in Clair image segmentation. ....................... 108
3.3 Misclassified pixels in noisy images. ................................. 109
3.4 Number of misclassified pixels for Clair and Table_Tennis image segmentation. ................................................. 109
4.1 Number of regions for segmentation of grey Miss America ......... 124
4.2 Number of regions for segmentation of colour Miss America ..... 124
4.3 Miss America grey segmentation smoothness ......................... 125
4.4 Miss America colour segmentation smoothness ....................... 125
4.5 Number of regions for segmentation of grey Guitar ............ 125
4.6 Number of regions for segmentation of colour Guitar ........ 126
4.7 Misclassified pixels in Foreman image segmentation ............. 128
5.1 Number of regions and region combinations at different resolutions . 161
5.2 Number of regions and region combinations at different resolutions . 164
5.3 Number of regions and region combinations at different resolutions . 169
5.4 Number of regions and region combinations at different resolutions . 169
5.5 Hausdorff distance of objects from ”head-and-shoulders” templates. 173
5.6 Number of regions and region combinations at different resolutions . 176
<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.7</td>
<td>Number of regions and region combinations at different resolution</td>
<td>179</td>
</tr>
<tr>
<td>5.8</td>
<td>Number of regions and region combinations at different resolutions</td>
<td>182</td>
</tr>
<tr>
<td>5.9</td>
<td>Hausdorff Distances of Tools from templates</td>
<td>182</td>
</tr>
<tr>
<td>6.1</td>
<td>Clair sequence smoothness</td>
<td>204</td>
</tr>
<tr>
<td>6.2</td>
<td>Table_Tennis sequence smoothness</td>
<td>206</td>
</tr>
<tr>
<td>6.3</td>
<td>Noisy Table_Tennis smoothness</td>
<td>207</td>
</tr>
<tr>
<td>6.4</td>
<td>Misclassified object’s pixels in noisy Table_Tennis</td>
<td>207</td>
</tr>
<tr>
<td>6.5</td>
<td>Hall_Monitor smoothness</td>
<td>207</td>
</tr>
<tr>
<td>6.6</td>
<td>Foreman sequence smoothness</td>
<td>212</td>
</tr>
<tr>
<td>6.7</td>
<td>Details of the proposed scalable video segmentation algorithm</td>
<td>214</td>
</tr>
<tr>
<td>6.8</td>
<td>Running times for the proposed and the alternative algorithms, performed</td>
<td>214</td>
</tr>
</tbody>
</table>
## List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-D</td>
<td>One-Dimensional</td>
</tr>
<tr>
<td>2-D</td>
<td>Two-Dimensional</td>
</tr>
<tr>
<td>3-D</td>
<td>Three-Dimensional</td>
</tr>
<tr>
<td>bpp</td>
<td>Bits per pixel</td>
</tr>
<tr>
<td>CFMRF</td>
<td>Compound Gauss Markov Random Field</td>
</tr>
<tr>
<td>CIF</td>
<td>Common Intermediate Format</td>
</tr>
<tr>
<td>DF</td>
<td>Difference frame</td>
</tr>
<tr>
<td>DWT</td>
<td>Discrete Wavelet Transform</td>
</tr>
<tr>
<td>EM</td>
<td>Expectation Maximization</td>
</tr>
<tr>
<td>GPE</td>
<td>Global Precedence Effect</td>
</tr>
<tr>
<td>HCF</td>
<td>Highest confident First</td>
</tr>
<tr>
<td>HMIS</td>
<td>Hierarchical Multiresolution Image Segmentation</td>
</tr>
<tr>
<td>HCMIS</td>
<td>Highly Correlated Multiresolution Image Segmentation</td>
</tr>
<tr>
<td>HMM</td>
<td>Hidden Markov Model</td>
</tr>
<tr>
<td>HSeg</td>
<td>Hierarchical Segmentation</td>
</tr>
<tr>
<td>HS-SPIHT</td>
<td>Highly Scalable Set Partitioning in Hierarchical Trees</td>
</tr>
<tr>
<td>HVS</td>
<td>Human Visual System</td>
</tr>
<tr>
<td>ICM</td>
<td>Iterated Condition Mode</td>
</tr>
<tr>
<td>ICE</td>
<td>Iterative Conditional Estimation</td>
</tr>
<tr>
<td>IRPRMD</td>
<td>Irregular Pyramid</td>
</tr>
<tr>
<td>JPEG</td>
<td>Joint Photographic Experts Group</td>
</tr>
<tr>
<td>MAP</td>
<td>Maximum A Posteriori</td>
</tr>
<tr>
<td>MDL</td>
<td>Minimum Description Length</td>
</tr>
<tr>
<td>Min</td>
<td>Minute</td>
</tr>
<tr>
<td>MMPM</td>
<td>Multiresolution Maximization of the a Posteriori Marginal</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>MMRF</td>
<td>Multiresolution Markov Random Field</td>
</tr>
<tr>
<td>MPEG</td>
<td>Moving Picture Experts Group</td>
</tr>
<tr>
<td>MRes</td>
<td>Multi Resolution</td>
</tr>
<tr>
<td>MRF</td>
<td>Markov Random Field</td>
</tr>
<tr>
<td>MRSeg</td>
<td>Multi Resolution Segmentation</td>
</tr>
<tr>
<td>MCC</td>
<td>Moving Connected Component</td>
</tr>
<tr>
<td>OFC</td>
<td>Optical Flow Constraint</td>
</tr>
<tr>
<td>PSNR</td>
<td>Peak Signal to Noise Ratio</td>
</tr>
<tr>
<td>QCIF</td>
<td>Quarter-CIF</td>
</tr>
<tr>
<td>RAG</td>
<td>Region Adjacency Graph</td>
</tr>
<tr>
<td>RDWT</td>
<td>Reduced Discrete Wavelet Transform</td>
</tr>
<tr>
<td>Sec</td>
<td>Second</td>
</tr>
<tr>
<td>Seg</td>
<td>Segmentation</td>
</tr>
<tr>
<td>SIF</td>
<td>Source Input Format</td>
</tr>
<tr>
<td>SMAP</td>
<td>Sequential Maximum A Posteriori</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal to Noise Ratio</td>
</tr>
<tr>
<td>SPIHT</td>
<td>Set Partitioning in Hierarchical Trees</td>
</tr>
<tr>
<td>SRSeg</td>
<td>Single Resolution Segmentation</td>
</tr>
<tr>
<td>SRES</td>
<td>Single Resolution</td>
</tr>
<tr>
<td>SSeg</td>
<td>Scalable Segmentation</td>
</tr>
<tr>
<td>SSF</td>
<td>Scale Space Filtering</td>
</tr>
<tr>
<td>VOP</td>
<td>Video Object Plane</td>
</tr>
<tr>
<td>ZSM</td>
<td>Zero Tree Significance Map</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Motivation

With the great advances in digital technologies, including telecommunications and networks, more and more audiovisual information are produced and accessed by many users through media such as storage devices, digital television and networks especially the internet. The increasing popularity of multimedia applications calls for the development of image and video processing methods for effective distribution and representation of the visual information to provide new image/video services, such as interactivity, manipulation, editing, content-based access and scalability. To achieve these demands, image/video processing has moved away from block-based towards object-based techniques. Object oriented processing provides the great flexibility needed for new content-based services such as interactivity and manipulation. To this end, industrial standards which support object-based representation of audiovisual information were introduced by the Moving Pictures Expert Group (MPEG) [7]. MPEG-4 and MPEG-7 provide flexibility in manipulation, interactivity, editing, easier archiving and content-based access and retrieval from audiovisual databases [7, 8].

To enable the object-based image and video processing, semantic segmentation which decomposes the scene into meaningful objects is essential. Automatic semantic segmentation without human intervention or high level knowledge remains largely
Introduction

unsolved as a challenge in image processing. In semantic video segmentation, motion is an important key and moving objects are extracted successfully. However, semantic segmentation of stationary objects remains as challenging in video as it is in still images. Image segmentation often starts with a low level segmentation such as edge/region-based segmentation which decomposes the image into basic and primitive components such as edges or homogeneous regions. The low level segmentation reduces the data and simplifies the irrelevant information. It extracts the perceptually important information, such as colour, contrast, optical flow, etc., and removes the other information. This makes the next stage of processing much faster. The visual content is then interpreted using a higher level of segmentation processing. The most challenging aspect of this process is the fact that low level features do not lead to semantic objects directly, because a generic object may contain different grey-levels, colours, textures, motions, etc. The gap between meaningful objects and low level features makes automatic and comprehensive semantic segmentation a very difficult task, although not inherently impossible.

Although a great deal of research in segmentation has been carried out, no dominant solution for this task has emerged. The proposed methods, by and large, remain ad hoc with little underlying theoretical foundation. Furthermore, segmentation is inherently an ill-posed problem [9]. This means that there is no unique solution to solve the multi-faceted segmentation problem. Semantic objects have no unique definition and therefore, segmentation algorithms are application dependent. There are many different segmentation algorithms designed for specific problems with some simplified assumptions. Consequently in the object-based processing standards such as MPEG-4 and MPEG-7, the image/video segmentation standards have not been defined. On the other hand, segmentation is a first stage of processing for many image/video processing applications such as pattern recognition, image analysis and understanding, computer vision, image and video databases with content-based access, object-based coding. In particular, the new advances in networking and digital processing offer the potential for an explosion in multimedia applications over networks which require enabling object-based processing.

In conclusion, there is a wide area of segmentation applications. Therefore it is
necessary to present a flexible image/video segmentation algorithm which extracts meaningful objects from the scene for different applications. This is a very important and formidable task with high demands and requires a great deal of intensive research.

1.2 Statement of the Problem

Semantically meaningful image/video segmentation, known to be the bottleneck in image and video processing, is an active and challenging topic of research. At this stage of technology, perfect partitioning of a generic image/video into the semantic objects existing in the scene is far from reality. Therefore in this thesis some aspects of comprehensive image/video “object-of-interest” extraction processing including low and high level segmentation algorithms are considered. The scope for research in this topic is very wide; however, the concern here are three areas of research which the available segmentation algorithms have not been able to effectively resolve. Underpinning all these three areas is the concept of (spatial) scalability, where the “object-of-interest” is searched for and segmented in a hierarchy of resolution levels. The main focus of this thesis, therefore, embraces three areas:

1. An effective, reliable and scalable multiresolution segmentation useful for object extraction at different resolutions.
2. Enhancing the visual quality of the extracted objects.
3. Effective hierarchical semantic segmentation.

Since segmentation is application dependent and considering the importance of coding for information distribution over networks, in this thesis special attention is given to the application of the proposed segmentation algorithms with scalable wavelet-based object coding algorithms, although the results are useful for generic applications.
1.2.1 Effective, Reliable and Scalable Segmentation

Traditional multiresolution segmentation algorithms in the literature are progressive and segment the image from the lowest resolution towards the highest resolution. The result at lower resolution is refined further at the next higher resolution until the highest resolution is segmented and the final result is obtained at the highest resolution. One of the challenges arising from this approach is that higher resolution segmentations are more rigorous than the lower resolution segmentations and segmentation maps at higher and lower resolutions are not quite identical. For example, some objects are not detected or are partly detected at low resolutions while they are perfectly detected at higher resolutions. This makes the higher resolution segmentation more reliable for object extraction applications. Therefore the semantic segmentation and object extraction algorithms extract objects from the highest resolution segmentation, and low level multiresolution segmentation algorithms are used to decrease the computational complexity, better capture the image structure, better noise tolerance, etc.

On the other hand, spatial scalability of object-based coding algorithms has opened a new application for multiresolution object extraction. In scalable object-based coding, a single codestream can be sent to different users with different processing capabilities and network bandwidths by selectively transmitting and decoding the related parts of the codestream. Some of the desirable scalable functionalities are signal to noise ratio (SNR), and spatial and temporal scalabilities [10, 11]. A scalable bitstream includes embedded parts that offer increasingly better SNR, greater spatial resolution or higher frame rates [10, 11]. Therefore considering the spatial scalability, which is the most requested kind of scalability, it is necessary to extract and present object shapes at different resolutions for the scalable object-based encoder/decoder systems. For effective scalable wavelet-based image/video object coding algorithms, maintaining the similarity of extracted objects’ shapes at different resolutions, the lower resolution object masks should be precisely the down-sampled versions from higher resolution [12].

It is therefore necessary to propose a multiresolution segmentation algorithm which
produces reliable and similar segmentation maps at different resolutions. To this end, a multiresolution segmentation algorithm is required where lower resolution segmentations are refined by higher resolutions as well as by the existing traditional refinement of higher resolutions by lower resolutions. Although there is no multiresolution segmentation algorithm in the literature that satisfies this requirement, segmentation algorithms which produce similar segmentation patterns at different resolutions are called scalable segmentation (SSeg) algorithms hereafter.

To produce the shape mask at different resolutions, one regular informally defined option is single level image/video segmentation where objects/regions are extracted at the highest resolution segmentation and then down-sampled onto the lower resolutions. However, this single resolution procedure fails to deal with the requirement of multiresolution scalable segmentation and extraction processes and loses the properties and advantages of multiresolution processing, such as less computational complexity, better capturing of the image structure and less noise sensitivity.

1.2.2 Enhancing the Visual Quality of the Extracted Shapes

In assessing the performance of the segmentation processes, traditionally, the main emphasis is placed on the statistical accuracy, while qualities such as well defined borders or visual merit of the extracted objects are not considered. Visual quality of the segmented objects, however, has great influence on the viewers. For example see Figure 1.1 where the objects are extracted by two different algorithms, one with a smoothness constraint as the visual quality criterion, and the other being a typical region-based video object extraction algorithm [1]. Therefore, as well as the statistical criteria, visual effect and quality criteria should be incorporated into the segmentation algorithms. In this thesis the visual quality pledge is extended to multiresolution segmentation where traditional algorithms result in visually unpleasant shapes at different resolutions.

1.2.3 Effective Hierarchical Semantic Object Segmentation

Generic semantic segmentation remains an elusive goal in the image processing research community. The task is complicated by the fact that most real life images
contain many objects in a cluttered background. To effectively tackle the task, it is broken down into various specific sub-tasks such as determining the foreground and background in a scene, or searching the scene based on the size of the “object-of-interest”. This process seems to be inherent in the human visual system (HVS) where a phenomenon called the global precedence effect (GPE) means that global (big picture) objects are processed first followed by local (fine detail) objects [13, 14]. In other words, in the visual processing, the global perception precedes the local analysis. For example the forest is seen before the trees, or car is detected before attention is drawn to its windows and wheels. This means that the low frequency visual perception paths precede the high frequency paths.

HVS is the best natural vision system for cognition and understanding, and simulating the features of HVS improves the efficiency of the segmentation and artificial vision systems. Therefore, inspired by the HVS, simulating the GPE will result in searching for big objects first, followed by search of smaller objects\(^1\). This can be a natural hierarchy for object examination/processing, which significantly reduces the computational complexity of the semantic segmentation algorithms. In image and video “object-of-interest” extraction algorithms, a hierarchical search similar to the

\(^{1}\)Note that an object can contain several homogeneous large and small size regions.
GPE has not been considered at all or considered effectively, and objects are searched for through the edges/regions of the segmented image with the same priority and high computational complexity [5, 15–18].

It is interesting to note that in this thesis the three above-mentioned areas, including scalable multiresolution segmentation, visual quality of the objects extracted, and the global precedence effect, are integrated as part of a cohesive process using multiresolution techniques.

1.3 Research Goals

Motivated by the wide usage and importance of the segmentation applications, the main goal of this project is to provide effective meaningful image and video segmentation algorithms which, in accordance with the human visual system, can extract visually pleasing “object(s)-of-interest” at different resolutions and are computationally simple. The results are applicable in object-based processing algorithms, in particular in (spatially) scalable wavelet-based image/video object coding. The combination of the following goals/aspects of this work sets it apart from other image and video segmentation algorithms:

- **Scalable multiresolution segmentation**: in scalable wavelet-based object coding algorithms, the object’s shape at different resolutions is coded with the constraint that the representation of the shape at different resolutions are similar. The required similarity means that the high resolution object masks are down-sampled to generate the corresponding low level object masks. In the segmentation algorithm, the required similarity should be considered as a constraint. This calls for a multiresolution analysis that keeps the similarity/scalability between different resolutions of the pyramid. This constraint is required for both image and video segmentation.

- **Reliable segmentation for multiresolution object extraction**: in ordinary multiresolution segmentation algorithms, the segmentation is progressive from low towards high resolution. However, considering the refinement of segmen-
tation at higher resolutions, the goal of presenting a more reliable segmentation leads to an algorithm which includes both low to high and high to low resolution influences in the segmentation result. The high to low resolution effect is a feedback from high to low resolution which influences low and consequently high resolution segmentation results. Implementation of the segmentation algorithm includes a loop from low to high and high to low feedback which continues until convergence.

- **Visual quality**: to have a favourable effect on the viewer, the extracted objects should be visually pleasing. In addition to statistical criteria, the visual quality criteria should influence the segmentation algorithm. This is needed to avoid semantic distortion visible to human visual or recognition systems. The attention to the visual quality extends to the multiresolution object extraction algorithm needed for scalable coding. Definition of an objective criterion for the visual quality, and a multiresolution framework to incorporate the visual criterion is the goal that can force the segmentation algorithm to extract visually pleasing objects/regions.

- **Hierarchical search for “object-of-interest” extraction**: consistency of the object extraction algorithm with the GPE feature of HVS is an important aim in the evolution of object-based segmentation and extraction algorithms. To consider the global precedence effect of the human visual system the large size objects (global and low resolution information) should be processed first, followed by the processing of the small size (local) objects. Implementing a hierarchical search through the image, which simulates the GPE in a multiresolution framework, is a goal that reduces the computational complexity of the semantic object extraction from natural images.

- **Reducing computational complexity**: traditionally, many of the segmentation algorithms are computationally complex. This renders them useless for the real time segmentation applications. Reduction of the computation complexity of the segmentation algorithm increases its applications especially for natural image and video processing, and also real time applications. It should be considered that adding features such as visual quality and reliable segmen-
The goal is to reduce the computational complexity of the proposed algorithms in the multiresolution framework as much as possible to make them practically applicable algorithms.

### 1.4 Thesis Outline

The thesis consists of seven chapters, including this introductory chapter. Most of the chapters start with an introduction and end with conclusion. The main topics of each chapter are explained below.

- **Chapter 1** introduces the research topic with its main goals, and provides contributions and publications related to the thesis.

- **Chapter 2** provides essential background on image and video segmentation. It starts with explanation about different approaches to image segmentation algorithms, and the two major region-based segmentation algorithms including morphological segmentation and Markov Random Field (MRF) based segmentation algorithms are discussed. The concept of multiresolution segmentation is then discussed, and the works about multiresolution segmentation in the literature are reviewed. Similarly, the concepts of semantic image segmentation and “object-of-interest” extraction are explained, and the outstanding works presented in the literature, are reviewed. Discussion on video starts with the motion concept and motion estimation. Occlusion and aperture problems are then explained. Finally, a development of video segmentation algorithms classification is introduced, and the different approaches of video segmentation algorithms are discussed. Some of the outstanding works in the literature are reviewed. The chapter concludes with the research direction where the appropriate approaches for achieving the major goals are selected.

- **Chapter 3** describes the scalability concept, and two novel multiresolution segmentation algorithms are introduced. First the scalability concept and the different kinds of scalability are briefly explained. The wavelet image decom-
position and down-sampling relation between objects at different resolutions are then described. Furthermore, a hierarchical morphology-based segmentation algorithm is proposed. The spatial scalability of this algorithm is analysed, and the shortcomings of the traditional hierarchical multiresolution segmentation algorithms are highlighted. A MRF-based scalable grey image segmentation algorithm is then introduced. This algorithm extends the regular single resolution clique concept to the multi-dimensional space of the pyramid. It also develops the objective function of the regular single resolution grey image segmentation algorithm [4], [19] to one suitable for scalable multiresolution segmentation. The optimisation method for this algorithm is explained in this chapter. The properties of the proposed scalable algorithm are compared with ordinary single and multiresolution segmentation algorithms.

- **Chapter 4** develops the scalable segmentation algorithm proposed in Chapter 3 further. It starts with discussion about the concept of visual quality of object/region segmentation, and introduces smoothness as the quantitative criterion for visual quality. The smoothness constraint is incorporated into the objective function of the segmentation algorithm. The scalable segmentation of grey-level images is extended into the colour space. The segmentation algorithm and results with different colour spaces are discussed. In particular, the segmentation of colour images in the databases is considered, where the chrominance components are presented in half-resolution.

- **Chapter 5** introduces a meaningful image object extraction algorithm. An affine invariant template matching is first proposed, and template searches through a single resolution segmented image with its computational complexity are described. To implement the global precedence effect and to achieve a reduction in computational complexity, a hierarchical template matching algorithm is introduced. First, as a result of the pyramidal scalable image segmentation, a hierarchy of fine resolution segmentation maps, organised in a stack, is introduced. The properties of the stack in deleting small size regions and high frequency components towards implementing the GPE is discussed. Finally, template searching in the stack is described, and the computational complexity
of the search with the proposed algorithm is also discussed.

- **Chapter 6** extends the regular single resolution video object extraction algorithms to scalable multiresolution mode. Moreover the region-based smoothness criterion is introduced, which contributes to the region classification decision and improves the visual quality of the extracted objects. First, a MRF-based backward video segmentation algorithm is introduced. Different terms of the objective function of the MRF classification algorithm, including temporal and spatial continuity, motion constraints and smoothness are explained. The development of the single resolution objective function to multiresolution segmentation is described, and optimisation of the objective function is explained. Motion validation for removing the occlusion problem is also discussed. The results are compared with the regular region-based video object extraction algorithms.

- **Chapter 7** summarises the thesis and draws the conclusions. Some directions for further research are suggested in this final chapter.

### 1.5 Major Contributions

The main contributions of this thesis are itemized as follows. They are presented according to the order they appear in the thesis.

1. While most of the multiresolution segmentation algorithms in the literature are MRF-based, a multiresolution image segmentation algorithm based on the morphological watershed operator and region merging is proposed. Smooth and well located borders in all resolutions of the wavelet pyramid are produced by matching the object/region borders to watershed contours. The projection of the lower resolution segmentation and refining it at uncertain areas around the projected border pixels significantly reduce the computational complexity of the segmentation algorithm. Detection of the new regions at the higher resolutions removes the over-segmentation associated with the regular multiresolution segmentation algorithm. Edge validity testing in the lowest resolution
segmentation, using the wavelet coefficients to define a criterion, reduces the number of regions and enables us to detect inhomogeneous regions.

2. Adding spatial scalability to the segmentation algorithm, which produces similar region patterns at different resolutions. A MRF-based multiresolution image segmentation algorithm is proposed which support the scalability. In the proposed algorithm, as well as maintaining the low resolution effect on high resolution segmentation of traditional multiresolution segmentation, a feedback from high to low resolution segmentation is introduced. These two-way effects make the segmentation more reliable, especially at low resolutions. This segmentation has the good features of both the single and the multiresolution segmentation algorithms. It detects more regions than ordinary multiresolution segmentation algorithms while avoiding over-segmentation, which is common in single resolution segmentation algorithms. It increases the grey-level variation detection but remains noise tolerant. The objects/regions produced, are usable for wavelet-based scalable image object coding algorithms, although they can also be useful for any generic segmentation applications.

3. To produce more visually pleasing objects/regions, a new quantitative criterion is incorporated in the segmentation algorithm. This criterion, which is a smoothness function based on the pixel segmentation labels, represents the visual quality of the objects/regions. Different smoothness coefficients considered at different resolutions reduce down-sampling distortion. The proposed smoothness definition is extended to region-based definition for video frame classifications. The subjective results confirm the correlation of the quantitative criterion with the visual quality concept.

4. The scalable grey image segmentation is extended into the colour space. In addition to having the advantages of the scalable grey image segmentation, the proposed scalable algorithm can segment colour images where the intensity component $Y$ is in full resolution and the chrominance components such as $U$ and $V$ are in half resolutions.

5. A novel “object-of-interest” extraction method is proposed which simulates the global precedence effect of the HVS. Scale and orientation invariant template
matching is introduced, and an irregular pyramid of segmentation maps organised in a stack is introduced, which presents a hierarchy of the segmentation maps with a gradually reduction of the number of regions to two. The template matching algorithm over the irregular pyramid simulates the GPE and significantly reduces the computational complexity of the search algorithm. Deformable templates and their matching are also discussed.

6. A MRF-based backward multiresolution region classification algorithm for video segmentation task is introduced. In addition to the temporal and spatial continuity and motion constraints, the region smoothness criterion is incorporated into the objective function of the classification algorithm, which improves the visual quality of the extracted objects. In addition to well defined object extraction at different resolutions, the proposed algorithm allows for larger motion, better noise tolerance and less computational complexity.

1.6 Publications

The following publications have been the result of the research presented in this thesis:


• F. Akhlaghian Tab, G. Naghdy and A. Mertins, “Scalable multiresolution image segmentation and its application in video object extraction algorithm,” Accepted in 2005 IEEE International Region 10 Conference (Tencon’ 05), Nov 21-24, Melbourne, Australia, 2005.


Chapter 2

Literature Review

2.1 Introduction

Visual processing is becoming increasingly important with the advance of broadband networks, high power workstations, and advanced imaging tools including digital cameras and scanners. Effective visual information management, including storage, retrieval, distribution and presentation, needs new object-based processing methods. Therefore, object-based image processing has been the topic of intensive research for many image and video processing applications such as image/video database management, retrieval, coding, editing, and interactive image manipulation. One of the main challenges for many object-based algorithms is semantic segmentation.

Low level image segmentation is a crucial initial step for the semantic image and video segmentation algorithms. The low level segmentation affects the accuracy and computational complexity of the high level segmentation. Therefore, in a full scenario object-based implementation, a well fitted low level segmentation is essential for efficient high level object detection.

This chapter provides a survey of the most important issues in the literature on low and high level image segmentation algorithms, video object tracking and segmentation algorithms. Section 2.2 is about low level image segmentation algorithms. It briefly explains the classification of different segmentation algorithms including edge-based and region-based approaches. In Section 2.3 two major region-based seg-
mentation approaches are discussed. Section 2.3.1 is on the theory of mathematical morphology and morphological segmentation, and in Section 2.3.2 Markov random field theory and Bayesian based segmentation algorithms are explained. Section 2.4 deals with the multiresolution segmentation algorithms, and the most outstanding works on multiresolution image segmentation algorithms are reviewed. Section 2.5 presents a discussion on the semantic image segmentation algorithms in the literature, including “object-of-interest” extraction and meaningful scene segmentation, and the outstanding works are reviewed. In Section 2.6 video segmentation algorithms are discussed, and the most important works on object tracking are reviewed. Finally a chapter summary, conclusion and research directions are given in Section 2.11. In this section, regards to the literature and the mentioned goals in the introduction chapter, the inferences for the selected approaches to solve the found gaps and achieving goals are explained.

2.2 Low Level Image Segmentation

Low level image segmentation is the first step in many image analysis tasks. Simply, the segmentation goal is to partition the image into regions that are correlated with the semantic objects or areas of the real world as much as possible. However, due to the lack of high level knowledge, the extracted regions do not directly correspond to the meaningful image objects. Therefore, in this stage a meaningful segmentation of the scene is not achieved. However, the substantial reduction in data volume achieved at this stage is very useful for the subsequent higher level processing. The extracted image regions are input materials for further analysis such as image understanding, scene interpretation and pattern recognition, etc.

Low level image segmentation algorithms are divided into two main classes, which are region-based and edge-based segmentation. These algorithms partition the image into regions or extract their edges. Regions are extracted based on the similarity while edges are found based on dissimilarity.
2.2.1 Edge-Based Segmentation

Edges and discontinuities are important in many image processing algorithms. One class of segmentation algorithms is based on using discontinuity of relevant features, such as grey-level, colour, texture, to establish the edge pixels. Discontinuity or abrupt changes of the relevant feature are detected with discrete differences or partial derivatives. The most important edge detector algorithms, such as the Sobel algorithm [20, 21] the Prewitt [20, 21] and the Fri-Chen operator [21, 22] examine the gradient function to find discontinuity. Change detector functions are very sensitive to noise, and the results needs to be filtered. For example, the non-maximal suppression filtering and criteria defined and applied by the widely known Canny edge detector [21, 23] have led it to be considered the best edge detector. Although the edge-based segmentation algorithms are often faster than region-based segmentation, they have some weaknesses which are:

- The produced edges are often unclosed contours. Of course there are some techniques for connecting the unclosed contours, which use the geometrical property of regions. Moreover to computational complexity, connecting based on the real contours cannot be guaranteed.

- Discontinuity is a local feature, and a small local error can have significant consequences. For example, a non-detected edge pixel can result in an unclosed contour.

- The width of an estimated transition can be more than one pixel, and thinning techniques must be employed to reduce the thickness to one pixel. However, this thinning may still not be accurate.

These problems limit the edge-base segmentation applications. In Figure 2.1 an image and its edges as extracted by Sobel and Canny edge detectors can be seen. More details about edge-based segmentation can be found in [24].

One of the problems with these algorithms relates to the thresholds used in the change detector. The edge-based algorithms, which are based on the threshold, can detect some non-edge pixels (over-segmentation) or adversely delete some edge pixels.
2.2.2 Region-Based Segmentation

These algorithms partition the image into regions with common features suitable for further analysis. The extracted regions are uniform with respect to some characteristic, such as intensity, colour, texture.

Homogeneity is the main criterion for region-based segmentation algorithms. However, similarity or homogeneity does not have a precise definition and, its value is determined depending on the algorithm, application, user, etc. Based on the different homogeneity thresholds and different segmentation algorithms, different segmentation results are obtained. Therefore there are many possible acceptable segmentation results, rendering the segmentation task an ill-posed problem [9].

2.3 Major Region-Based Segmentation Approaches

There are many region-based segmentation algorithms which can be classified into several classes such as the clustering methods [21, 25, 26], region growing [27–30], region split and merge [31–33], minimum description length (MDL) [34–36], mathematical morphology [37–40] and Bayesian based segmentation approached.
The survey of all these approaches is beyond the scope of this thesis and there are some textbooks which cover the introductory concepts of these approaches [21, 24]. The focus is placed on two approaches namely morphological and Bayesian based segmentation algorithms, which have proven to be more successful and have received more attention in recent years.

### 2.3.1 Mathematical Morphology

The word morphology stems from a branch of biology that deals with the geometry of animals and plants. In the same way, in image processing, the expression “mathematical morphology” is used, indicating a geometrical approach to image and video processing. It is used for extracting geometrical properties from image and video frames and has numerous applications in image processing and analysis [37–40]. Some of its applications are shape representation and description, automated industrial inspection, computer vision, shape recognition, enhancement and noise suppression, texture analysis, radar object detection and range imagery [46–51].

The main language of mathematical morphology is set theory and the key point is the representation of signals and systems in terms of sets and set transformations. These capabilities allow us to represent and manipulate geometrical structures in images and other signals [40]. For review of the concept of morphology and some of its applications refer to [37, 52].

#### 2.3.1.1 Morphological Segmentation

Morphological tools are used in morphological segmentation. The method looks like a region growing algorithm, starting from a set of markers for all zones of interest and extending to all pixels of the image. For morphological segmentation there are three main steps which are:

**Stage 1:** simplifying the image, such as removing the noise, which is important in preventing over-segmentation.
Stage 2: markers extraction, which is necessary in segmenting the patterns which must be extracted.

Stage 3: applying watershed algorithm. In the last stage this algorithm segments the image by using gradient and markers information.

These stages are further elaborated in the following sections.

2.3.1.2 Simplifying Image(s)

In this step, the image is simplified to remove non-useful small portion of information, which makes it easier to segment. The amount and nature of the information are controlled in this simplification. In particular, eliminating the noise and removing the very small regions are done in this step of the segmentation process.

The most classical simplification tool is the linear low pass filter. However, it is well known that this filter blurs edges and does not preserve contour information, which is important for the segmentation algorithm. Therefore, a simplification tool capable of preserving the object/region contours is required. Many nonlinear filters such as median, rank order and morphological filters have been proposed but they often degrade the 2-D signal. However, a class of Morphological filters, called filters by reconstruction, are very efficient for simplification, and can perform the task with contour preservation constraint. These morphological operators belong to the class of so-called connected operators. Details and analysis of these filters can be found in [53, 54].

2.3.1.3 Marker Extraction

Each marker is in the form of an initial seed for a region in the final segmentation and detects the presence of a region in segmentation. This step decides how many regions exist in the final partition. Ideally each marker corresponds to a meaningful object area. Marker extraction is not an easy task and often is dependent on a particular application. For example, sometimes it is done by using some external high level knowledge of the collection of images under study [54–56]. Although the initial shape, and position of markers are not very crucial, finding markers is a drawback
for morphological segmentation. There is no general theory for marker extraction, and often marker extraction has to be solved for each special case [56, 57]. Therefore more complex automatic marker extraction methods should be developed to find such markers automatically.

2.3.1.4 Watershed Algorithm

The main morphological tool for segmentation is a famous filter named the watershed. Its concept is simple and similar to region growing techniques where iteratively undecided pixels are assigned to a region [55, 57]. The morphological watershed filter segments the image into some homogeneous regions called catchment basins. If the 3D topographic surface image of a gradient function where the gradient values represent the altitudes is considered, region interiors correspond to catchment basins and region edges correspond to high parts of watershed dams. Therefore the region borders or watershed contours correspond to the high gradient values, and interior catchment basin regions correspond to low gradient values. The main feature of any catchment basin is that any pixel in a catchment basin is connected by a monotonic decreasing line of pixels to the minimum altitude (gradient) in the basin.

While the concept of watershed and catchment basin are clear, the implementation of watershed segmentation is a complex task. Many early implementations result in high computational complexity and inaccurate results [57, 58]. The algorithm presented by Vincent et al. [54] makes the idea practical. They start from flat zones\(^1\) as markers of regions. Then the borders are moved toward the watershed dams or the maximum gradient values.

The method of moving the borders of regions is an interesting feature of different watershed algorithms. Vincent et al. consider the morphological segmentation as a flooding procedure. Imagine that each minimum of the topographic surface of the gradient image is pierced, and that this surface is plugged into a lake with a constant vertical speed. The water entering through the holes floods the surface, and during the flooding, two or more floods coming from different minima may merge. This event is avoided by a dam, built on the points of the topography surface where the

\(^1\)Flat zones are regions with constant grey-level or zero gradient.
floods would merge. At the end of the process, each minimum is completely surrounded by dams, which delimit its associated catchment basin. These dams define the catchment basins and watershed contours. Figure 2.2 shows a 2-D topographic image of catchment basins and their dams. Dams are located on the local maximum of the gradient image. Therefore, dams or watershed contours determine the boundaries of regions resulting in image segmentation. In Figure 2.3, the Camera Man watershed regions are shown. For a more mathematical definition of the immersion process and for a fast implementation of the watershed algorithm refer to [59].

Now the watershed algorithm by using the extracted markers is reviewed. The topographic surface of the gradient image and watershed process is used, but instead of piercing the minima of this surface, holes are made only through the components of the extracted marker set. The flooding will invade the surface and produce as many catchment basins as there are markers in the marker set. The catchment basins of the minima which are not pierced are filled up by overflow of the neighbouring catchment basins: as soon as the water reaches the saddle point between basins, the water rushes through the pass and fills the so-far empty basin. No dam is constructed between such basins. A dam is only constructed for separating floods originating from different pierced minima [60].

For many images there is no clear algorithm to extract the markers of semantic objects. If the gradient flat zones are considered as markers many regions will be extracted, creating an over-segmentation as Figure 2.3(b) and (c) shows. One way to
solve this problem is to do a region merging after watershed segmentation. Similar regions are merged based on a predefined criterion. The merge procedure continues until a predefined number of regions remains, and there are no more similar regions to merge [61–63]. Based on this idea a novel multiresolution morphological based image segmentation algorithm will be proposed in Chapter 3.

### 2.3.2 Bayesian Inference Theory and Markov Random Fields

Bayesian theory is one of the most fundamental theories in probability with the widest applications in image processing, such as segmentation, restoration, motion estimation, computer vision, scene analysis and image understanding [6,41–45]. The Bayesian technique is based on Bayes formula, which is:

\[
P(X|Y) = 
\frac{P(Y|X)P(X)}{P(Y)}
\]  

(2.1)

The Bayesian applications rely on maximising the probability function of the unknown parameters \(X\), given the observed data \(Y\). The MAP estimation aims at maximising \(P(X|Y)\) with respect to \(X\), which is equal to maximising the right hand side of (2.1). In (2.1), the denominator is independent of \(X\), so it is enough to maximise the numerator of the Bayesian formula.
\[ P(X|Y) \propto P(Y|X)P(X) \]  

(2.2)

The pick of the conditional probability function gives the likelihood estimation of \( X \). For simplifying the notation it is more convenient to minimise the negative logarithm of \( P(X|Y) \). Therefore the following cost function is defined which should be minimised:

\[ Cost = -\log P(Y|X) - \log P(X) \]

So there is a need to estimate two probabilities \( P(X) \) and \( P(Y|X) \). Theoretical or experimental knowledge often determines the a priori probability \( P(X) \) \([41, 64, 65]\). It is known that the Gibbs distribution is one of the most popular choices for \( P(X) \) in image processing applications \([3, 41]\). This means that \( X \) is assumed to be a sample of Markov random field (MRF) variables. The conditional probability function \( P(Y|X) \) on the other hand describes how well \( X \) explains the observation \( Y \) and therefore can be viewed as an observation model. From combining the a priori knowledge and observation the posteriori probability \( P(X|Y) \) is obtained which is a measure of the goodness of fit of \( X \) to the data which is a criterion for the Bayesian inference. There are many applications for the estimation techniques in image processing and computer vision such as image restoration \([66–68]\) image segmentation \([19, 69–71]\), motion estimation \([72–74]\), etc. In these applications there is some degraded information such as \( Y \), and the unknown parameters \( X \) is estimated. In these algorithms Bayesian techniques are applicable. The details necessary in Bayesian inferences, in particular for segmentation applications, are explained in the following sections.

2.3.2.1 Markov Random Field (MRF)

A Markov chain \([75, 76]\) is a sequence of random variables \( X_1, X_2, \ldots \), each representing the states of some physical system. The primary characteristic of a Markov chain is:
or, in other words, the conditional probability of the current state, given all previous states, depends only on the most recent state. This is often referred to as the “one-sided” property [66]. The “two-sided” property can be represented as:

\[ P\{X_n = x_n|X_{n-1} = x_{n-1}, \ldots, X_1 = x_1\} = P\{X_n = x_n|X_{n-1} = x_{n-1}\} \]

or the conditional probability of the current state depends only on the previous state and the next state. This can be applied not only to a sequence of states in time, but also to an array of states in multidimensional space such as an image [3, 41]. In multidimensional space (such as an image) the inherent sequential time order is replaced with the neighbourhood concept. Therefore the term “Markov chain” is replaced with Markov mesh Random Field or, simply Markov Random Field (MRF) [77, 78]. It is the most important statistical model in image processing and computer vision, which can represent the spatial continuity that is inherent in natural images.

Let \( X \) be a two dimensional random field defined on \( \Omega = L = \{(i, j)|1 \leq i < M, \ 1 \leq j < N\} \). Further, let \( \Omega \) denotes the set of all possible realisations of \( X \). Then, \( X \) is a Markov random field (MRF) with respect to neighbourhood system \( N \) if:

\[ P(X(i, j)|X(k, l), all \ (k, l) \neq (i, j)) = P(X(i, j)|X(k, l), (k, l) \in N_{i,j}) \] (2.4)

where \( N_{i,j} \) in the above formula, describes the neighbours of \( N \) according to a neighbourhood system such as the 4 or 8 pixels neighbourhood system [3]. This property restricts the complexity of the statistical dependency of pixel \((i, j)\) on its neighbours or boundary set and thereby significantly reduces the complexity of the model.
2.3.2.2 The Gibbs Distribution

There is an important theorem related to MRF which implies that a random field $X$ is a MRF variable if and only if $P(X)$ can be written as a Gibbs distribution. That is:

$$P(X = x) = \frac{1}{Z} \exp\left(-\frac{1}{T} U(x)\right)$$  \hspace{1cm} (2.5)

This distribution was first used in physics and statistical mechanics. Due to its analogy to physical systems, $U(x)$ is called the energy function and $T$ corresponds to temperature. At a high temperature $T$, the system is melted, and all realisations $x \in \Omega$ are more or less equally probable. At low temperature, the system is forced to be in a state of low energy. Thus, in accordance with physical systems, a low energy level corresponds to high likelihood and vice versa. The constant $Z$ is a normalising factor and usually does not have to be evaluated. The energy function $U(x)$ is written as the sum of potential functions $V_C(x)$:

$$U(x) = \sum_{\text{all cliques } C} V_C(x)$$  \hspace{1cm} (2.6)

A clique $C$ is defined as a subset $C \subset L$ of an image that contains either a single pixel $x$ or several pixels that, according to the neighbourhood system, are all neighbours of pixel $x$. Figure 2.4 shows all cliques in a second order neighbourhood system. This shows that an energy function $U(x)$ and therefore the likelihood estimation of $P(x)$, consists of contributions from local interaction within cliques, which conforms with the MRF property of $X$ where pixels are statistically distributed depending only on their neighbours. By the Gibbs distribution, the MRF distribution, $P(X)$, is expressed as a combination of clique functions. In the next section, an example of the clique potential function and cost function for segmentation applications can be seen.
2.3.2.3 Bayesian Segmentation

Any segmentation technique which maximises the posteriori probability of the unknown segmentation field is a typical Bayesian segmentation approach. The most Bayesian segmentation algorithms are iterative and thus improve the result iteratively. At each iteration, the segmentation estimation is updated to decrease a cost function. Bayesian based segmentation algorithms can vary in the observation model $P(Y|X)$ and the choice of the energy function $U(X)$ for the Gibbs distribution $P(X)$. Some other details such as the estimation technique for the probability density function parameters, the neighbourhood system, and the numerical optimisation method can also be different. At this point, a typical Bayesian based segmentation algorithm similar to the well known algorithm proposed by Pappas [4] is described. According to the application, the proposed algorithm can be modified to better fit the task. As a first step, the cost function extraction is explained and then its iterative optimisation is described.

To each pixel $(i, j)$ a label $m \in \{0, \cdots, K - 1\}$ is assigned so that $X(i, j) = m$, means that the pixel with coordinates $(i, j)$ belongs to region $m$. One of the shortcomings in most Bayesian segmentation algorithms is the need for the number of classes or parameters $K$ to be entered. Furthermore, the algorithm needs the initial segmentation estimation. This can be extracted from a simple clustering algorithm such as k-means clustering [21]. The initial segmentation is refined in an iterative procedure.

To compute the Gibbs distribution of the a priori probability function defined in formula 2.5, the clique function should be defined. Pappas in [4] proposes a clique func-
tion coming from only two-pixel cliques as shown in Figure 2.4. The proposed clique potential function $V_c(X)$ associated with the pairs of pixels in an 8-neighbourhood system is defined as:

$$
V_c(x) = \begin{cases} 
-\beta, & \text{if } x(i, j) = x(k, l) \text{ and } (i, j), (k, l) \in C \\
+\beta, & \text{if } x(i, j) \neq x(k, l) \text{ and } (i, j), (k, l) \in C 
\end{cases}
$$

(2.7)

The positive parameter $\beta$ is entered into the algorithm. This clique function gives less energy to equal labels of adjacent pixels and more energy to unequal adjacent pixel labels. Therefore it encourages the adjacent pixels to have the same label. Increasing the value of $\beta$ increases the effectiveness of the clique function.

To derive the conditional distribution $P(Y|X)$, Pappas considers any region as a uniform or slowly varying grey-level. The effect of image degradation is modelled by an additive normal noise. Therefore the intensity of any region is considered as a normal distribution function with constant or slowly varying mean $\mu_{X(i,j)}$ and with variance $\sigma^2$. The mean and variance parameters should be estimated from the image regions. Therefore, if the statistical independence between different pixels is considered, the probability $P(Y|X)$ is estimated by the following equation:

$$
P(Y = y|X = x) = \prod_{(i,j)} \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left( -\frac{(Y(i,j) - \mu_{X(i,j)})^2}{2\sigma^2} \right) 
$$

(2.8)

Therefore considering equations 2.8, 2.6, 2.5 and 2.2 the posteriori probability function is computed as the following:

$$
P(X = x|Y = y) \propto \exp \left( -\sum_{(i,j)} \frac{(Y(i,j) - \mu_{X(i,j)})^2}{2\sigma^2} - \frac{1}{T} \sum_{all\ cliques} V_c(X) \right) 
$$

(2.9)

Therefore the cost function is equal to:
\[ Cost(X) = \sum_{(i,j)} \frac{(Y(i,j) - \mu_{X(i,j)})^2}{2\sigma^2} + \frac{1}{T} \sum_{all\ cliques} V_c(X) \quad (2.10) \]

The parameters \( \sigma \) and \( \mu \) which are the variance and the mean are calculated for each region or pixel depending on the optimisation algorithm. The parameters \( \beta, T \) and \( m \) the number of segmentation classes, are entered to the algorithm. The cost function has two components. The first part encourages the data in a region to be close to the mean, and the second term in the clique potential function which encourages the adjacent pixels to have the same segmentation classification. The final result is a compromise between these two values. The minimisation of the cost function depends on the optimisation method. Different approaches to the optimisation algorithm exist. Depending on the selected optimisation method, the cost function is simplified. In the next section some of the optimisation algorithms are briefly explained.

2.3.2.4 Numerical Approximations

Finding the MAP estimates of \( X_{MAP} \) can be viewed as a combinatorial optimisation problem. The large dimensionality of the unknown parameter \( X \) and the presence of a local minimum make it normally very difficult to find \( X_{opt} \). For instance, if \( Y \) is a 256 × 256 image with 8 different segmentation labels for each pixel, the set \( \Omega \) of all possible answers contains \((256 \times 256)^8 \propto 3.39 \times 10^{38}\) possible realisations, which makes it impossible to search all the possible results, so consequently it is necessary to use an approximation of the optimal solution. There are some numerical methods which will be classified into two groups as stochastic and deterministic solutions.

**Stochastic Solutions** This group of algorithms uses a controlled random search in the solution space. They analyse the present solution, and on a random basis move to another situation. Therefore these algorithms could sometimes move to a worse situation (decrease the probability) compared to the present solution. This is because there are many local minimums, and at times escaping from a local minimum requires going uphill instead of downhill towards a local minimum. Eventually, these algorithms find the global optimal solution but their computational complexity is very high. Therefore they are used for special applications where only the global result
is searched for. There are some versions of these algorithms, such as the simulated annealing, the metropolis algorithm, and the Gibbs sampler [6, 41, 66].

**Deterministic Algorithms** The problem with the stochastic algorithms is their computational complexity, which often makes their application impossible in practical situations. Deterministic algorithms are faster, but they are more likely to get trapped in a local minimum. Two famous deterministic algorithms are Iterated Conditional Modes (ICM) [67] and Highest Confidence First (HCF) [65] which are briefly explained in the following sections:

**Iterated Conditional Modes (ICM)** The idea for ICM comes from the Gibbs sampler algorithm [67]. The difference is that the new answer $X^{(n+1)}$ with $\Delta Cost > 0$ is not accepted. This means that only downhill is accepted, resulting in faster convergence, albeit, in a local minimum. In each step, ICM updates one pixel. It starts from an initial configuration, and the estimate is iteratively improved by visiting and updating the label $X(i, j)$ in a raster (or similar) scan order. At each pixel, $X(i, j)$ is updated by maximising the conditional probability, which is dependent on the pixel $(i, j)$ and its neighbours. Similar to equation 2.9, the conditional probability $P(X(i, j)|Y, X)$ is extracted by:

$$P(X(i, j)|O, X(K, l), \text{ all } (k, l) \in N(i, j) \& (k, l) \neq (i, j)) \propto \exp \left( -\frac{(O(i, j) - \mu_{X(i, j)})^2}{2\sigma^2} - \frac{1}{T} \sum_{C \in C_{i, j}} V_C(X) \right)$$

(2.11)

$X(i, j)$ is updated to maximise the probability. Therefore the cost function at $(i, j)$ is equal to:

$$Cost(i, j) = \frac{(O(i, j) - \mu_{X(i, j)})^2}{2\sigma^2} + \frac{1}{T} \sum_{C \in C_{i, j}} V_C(X)$$

(2.12)

$\mu_{X(i, j)}$ is the mean of region $X(i, j)$ at $(i, j)$. The parameters $T$ and $\beta$ in the clique function and $\sigma$ in the following cost function are interdependent. The ratio $\beta/T$ can
be replaced by a parameter. Therefore $T$ is replaced by $T = 1$. The effect of the two terms $\beta$ and $2\sigma^2$ in the cost function can be adjusted by changing the $\beta$ parameters. Therefore in a similar way to $T$, the $2\sigma^2$ parameter is replaced with one. Therefore the simplified cost function is:

$$Cost(i, j) = (O(i, j) - \mu_{X(i,j)})^2 + \sum_{C \in C_{i,j}} V_C(X)$$ (2.13)

When $X(i, j)$ is updated, it is necessary that the analysed neighbouring pixels be updated again. The updating procedure continues, until equilibrium is reached. One problem with the ICM algorithm is the order in which pixels are visited. The raster scan order that is commonly used has the undesirable property of propagating pixel labels in the direction of the scan order, because the algorithm encourages the adjacent pixels to have similar values. This problem is reduced in the Highest Confident First optimisation (HCF) algorithm. HCF algorithm uses the same cost function as ICM approach in equation 2.12, but the order of visiting the pixels is changed. Based on the maximum value of the $\Delta Cost$ function, the next pixel is selected and its segmentation is updated. The pixels are ordered in a queue based on the $\Delta Cost$ value. When a pixel is updated, its neighbouring pixels $\Delta Cost$ are also updated, therefore their order in the queue is also updated. HCF computational complexity is greater than with ICM, and it can also become trapped in a local minimum.

In summarising this section, it is worthwhile to reiterate that Bayesian segmentation has good performance and high flexibility making it suitable for many applications. However, it suffers from two weaknesses 1) the requirement for the $k$ parameter, indicating the number of labels, to be set by the user 2) the need for an initial segmentation estimate. There are a number of works aimed at addressing these weaknesses. For example, Meier et al. [69, 79] proposed a segmentation algorithm, which is a combination of Bayesian and morphology based approaches. It optimises the segmentation label based on HCF, and it does not need an initial segmentation estimation. Although Meier’s proposal addresses the above mentioned problems, its combination of Bayesian and morphological approaches rendering it impractical for multiresolution segmentation.
However, more research is needed to fully overcome the shortcomings of the Bayesian algorithms.

### 2.4 Multiresolution Image Segmentation

Traditionally, multiresolution image segmentation algorithms analyse the image data at different resolutions, which results in some advantages over single resolution segmentation such as:

- Less computational complexity
- Improvement of convergence rate
- Reduction in over-segmentation
- Less sensitivity to noise
- Ability to capture the image structures at different resolutions
- Less dependence on initial segmentation

Furthermore, multiresolution analysis and segmentation is needed to ensure the required spatial scalability of the extracted objects/regions for recent scalable object-based coding algorithms [11, 80].

Multiresolution segmentation algorithms consider the inter-scale image data correlations in the segmentation procedure. In the most straightforward case, these algorithms consider inter-scale correlation by projecting the lower resolution segmentation result to the next higher resolution as the initial segmentation estimation. The segmentation is further refined at the current higher resolution. This procedure continues progressively until the highest resolution is segmented. However, for spatial scalable coding the extracted objects/regions should be similar at different resolutions. In other words, the refinement of the projected higher resolution segmentation should affect and correct the lower resolution segmentation and vice versa. This
constraint maintains the similarity between different resolutions, and more reliable segmentation at different resolutions is obtained.

Although none of the algorithms in the literature have this feature, algorithms that consider the inter-scale correlations effectively have more potential to meet the similarity requirements. Therefore the multiresolution segmentation algorithms are analysed from this point of view and are thus classified into two groups. In one group the inter-scale resolution is not considered effectively, and after projecting the initial segmentation estimation from lower resolution, the current resolution is segmented by a single resolution segmentation algorithm [4, 81–83]. This group is described as “Hierarchical Multiresolution Image Segmentation” (HMIS).

In the second group of multiresolution algorithms, the inter-scale correlation is more effectively considered. They incorporate the statistical models, and the decisions at each pixel/block is based on the information from different resolutions [84–88]. However, often the causal inter-scale correlation with only the latest lower resolution [85, 87–89] or the next higher resolution is considered [86]. Considering the correlation with other resolutions results in a very complex model and increases the computational complexity. While algorithms in the first group segment grey or colour images and sometimes textured images, the algorithms in the second group are often used to process textured images. However, segmentation algorithms can often be modified to segment based on texture, grey/colour or other features of interest. These algorithms are described as “Highly Correlated Multiresolution Image Segmentation” (HCMIS).

There is another group of hierarchical segmentation algorithms which progressively segment the image at different scales. The original image is filtered by a Gaussian low-pass filter with parameter $\sigma$. Different values of $\sigma$ create the images corresponding to different scales. The images have the same size as the original image, and the hierarchical algorithm segments the sequence of images at different scales from low to high scale [55, 90]. In these algorithms, similar to the first group, the lower scale segmentation is projected as the initial segmentation to the higher scale. Therefore the literature regarding these algorithms is reviewed under the HMIS algorithms group.
As well as the above-mentioned differences, the multiresolution segmentation algorithms are different in many other details. For example, the selected segmentation approach can be very different, such as the cluster-based [91, 92], the morphological [63, 90], or the Bayesian method which is well suited for multiresolution image segmentation [84, 86, 93]. In the optimisation methods they either search for global results such as with stochastic approaches [86, 88, 94] or they search for local optimum results such as with deterministic algorithms [3, 95]. They could be unsupervised and estimate the parameters of the defined models [85, 94] or they could ask for the parameters to be entered [93, 96, 97]. They might be designed for a very specific segmentation application such as sonar images [98], or low depth field images [99], or they are proposed for general segmentation applications [81, 88, 100, 101]. Although each of these differences are active topics of research, the concern here is multiresolution image segmentation algorithms for object-based spatially scalable coding applications. The extracted objects/regions should be similar at different resolutions and the algorithms which consider the inter-scale correlation have more potential to present similar objects/regions at different resolutions. Therefore from this point of view, inter-scale correlations are important.

In the next two Sections, 2.4.1 and 2.4.2, the literature on multiresolution segmentation is reviewed. The review starts with the work on hierarchical multiresolution image segmentation.

### 2.4.1 Hierarchical Multiresolution Image Segmentation (HMIS)

Pappas in [4] presents one of the best MRF-based grey-level image segmentation algorithms which was described in Section 2.3.2.3. It has been used and further developed in some other works such as [19, 81, 102, 103]. In its hierarchical implementation, the lowest resolution segmentation is initialised by k-means clustering, and the adaptive clustering algorithm further improves the segmentation estimation. Subsequently, in an iterative procedure, the current segmentation is projected to the next higher resolution as a good starting point, and similarly the adaptive clustering algorithm further improves the segmentation estimation. This continues until the highest resolution segmentation is achieved. This approach reduces the computational com-
plexity significantly, since most of the iterations are at the lowest resolutions. The improvement in the performance is also significant. This algorithm does not consider inter-scale correlations effectively and the under segmentation is another problem of this and similar multiresolution segmentation algorithms.

To remove the under-segmentation problem of the multiresolution segmentation algorithms, edge information is used [81, 83, 104]. Edge pixels indicate the presence of different regions and improves the detection of small objects/regions. Tolias et al. [81] modified Pappas work to consider edge information. They extract edge information from the high pass sub-bands wavelet coefficients. The neighbouring pixels with different edge map values are assumed to belong to different regions. They added a term to the objective function of the MRF-based segmentation algorithm which encourages the neighbouring pixels with different edge map values to have different segmentation labels. This is different to the normal MRF segmentation that encourages any neighbouring pixels to have the same segmentation labels. A similar work is proposed by the Kopparapu et al. [104]. They used k-means clustering at the lowest resolution and at each resolution, the final segmentation is projected to the next higher resolution as initial estimation. At each pixel an edge processing refines the segmentation. They recognise a pixel as edge or non-edge pixel by calculating the high pass sub-band wavelet coefficients. The pixels at the opposite side of an edge pixel \((i, j)\) such as \((i - 1, j - 1)\) with \((i + 1, j + 1)\) are modified to the label of the closest cluster center, to have different labels. Their algorithm does not include the spatial continuity and the final result depends on the initial k-means clustering.

In [69], Miere et al. introduces a MRF-based algorithm, which include spatial continuity and edge processing, but only in single-resolution mode. The Canny edge detector is used in this algorithm. The edge information is processed only in the edge and its neighbouring pixels. The idea is that if there is an edge pixel between two non-edge pixels they are likely to belong to different regions. In [83], Munoz et al. propose a similar algorithm, in multiresolution mode. First, the most relevant edges are detected at the coarsest resolution. Then seeds are placed far from the edges and the region growing algorithms obtain the regions. Using a global similarity and gradient energy function with a greedy optimisation algorithm, all the
pixels are classified. Since in this algorithm regions move and shrink or expand to form final regions, they call their single resolution segmentation algorithm active region segmentation. The edges are dams for the region growing algorithm. Low resolution segmentation is iteratively projected to the next higher resolution, where non-boundary projected pixels model the cores of the regions. The greedy optimisation algorithm again obtains the regions at the highest resolution. The algorithm continues until the fine resolution is segmented. The results depends on the initial seeds locations. The algorithm cannot detect small objects/regions if they are not detected in the lowest resolutions. Considering the inter-scale correlation in the pixel classification procedure or extension to scalable mode is not easily possible.

To reduce the computational complexity, in some works, only regions around the projected borders are refined [83,93,105,106]. Gao et al. [93,105] propose a MRF-based multiresolution colour image segmentation algorithm which significantly reduces the computational complexity. They use a MRF expectation maximisation (EM) algorithm [107] which iteratively alternates between parameter estimation and segmentation optimisation. The low level estimation is projected to the next higher resolution and is refined, but in fine resolution, only a narrow band around the projected regions’ border is refined. Their algorithm assumes that the regions’ interiors at fine resolution are identical to their corresponding low resolution. The inter-resolution correlations are not considered effectively in this algorithm. If a region is not detected at lower resolutions, similarly, it is not detected at highest resolution.

One of the main applications of the segmentation algorithms is image/video coding. In the region-based second generation coding algorithms, one of the problems is the necessity of sending the segmentation map to the decoder side which allocates considerable part of the channel bandwidth. Amonou et al. [106] propose an algorithm which integrates multiresolution image segmentation with object/region-based coding, and only needs to sent the segmentation map at the lowest resolution. The idea is that, at the decoder and encoder side, run the same segmentation algorithm on the low pass-band of the current resolution. At the decoder side, the low pass band is reconstructed from the lower resolution sub-bands. The lowest resolution is segmented by a mono-level morphological segmentation algorithm adapted from [108].
The segmentation algorithm projects the lower resolution to higher resolution, and the areas around border are refined. The performed segmentation at the decoder side is different from the traditional policy, where segmentation is performed just once at the encoder (server) side and is used several times by the decoders or application side without running the semi-automatic segmentation algorithm with high computational complexity. Moreover, depending on the compression rate, the decoded low pass image can have very poor segmentation results. The algorithm sends the full sized sub-bands information at different resolutions and it is not useful for object-based applications.

To remove the over-segmentation and noise effect in segmentation of some applications such as remote sensing images, Zheng et al. [109] propose a DWT-based multiresolution segmentation algorithm. The algorithm modifies DWT transform with RDWT to extract a noise-free pyramid. The algorithm filters the image $L$ times, which removes the noise in the low-pass band image and then down sampling is performed $L$ times. This transform removes noise better than the traditional wavelet decomposition at some computing cost. Then a multiresolution segmentation similar to Pappas’ work is performed. One of the disadvantages of this algorithm is under-segmentation. In this algorithm high resolution segmentation can be significantly different from low resolution segmentation, and high resolution segmentation refinement has no effect on low resolution segmentation.

In a group of algorithms, where pixel-wise accuracy is not necessary, the block-based segmentation is proposed, which significantly reduces the computational complexity [91, 99–101]. They divide the pyramid to rectangular blocks. Bongiovanni et al. in [91] propose a multiresolution clustering algorithm for bimodal images which their histograms include two main peaks. The algorithm decomposes the image, and the pyramid is divided into rectangular blocks where the parent/children relationship of blocks establishes a quad tree. The leaf nodes of the tree are determined to be bimodal or unimodal by an exhaustive search of their histogram. The bimodal blocks are divided into two populations. Each parent receives the statics of its four children, and ranging over all sent statics, the parent node is examined as whether to be split into two populations or not. Now in a top-down procedure any parent node sends
its mean to its four children. In a child node, the pixel grey-level is transformed to the mean value(s) it received from the ancestor. This is done by proximity. The algorithm works well if the image is really bimodal, which is an image including of 2 sub-populations which is not right in generic real images. Liu et al. in [100] propose an algorithm, which replaces the block partitioning with clusters obtained from a single resolution segmentation [110]. The nodes are examined and they might be split or merged with their neighbouring regions by a relaxation algorithm.

Wang et al. [99] propose a multiresolution segmentation algorithm for segmentation of images with low depth of field. In this sort of image, the “object-of-interest” is sharply focused as foreground and the background is out of focus. The algorithm separates the image into two clusters as foreground and background. The algorithm starts from the lowest resolution and divides the image into blocks of size $S \times S$. Each block is coarsely classified by testing the block features by the k-means clustering algorithm. The feature is the variance of the wavelet coefficients. Then the classification is refined at higher resolution until the highest resolution. The algorithm is useful for foreground/background separation of images with low depth of field as two cluster segmentation. A similar multiresolution foreground/background separation [111] is described in Section 2.5.1 as a semantic segmentation algorithm.

Roma et al. [101] propose a multiresolution decision algorithm which segments/classifies the blocks of image into three categories, including textured, edge or smooth regions. The image is divided into blocks and for each block, the analysis of the wavelet coefficients of 3-levels of sub-band/wavelet decomposition classifies the blocks. The large number of high frequency components, with absolute values larger than the standard deviation of their corresponding bands declares the region as textured, and a low number of small coefficients indicates smooth regions. The threshold for edges is a value between the textured and smooth regions. The proposed algorithm segments the block of the fine resolution image into one of the 3 classes and cannot be extended to general multiresolution segmentation applications. It is block-based and it includes block artifacts.

The algorithm proposed by Makrogiannis et al. [90] replaces the blocks with the watershed basins which removes the artifact. Considering the similarity for grouping
the regions, they extend the application of the algorithm to segment the real images. The algorithm is a multiresolution decision-processing segmentation algorithm which robustly groups and merges basins to reduce the over-segmentation. The multi-scale dissimilarity function is defined by combining the non-similarities at different resolutions, which takes into account the structure of clusters at different resolutions. A region merging/grouping algorithm is then defined by a region adjacency graph (RAG). The RAG is divided to some subtrees called forest by a region similarity criterion. Then the two forests with minimum distance are merged, and the forest distances are updated. The forest dissimilarity values are calculated by the maximum of partial dissimilarities between the two forest members. The merge process continues until convergence or the final number of forests is formed. The clustering algorithm and similarity function are multi-scale, but the final result is extracted only at the finest resolution. The results, such as the Lena image segmentation, are not semantically satisfactory. Some parameters such as the number of forests are entered, and finally, similar to all morphology-based algorithms, the spatial continuity is not included.

The multi-scale segmentation lets that different areas of the image be detected at different scales. Smooth areas can be detected at lower scales and active and textured regions are better detected at higher scale segmentation. Therefore, considering the multiscale processing better fits to the segmentation [112]. These algorithms do not have the advantages of the multiresolution segmentation algorithms. In [112], a multi-scale image segmentation is proposed by Bertolino and Montanvert. The original image is segmented by a regular segmentation algorithm and the standard deviation of each region is compared with a threshold value $\sigma_M$ to decide if region must be split or not. Therefore the image is split into several regions with standard deviation less than $\sigma_M$. By changing the scale space parameter, $\sigma_M$, many segmentations at different scales are obtained, which create a pyramid. Tuning of the scale parameters allows the user to extract the entities at the desired level of detail. It needs user intervention to extract the “object-of-interest” at the proper scale. The scale parameters can have continuous values, and the numbers of possible segmentation maps is infinite. A more effective region splitting method can be used which produces a lower number of regions.
MRF-based multiresolution segmentation algorithms are used for textured image segmentation [113–115]. Multiresolution processing improves the accuracy and reduces the computational complexity of the segmentation algorithm. Bouman et al. [113] present a multiresolution segmentation algorithm. At each resolution the segmentation algorithm is based on the MAP estimation derived from MRF Modelling. A causal non-homogeneous Gaussian autoregressive model is used which allows to extract the statical model for texture at each pixel. Minimisation is based on the steepest descent algorithm, which has lower computational complexity than the ICM optimisation algorithm and about 1% to 10% of simulated annealing algorithms. In [84] they update their algorithm to consider the inter-level correlations which is reviewed in the next section. Salari and Ling [114] propose a multiresolution segmentation based on the features classification. At each level, four operators are convolved with the image to obtain a set of texture features and the image is segmented by k-means clustering of the features. Lower resolution segmentation is used as the initial segmentation estimation at the next higher level, and the pyramid is segmented progressively. At higher resolution, the features of each pixel are calculated and they are classified to the closest cluster center. Coarse resolution carries information corresponding to the large structure, while the fine resolution contains the necessary details to refine the segmentation.

Debure et al. [115] propose a multiresolution texture segmentation algorithm for wavelet-based image coding applications. They segment the residual information on the difference between the original and the compressed image. A region’s texture is modelled by an autoregressive model, and the model coefficients are also used to reconstruct the original image from the compressed image. The segmentation starts at the lowest resolution and the initial segmentation is obtained using the k-means clustering algorithm on the texture coefficients. The ICM optimisation algorithm then minimises the energy, and the result is used for the initial conditions of the next finer resolution. The texture model parameter estimations are incorporated into the iterative segmentation process. In order to provide visually acceptable synthesis results, the AR model is set to a large number such as 24. The number of clusters is treated as a user given input. The algorithm is useful for texture-based segmentation but its extension to scalable segmentation is very complex.
2.4.2 Highly Correlated Multiresolution Image Segmentation (HCMIS)

The algorithms of this approach consider the inter-scale correlations in their energy function. Therefore the results at different resolutions are more similar, and the algorithms have more potential to be fitted with the scalability feature, required for the applications such as scalable object-based coding algorithms. The outstanding works that use this approach are reviewed in the following.

Bouman et al. [84] consider the correlation with the last resolution, Kato et al. [86, 116] and Comer et al. [85] extend the correlation to the last and the next resolutions. Saeed et al. [96] consider inter-scale correlations with the last and the two last resolutions. More details of the works are reviewed at the following.

Bouman et al., after their first work [113], which was explained in the last section, propose a multi-scale approach to Bayesian image segmentation [84]. They replaced MRF with MSRF variables or Markov chain levels which are are composed of a series of random fields processing from coarsest to finest resolutions. The associated interaction structure is a quad tree which correlates the current pixel with the parent at the coarser resolution. Therefore, spatial correlations of the pixels have not been considered. At each resolution, segmentation depends only on the last level segmentation. At each level the MRF parameters and the probability density function values are estimated based on the expectation maximisation (EM). Since the spatial correlation is not considered the segmentation algorithm is not iterative and can be computed in a time proportional to MN when M is the number of classes and N is the number of pixels. As the authors claim, in some experiments its computational complexity is less than with the ICM approach, and its performance is comparable to simulated annealing optimisation. Deleting spatial correlation and the optimisation iteration could results in error propagation. If pixel segmentation is wrongly classified, the error will propagate through many descendant pixels until the highest resolution, because a pixel at low resolution affects 4 pixels at the higher resolution.

Kato et al. [86, 116] define a new multi-scale MRF model. First, a local interaction between two neighbouring pixels is defined. In addition the interactions between the
pixels on the lower resolution (the parent) and the next higher resolution (children) are also considered. Based on this new neighbourhood system, the energy function of the MRF variable is defined. This neighbourhood model and its corresponding energy function allow the more efficient propagation of the local interactions, resulting in estimates closer to the global optimum for the optimisation method. However, it also makes the model more complex and increases the computational complexity. The optimisation is implemented using a parallel simulated annealing algorithm. The optimisation can be run in parallel on the entire pyramid. The interaction between pixels is limited to between neighbouring resolutions. In their further works, they added MRF parameter estimation [94, 117]. The algorithm iteratively alternates between parameter estimation and segmentation. They aim for an unsupervised algorithm. However, the number of classes needs be entered. It is stated that considering the correlation between other resolutions results in a very complex model, increasing the computational time considerably [94].

Comer et al. [85] propose a multiresolution segmentation which includes inter-scale correlation. The proposed algorithm fits an auto regressive model to the pyramid representation of the textured image. The correlation between different resolutions of the pyramid is incorporated in the objective function of a multi-scale MRF model. The MAP optimisation criterion is replaced with the multiresolution maximisation of the a posteriori marginal (MMPM) estimation which facilitates the use of the EM algorithm to estimate the parameters such as the autoregressive model coefficients. The coarsest resolution is segmented in a single resolution mode and the segmentation is propagated down to the other levels of the pyramid. In this approach only the correlation between adjacent resolutions is considered. The error in low resolution segmentation will propagate to higher resolutions. Although some parameters are estimated, but many such as the spatial interaction coefficient and the number of segmentation classes are entered manually.

Saeed et al. [96] propose a multiresolution clustering algorithm derived from the EM estimation for both the model’s parameters estimation and segmentation optimisation. The image at each resolution is modelled as a mixture of Gaussian variables. To consider the spatial correlation of adjacent pixels, the log likelihood equation of
the Gaussian mixture model is penalized with a term $V(Z)$ which incorporates the intra and inter-scales correlations. The inter-scale correlation has been extended to both the last and the second last (grandfather) lower resolutions. In an iterative procedure from the lowest resolution, the mixture model parameters are estimated, and then a new classification is obtained until convergence. The causality of correlation between resolutions is considered and extension to more than three resolutions would be very complex. Therefore it is not useful for a scalable segmentation algorithm.

Wilson and Li [87, 88, 118, 119] propose a block-based grey/texture image segmentation. The algorithm includes the spatial and causal correlations with the last resolutions and also reduces the computational complexity. However, to reduce the block-artifacts, the algorithm is followed by a line processing which refines the borders. They present a multiresolution segmentation which doesn’t require the number of classes as an input parameter, which is necessary for most MRF-based segmentation algorithms [88]. It updates the statistical models proposed by Bouman et al. [84] with the view that at each scale or resolution data is conditioned not only by its immediate predecessor (parent), but also directly dependent on its neighbours at its own scale. At each resolution, the image is divided into blocks, and every block is classified by MRF-based segmentation. The initial segmentation comes from the lower resolution segmentation and it is optimised by a simulated annealing technique. After the highest resolution optimisation, line processing refines the regions’ borders to the actual borders. Later in [87] they upgrade their algorithm by a region merging algorithm. After every $i$ iterations, two neighbouring regions are merged if the merging criterion is satisfied. This region merging removes over-segmentation and helps the optimisation algorithm to escape from the local optimum trap. Detecting new regions at higher resolutions, especially small regions is not possible. It needs a region boundary refinement which increases the computational complexity, and also, its procedure does not interact with region labelling. The number of classes at the lowest resolution is a random number that should be greater than the expected number of classes, which again is a limitation for an automatic segmentation procedure. In 2003 they introduced a genetic approach multiresolution segmentation [118, 119] which similarly classifies the blocks. This algorithm has the computational complexity of genetic algorithms and also needs boundary refinements. Furthermore,
detecting small regions is a major challenge. Advantages of MRF segmentation such as noise sensitivity and spatial connectivity are lost.

Some segmentation algorithms are proposed for particular applications. Cheng et al. [120, 121] propose an algorithm which extracts the background, documents and figures. Mignitte et al. [98] propose an algorithm to segment sonar images. Neelamani et al. [122] propose a segmentation algorithm for region-based coding applications.

Cheng et al. [120, 121] present a trainable multi-scale Bayesian segmentation which can model the context of images in a limited class of images with a combination of documents, background and figures. It uses a binary classification tree to model the transition probabilities between pixels at adjacent resolutions [123]. The image is decomposed by a wavelet decomposition, and each pixel segmentation label is supposed to be dependent on its $5 \times 5$ neighbourhood of pixels at the coarsest resolution. The transition probabilities are estimated from some ground truth segmentation examples, which leads to training the essential aspects of the contextual behaviour model. Finally, by using sequential markov random field variables and transition probabilities, the pyramids pixels are classified from the lowest to the finest resolution. The algorithm is limited to document classes of applications. Low resolution classification errors are spread to high resolution classifications. Each pixel transition probability depends on a large neighbourhood of $5 \times 5$ pixels at coarser resolution, which renders the algorithm useless for the scalable segmentation algorithms. The algorithm suffers from spread of low resolutions to high resolutions.

Mignitte et al. [98] propose an unsupervised hierarchical MRF model to segment sonar images. The algorithm has two phases: at the first step parameters are estimated, and the second step is devoted to the hierarchical segmentation. Parameters of the data model are estimated in an iterative manner called iterative conditional estimation (ICE). It combines a maximum likelihood approach for noise model parameters estimation with a least squares method. ICE offers flexibility, which allows an efficient adaptation to the MRF model. The initialisation of the iterative parameter estimation is provided by a simple clustering technique based on the luminance distribution in a small window. In the second part hierarchical segmentation is performed.
It is based on a multi-level prior model involving scale-space causal interactions between adjacent consecutive levels (Parent-child interaction) and spatial interactions between sites on a level. The algorithm is designed for sonar images, which are limited to three classes of objects. The scale space relation is also limited to the last scale, and the computational complexity is another problem of the algorithm.

Neelamani et al. [122] propose a multiresolution image segmentation algorithm which simultaneously extracts the coding coefficients. It is argued that the MAP based segmentation is equal to the Minimum Description Length (MDL) segmentation approach. A multiresolution MDL based segmentation is proposed which minimises the number of bits in a Zero tree Significance Map (ZSM) coding algorithm. The statistical texture model is first characterized by the hidden Markov statistical model, proposed by Crouse et al. [124]. The initial segmentation estimation is organised in a quad tree, and based on the MDL criterion, a dynamic programming algorithm optimises the segmentation estimation by minimising the code length of the ZSM coding algorithm. The algorithm models inter-resolution correlations. They present a solution for the case of textured images with only two classes. Extending the algorithm to the general case, will increase the computational complexity and render its application to real images very limited.

2.5 Semantic Image Segmentation

While the low level segmentation partitions the image into different homogeneous regions, the final goal of the segmentation processing is to divide the image into the meaningful objects/regions such as human, car, sky, sea, etc. This is important for the next stage of processing such as object and pattern recognition, computer vision, content-based retrieval and coding, etc. In this section the outstanding works in the literature related to semantic image segmentation are discussed and reviewed.

Although there are many works about pattern and object extraction and recognition, very few consider a real segmentation stage. Some of the works on object recognition assume that the objects’ shapes are already extracted [125–130], while others use a simple segmentation algorithm by considering the object(s) in a very simple scene
rather than a real image [131–134]. These algorithms are more about recognising and classifying the detected shapes. However a comprehensive solution includes both (1) segmentation (2) recognition or classification of the extracted object, and should be applicable to real images.

The works in the semantic segmentation literature can be separated into two different categories. In the first group, by using some high level knowledge, the “object-of-interest” is detected, while in the second group the entire image is segmented into meaningful regions. These algorithms are useful for scene interpretation and understanding but suffer from many limitations which reduces their affectivity.

2.5.1 “Object-of-Interest” Extraction

In the majority of algorithms for “object-of-interest” detection, some high level knowledge about the objects’ characteristics, such as the object model and qualitative and quantitative relationships are employed. In most of the algorithms of this group, the “object-of-interest” model is searched for in the image by considering the low level information in the segmented image. These extraction algorithms can be designed for specific applications, such as car or human extraction [135, 136]. An extension to these algorithms is searching the scene for objects which also exist in a library of templates. This is aimed toward full implementation of HVS perception. Some simplifying assumptions about the scene will reduce the computational complexity.

A basic concept related to objects is their shape which is determined by the objects contour. Most parts of the objects’ contours are extracted by the edge extraction algorithms with low computational complexity. Therefore many “object-of-interest” extraction algorithms are edge-based. The most important points in the shapes are high curvature points. Therefore one group of edge-based algorithms extract the high curvature pixels to find the “object-of-interest” [137,138]. These works are reviewed at the following.

Stein et al. [137] proposed an edge-based object extraction based on the super-segments. A set of consecutive line segments make a super segment which is coded
by the curvature angle of the super-segment angles and eccentricity. The codes are recorded in a hash table. The coded super-segments which are related to the same model make a group. Then the transformation from the model coordinates (in the database of models) to the scene coordinates is computed by applying a least square which matches the model to the super-segments of the group. One of the major problems of this algorithm is that in a cluttered background or with a complex object, the number of super-segments increases rapidly. In addition, the matches of some super-segments of an occluded object with the corresponding model can result in wrong detection of the object in the scene.

In 1997 Bennamoun et al. [138] proposed an edge-based approach to model-based segmentation. First the edges are extracted. Then in a shape decomposition stage, different parts of the object are isolated based on the dominant pixels with high curvatures as key points. These parts are then modelled with 2-D superquadrics such as squares and circles. Each superquadrics specify an object’s regions. The parameters corresponding to each regions such as position, orientation, size, shapes and spatial relationship are compared with the objects information stored in the database so that the object can be identified accordingly. The algorithm is scale, orientation and translation invariant. The algorithm has been tested with images composed of objects over a simple background. The closed edges of the objects have been used to separate object from background. However, in real images with a cluttered background, some parts of the edges are often not detected. Moreover, in textured images where many edge pixels are detected, the computational complexity of the algorithm increases significantly. The algorithm is more fitted to the task of classifying a detected object to one category of known objects defined in a library, than a semantic segmentation algorithm.

Curvature-based object extraction are very sensitive to the accuracy of the detected points. In noisy or real images this can adversely effect the object detection. This problem is reduced in the approaches which process all pixels of the object’s contour [139, 140]. Li et al. [139] propose a multiresolution approach for object shape description and recognition. First, a morphological filter removes the noise. Subsequently a change detector is used to identify the edge pixels followed by a linking
algorithm to extract the closed contour. A normalisation process is performed to produce the normalised edge coordinates to represent 2D shift, scale and rotation invariant features of the system. For normalisation, the edge pixels are transformed to polar coordinates, the pixels’ radius are normalised to the distance $[0, 1]$, and the shape is rotated so the first moment of the curve is minimised. In the next step, the normalised polar coordinates are transformed to the wavelet representation at different scales as the representation of the shape at different scales. Multi-scale features are then matched in a hierarchical way. Matching starts with the coarsest scale and moves up to finer scales. The program is terminated when the target is completely identified or completely rejected. Although the presented multiresolution approach reduces the computational complexity, the algorithm is not sufficiently involved in the segmentation stage. It uses edge links algorithms which are very complex in a real scene and can result in the wrong detection of the edge pixels. Furthermore, the system does not explain how to deal with the cluttered background and the complexity created when different objects exist in a scene. In other words, the algorithm uses some simplifying assumptions at the segmentation stage which are far from a real scenario.

A geometrical invariant object recognition algorithm for image retrieval and recognition applications was proposed by Alfrereze et al. [140]. For the recognition application, first the templates are stored in an image database, then the input image is segmented. They introduce a new contour parametrization for the object’s contour which is affine invariant. Based on the transferred contour coefficients, a function called the shape’s signature is defined. Affine invariant signatures of the object models are also stored in the database. A resolution scale selection is performed by computing the energy of the models’ signatures at different scales. The scale where the energy appears to be concentrated is selected for recognition, because large values of energy imply more information. For each observed image the affine invariant signature is compared with the signatures of all models in the database. Correlation coefficients are used to determine the similarity between each pair of signatures. In [141], they have extended their work to the illumination invariant features, but again they have the similar assumption that the object contour has been properly extracted or that the object can be easily distinguished from the background. In this work, multiple
images or a library of “objects-of-interest” are organised in a database of templates. However, it is acknowledged that the segmentation stage is very challenging and a perfect segmentation is like the Holy Grail. Therefore, it is assumed that there is a simple image background and if it is necessary, user intervention refines the segmentation result.

The edge-based work of Jurice and Schmid [142] replaces the high curvatures points with the most salient convex edges to detect the object area. In addition, it does not need to the closed boundary. However, the algorithm has not pixel-wise accuracy. The algorithm is reviewed at the following. At first, the Canny algorithm extracts the image edges. Local convexity is measured by the extent to which the detected contours support circle or arc-line structure at each position and scale in the image. Convexity support is measured by combining two terms based on the edges near the circle, by a classical edge-energy term that encourages strong tangentially aligned edge-energy and a novel entropy term which ensures that supports come from a broad range of angular positions around the circle, not from a few isolated positions with unusually strong edges. A search across position and scale finds local maxima of saliency (convexities) over position and scale. The found edges with high convexity determine a circle area which belong to the object. All the found circles, determine the object area. For the object category detection algorithm, at first the algorithm is trained by the feature extraction of some examples (database) and then it is compared with the extracted features of the object area. Using an ellipse instead of circle extends the algorithm to affine invariant shape comparison. The search for salient edges in scale space has large computational complexity and in a cluttered and textured background wrong edges can be detected. While the algorithm doesn’t need closed contours, the extracted regions are not matched with the real object’s border, which is necessary for some applications such as video editing.

Edge-based algorithm have some fundamental problems. They are very sensitive to noise. In textured or cluttered background images, many edge pixels are produced which makes the algorithm very complex. In addition, the threshold for edge detection which determines the level of the detected pixels is a key parameter that should be entered manually. Region-based algorithms do not have these problems.
Because they are less sensitive to noise and parameter values. In addition region-based algorithms can effectively segment the textured images. Therefore, these algorithms are more suitable for automatic procedures in different segmentation applications. Recently, some region-based “object-of-interest” extraction has been proposed [143, 144].

An algorithm for the object search using the image partition information is proposed by Marques et al. [143]. The image is segmented into different regions using a region growing and merging algorithm. The region’s borders are used to search for the “object-of-interest”. The curve of the template of the “object-of-interest” is transformed into the affine parameters and is searched for in the regions’ borders leading to the best distance information. The transform parameters are changed until the best match in the image is found and the object is extracted. The transform parameter distance is sampled and quantized so that the number of all possible solutions is reduced. The high computational complexity of this algorithm is its main shortcoming. How the range of the parameter space should be estimated and the search method are not fully explained. The exhaustive search in the parameter space significantly increases the computational complexity.

Xu et al. [144–146], propose an object segmentation algorithm for content-based image database search applications. At first the image is segmented into homogeneous regions by the MRF-based colour image segmentation algorithm. All possible combinations of various regions are compared with the template in an affine invariant matching. The Hausdorff distance between the template and the transformed object’s shape is defined as the shape’s distance. A group of regions with distance less than a threshold represent a possible “object-of-interest”. Due to the great number of possibilities for combining the regions, the search is computationally very complex. To lighten the computational burden, Xu et al. [144–146] propose multiple segmentation maps, organised in a stack, be searched. At the bottom of the stack is a single resolution segmentation map, and at each level of the stack, the two most similar regions are mixed to produce the coarser/lower resolution segmentation map at the higher level of the stack. At the top of the stack there is an image segmentation with only 2 regions. The template of the “object-of-interest” is searched for starting
from the coarsest level at the top of the stack towards the finest map at the bottom of stack. The main problem with this approach is the number of segmentation maps in the stack. There is too many segmentation maps in the stack proportional to the number of regions in the single resolution segmentation map at the bottom of the stack. Two consecutive segmentation maps on the stack are very similar, which increases the number of segmentation maps and computational complexity for search through the segmentation maps. The region selection policy for region deletion, which is the highest similarity criterion, can be replaced by a better selection criterion with better performance as is proposed later in Chapter 5.

Some works are proposed for special applications such as car detection on the road [135], face detection [147], human detection [15, 136]. These algorithms use simplified assumptions to reduce the computational complexity. In the following these works are reviewed.

Tan et al. [135] proposes an edge-based image segmentation and object recognition for car recognition in a road traffic scene application. The gradient direction at each pixel is tested and the image line segments are extracted. It is assumed that the structure of a road vehicle includes two sets of parallel lines (related to car’s roof), one along the length and one along the width direction. By testing the parallel lines, the orientation of the car compared to the ground plane reference is found. The car model is searched using the known direction. To search the car location, the 3-D model is projected to the image plane according to the direction found and it is matched with the image. The similarity is measured by the cross correlation, and the peak of the correlation determines the image plane projection of the model. The proposed algorithm is designed for real time applications. It uses too many assumptions related to the car traffic application to simplify the algorithm. In a cluttered background too many edge pixels are detected, which increases the complexity of the algorithm. These problems limit its application for more general purposes.

Many works on face detection can be found in the literature [15, 147, 148]. The overview of all these works is beyond this thesis scope. The algorithm proposed by Zehang et al. [147], is reviewed as a typical work in this area. They propose a scalable multiresolution face detection algorithm. The image is decomposed into multireso-
olution using a wavelet pyramid. Human skin regions are detected at lower resolutions by a statistical Bayesian based decision algorithm. The high frequency wavelet coefficients corresponding to face details such as eye, brow, nose, supports separation of the face region from the other skin regions. The existing parent-child relationships between regions at different resolutions help to detect the object at the other finer or coarser resolutions. The proposed segmentation algorithm offers the shape at different resolutions, which is useful for progressive image coding algorithms where the bitstream is embedded (spatially scalable coding). The multiresolution analysis decreases the computational complexity effectively but the algorithm cannot be easily extended to detect the other kind of “objects-of-interest”.

Fan et al. [15, 136, 149] present an automatic image segmentation algorithm for semantic human object extraction. First the colour edge pixels are extracted and then the region’s seeds are placed automatically using edge information. The image is then segmented into homogeneous colour regions by a seeded region growing algorithm. The segmentation is further refined considering the regions’ border and edge information [150]. Subsequently the human skin regions are detected. The face of the human object is then extracted from among the skin regions by some geometrical constraints. The face region is the semantic seed for the human object. A perceptual model of the object’s adjacency relation and size is then matched with the image sub-regions. If the model and the regions are well fitted, the adjacent regions that correspond to the object’s region are merged to produce the semantic object. It is hoped that the algorithm can be extended to include other objects such as cars and airplanes. This, however, might prove to be a formidable task because, while human face detection is a good key point to find a semantic seed, generally it is not easy to find a suitable seed for other objects.

2.5.2 Scene Segmentation and Interpretation

In the second group of algorithms, using the low level features of the image such as colour, texture, edges, etc., the image regions are extracted, refined and combined to establish their correspondence to higher level image descriptions. For example, regions belonging to the same class such as grass or sky are mixed together. These
algorithms are more useful than the “object-of-interest” extraction applications for scene analysis and interpretation. However, many of these algorithms don’t often guarantee that the final regions are all semantic regions representing meaningful objects or regions. They often cannot extract a complex object which includes different regions such as human, car, etc., from a real image. Therefore many of these algorithms are related to natural image segmentation [151, 152]. For example, in many works the detected meaningful regions are rigid and simple, such as sky, water, etc., which form a homogeneous region. An example of the application of these algorithms is remote sensing image analysis [153]. These algorithms often cannot overcome scene changes such as light variations which affect the low level features. Some outstanding works of this group of semantic segmentation algorithms are presented in this section.

Some of these segmentation algorithms are designed for special images, and by using specific assumptions simplify the process. Huang et al. propose a foreground/background separation for segmentation of the museum or catalog images [154]. Dos et al. [155] propose an image segmentation for the birds image with simple background area. Bolddys [152] suppose the image include only 11 kinds of known meaningful regions.

One of the first approaches to object extraction is the foreground background separation proposed by Huang et al. [154] in 1995. They suppose that the background is smooth but may have spatially varying colours or textures. At first, an adaptive edge estimation algorithm detects the edge pixels. These (closed edge) pixels, identify the foreground boundary. Because of the local nature of the gradient and its sensitivity to noise, the detected foreground boundary is noisy. To refine the noisy borders, the detected foreground is overlapped with an MDL-based spatial segmentation. Regions with more than 50% foreground pixels are detected as foreground. The algorithm is designed for specific applications such as object-based museum or retail catalog image database retrieval where the background is smooth and the object has sufficient contrast.

A foreground and background extraction algorithm for content-based image retrieval used to be for a database of bird images is proposed by Dos et al. [155]. It is assumed
that the “object-of-interest” (bird) is in the center of the image with dominant size and that the image background is less prominent. The background colours are determined by the blocks at the image margin. The background area is then eliminated, and the remaining regions with sizes greater than a threshold and away from the border are processed by an edge detector. Considering the well contrasted object and blurred background, background edges exist only at low scales, while object edges are present at all scales. Therefore, object edges are detected at a high scale, and long edges which belong to the object are kept while the short edges are deleted. Finally the object area is detected by the remaining edges. This algorithm is designed for a specific application, and cannot be used generally. The extracted object’s border is not very precise and if the foreground regions or colours exist in the peripheral areas, the background colour cannot be detected and deleted correctly. If the background is not out of focus or if the object is not the dominant object it may not be possible to discriminate between the foreground and background. The proposed algorithm cannot discriminate between different objects in the image.

Bolddys [152] in 2003 proposed an algorithm for segmentation of 11 semantic objects in natural images such as sky, grass, and sand which are the combination of a few homogeneous regions. At first, the algorithm is trained with about 500 images which are manually segmented. The features of each extracted object including colour and texture are extracted. A very conservative and fast initial segmentation is then performed. The region merging is performed based on regions’ features. The output of the algorithm is not necessarily the semantic object, and often oversegmentation occurs. Therefore it is useful for the next step of processing to includes higher level processing. It is not applicable for other semantic objects such as a car or human, which are not homogeneous, and their features cannot be extracted.

The other algorithms which are presented for generic applications include some different assumptions that simplify the algorithm and practically limits its applications. Newsman in [151] supposes that each semantic regions is a homogeneous region. Paro [156] and Hirata et al. [157] merge the segmented regions using different criteria to obtain the semantic objects. Their algorithm is successful for simple images and in many cases needs user intervention to correct the result. The algorithm pro-
posed by Lefevre et al. [111] segment the image into foreground and background area. It is successful in images where the background is the dominant area.

Newsam [151] proposes a content-based image representation for image database, retrieval and analysis applications. The image is segmented by a texture segmentation algorithm. It is supposed that the semantic objects, such as lake, highway, etc., have a homogeneous region. An object descriptor based on shape, colour and texture is created to facilitate search and retrieval from an image database. It is clear that this algorithm has many limitations and cannot be extended to general applications, because normally semantic objects such as a car, human, etc., include different homogeneous regions.

Paro [156] proposes an algorithm for semantic image segmentation. The algorithm extracts all the edges of morphological regions called the level set. The edges are filtered and deleted according to filtering criteria based on T-junctions, compactness and contrast. T-junctions appears at the borders of two objects that are occluding each other. Compactness is the perimeter to area ratio, which penalizes complex shapes. Contrast is another important feature to define perspective objects. Finally, the filtered edges detect the object’s region in the image. The extracted regions are not necessarily semantic, and there is over-segmentation. Therefore it is used as initial segmentation for Paro’s further work [158]. This introduces a semantic object extraction algorithm based on a perceptual metric of the regions. After the first initial segmentation obtained by morphological tools, a region merging algorithms with a statistical similarity measure criterion reduces the number of regions. The region merging process continues through considering the perceptual information for merging criteria. Perceptual information includes low level features such as a region’s contrast, size, shape, and high level features such as foreground/background and location. To determine if a region is part of the background, the number of edge pixels that belong to the region is counted. For the location feature, it is assumed that viewers focus on the center of the image. Therefore, the number of pixels of the region which are within 25% of the center of the image is the location feature of the region. The merging of regions continue until the number of regions will be less than a threshold or until a single region is found. These algorithms detect the object
in simple images such as the first frame of the Clair sequence, however in a more complex or real images with cluttered backgrounds, it needs a user’s intervention, which makes the algorithm semi-automatic.

An algorithm for semantic image segmentation suitable for image retrieval applications is presented by Hirata et al. [157]. Their algorithm integrates contour-based analysis with region-based analysis to extract boundaries and delete other edges such as texture’s edge. At first the image is segmented into some homogeneous regions, and then by a boundary complexity analysis some of the adjacent regions with similar colours and complex boundaries are merged. This will continue, and more regions are merged by different similarity criteria in an iterative region merging procedure. At this stage the location, colour and texture distances are considered. Finally the small regions surrounded by a region are deleted. The system successfully extracts simple objects for image retrieval applications with some kind of user intervention, but if the objects are more complex such as real objects only some regions are extracted. Therefore the algorithm cannot be used as a general and automatic object segmentation algorithm.

A block-based multiresolution colour image segmentation algorithm for foreground/background extraction from an outdoor image is proposed by Lefevre et al. [111]. They assume that the background area has a uniform colour feature. A pyramid of decomposed images at different resolutions is created. The background is the largest region at the lowest resolution. The foreground/background segmentation is propagated and refined iteratively from lower resolutions toward higher resolutions. The image at higher resolution is divided into different rectangular regions. The feature of each region is compared with the same feature value of the background area at lower resolution to be classified as background or foreground. The H component of HSV colour space is the compared feature for comparison because the hue colour components are robust to illumination change and also to the successive averaging and filtering phase processed in pyramid creation. The proposed algorithm has low computational complexity and is useful for real time applications, but the assumption about the large and uniform background area limits its applications. The separated foreground areas are unions of rectangular regions and do not have pixel-
wise accuracy which is important in many object-based applications such as video editing and manipulation.

2.6 Video Segmentation

Video segmentation has been studied for more than thirty years. The first generation of video coding algorithms divided the sequence of images into rectangular regions for block-based coding applications [159]. Towards achieving a higher compression ratio and removing the blocky artifact effect of block-based video coding algorithms, the second generation coding algorithms partitioned each image frame into several homogeneous regions on the basis of low level features such as grey, colour, texture, motion, and following motion compensation the segmented regions were coded [55, 106, 160]. However due to the shift of the signal processing focus toward content-based processing and the popularity of object-based multimedia applications, the concept of semantic video segmentation has introduced a new challenge for segmentation algorithms. In particular, MPEG-4 which has emerged as the image/video coding standard for multimedia coding and communications, has increased the motivation of researchers to develop an effective object-based video segmentation algorithm [5, 161–165]. Ideally, the aim of segmentation should be partitioning of the scene into meaningful objects/regions. However due to the huge amount of data in digital video clips, the aim is simplified to extraction of “object(s)-of-interest” and moving objects from the scene. The MPEG-4 standard defines the extracted objects as the video object plane (VOP). Due to the sensitivity of the HVS to borders, the extracted objects should have pixel-wise accuracy. Minimising the user intervention and reducing the computational complexity, especially for real time applications, are the other challenges in video segmentation algorithms.

2.7 Motion

Video has the concept of motion which is a very useful feature for discrimination of the moving objects in the scene. In the computation phase, motion is considered as a low level feature, while motion also contains high-level information such as object
motion and object membership. The fundamental assumption for motion estimation is that the luminance/colour of a pixel $P$ on moving objects remains constant along $P$’s motion trajectory $[3, 6, 166]$:

$$I(x(t), y(t), t) = C$$  \hspace{1cm} (2.14)

which is called the “optical flow constraint” (OFC). The motion of regions/pixels is determined by identifying the position of the corresponding regions/pixels in successive frames. Therefore the corresponding region/pixel is determined by searching and minimising a criterion such as least square error. In the non-parametric motion field, a motion vector is assigned to each pixel. In the parametric motion field estimation, the motion of each region is described by a model, making it very compact in contrast to the non-parametric dense field description. Different models exist in the literature $[3, 6, 166]$. The most famous and frequently used model is the affine motion model which describes the displacement from frame $k$ to $k + 1$ of a pixel of a region, by translation, rotation and linear scaling given by the following equations:

$$x' = a_1 x + a_2 y + a_3$$
$$y' = a_4 x + a_5 y + a_6$$ \hspace{1cm} (2.15)

where pixel $(x, y)$ in frame $k$ corresponds with $(x', y')$ in the frame $k + 1$. The parameters $a_1, \ldots, a_6$ describe the model. They should be estimated for any object/region by an algorithm such as least square, regression, iterative estimation $[166]$. One advantage of the parametric model is less sensitivity to noise, because many pixels contribute to the parameter estimations.

2.7.1 Motion Estimation

There are different methods for motion estimation which can be classified into two main groups: (1) block-based matching and (2) recursive methods. Both estimation methods rely on the OFC assumption that luminance of pixels is not changed on the motion trajectory.

For dense motion field estimation, block matching, due to its simplicity, is the most
popular approach. The current frame is divided into blocks of size $n \times n$ and due to the small size of the blocks, the pixels of any blocks are assumed to undergo the same motion vector. The motion vector is found by the best match in the next frame (forward motion) or the last (backward motion) frame. The least squares error or mean absolute difference are the typical criteria for matching. A large block size can contain more than one motion direction, therefore it cannot determine the borders accurately. The small size blocks are at more risk of incorrect matching. A full search for the best match requires a lot of computational complexity, and it is limited to the maximum displacement estimation. Multiresolution and hierarchical search algorithms decrease the computational complexity and increase the accuracy of the search [166, 167].

There are several recursive estimation algorithms such as pixel-wise gradient-based and Bayesian-based algorithms [6, 166, 167]. They both recursively optimise an objective function to find the motion estimation. Details can be found in the references.

There are different methods for parameter estimations of model based motion. The most popular one is first estimating the dense motion field and then by regression or the least squares method fitting the model to the dense motion fields [6, 166, 168]. There are approaches which support the direct estimations of the model parameters [169, 170].

### 2.7.2 Apparent Motion

Motion in the video signal is the projection of the three-dimensional motion onto the image plane. The only available observation is the time varying intensity (colour) $I(x, t)$. Therefore the apparent estimated 2-D motion vector has less information than the real motion vector, and sometimes they are different. For example, consider a static scene with time varying intensity. The real motion is zero but the apparent estimated motion vector is not zero [3]. The two inherent problems with the optical flow assumption and apparent motion estimation are (1) the aperture problem and (2) the occlusion problem. These problems are related to the fact that although the projected motion is considered a low level feature, while it contains high-level information such as object motion and object membership [2].
2.7.2.1 Aperture Problem

Motion estimation needs enough contextual information for finding and matching the corresponding region in the consecutive frames. The lack of sufficient texture causes ambiguities in determining the corresponding region in the following frame. For example, a circle of uniform luminance rotating about its center, does not produce any apparent motion vector [3]. In the other example [2] consider two identical grey squares that move vertically. In the next frame, there are two grey squares that have been displaced up. However, the optical flow constraint allows the possibility shown by Figure 2.5. In other words, the aperture problem points to the ambiguity in determining of the corresponding region in the consecutive frames.

2.7.2.2 The Occlusion Problem

A moving object naturally creates covered and uncovered background in the image as is shown in Figure 2.6 [3]. The optical flow constraint for these two background regions determines a non-zero motion vector which will inevitably result in misclassification as foreground. Therefore the apparent motion vectors of these regions are not valid, and these regions should be dealt with and deleted from the foreground with some post processing. For example, a large difference between the motion vector of the uncovered region and the back-projected motion vector of the corresponding region determines uncovered background regions [165].

**Figure 2.5** Aperture problem; the movement of two identical blocks upward has two explanations: (a) both of the blocks moved up in the consecutive frame; (b) or diagonally switched places. Figure from [2].
2.8 Different Approaches in Video Segmentation

There are many different algorithms for the video segmentation task [5, 163, 164, 171–173]. Different classifications of these algorithms are in the literature [3, 6, 174]. Meier [3] classified motion segmentation into four categories as 3-D motion segmentation, motion-based segmentation, joint motion estimation and segmentation, and spatio-temporal segmentation. However it can be reduced to two groups, the motion-based and spatio-temporal segmentation approaches [174]. Tekalp’s classification is based on motion estimation by direct methods (change detection), optical flow-based segmentation and simultaneous motion and segmentation estimation [6]. Zhang classifies the algorithms into two groups as motion-based versus spatio-temporal, and each approach includes several sub approaches [174]. All the aforementioned classification methods are acceptable based on a specific point of the view. In this review, a new classification based on the combination of these classifications with some newly defined groups is considered. First, segmentation algorithms are divided into two main approaches, which are the region-based and the semantic segmentation algorithms. Region-based algorithms are important because they have been a basis towards the development of meaningful segmentation. Region-based segmentation
methods are divided into three approaches, 3D video segmentation, motion-based segmentation and spatio-temporal video segmentation. The semantic segmentation algorithms are divided into three categories; one group is based on the change detection algorithms, and the other one tracks the “object-of-interest” and the third one consists of hybrid algorithms that track the objects detected in the previous frames and also detect newly appearing objects. The classification can be seen as the following\(^2\):

1. Region-based video segmentation
   - 3D video segmentation
   - Motion based segmentation
   - Spatio-temporal segmentation

2. Semantic video segmentation
   - Change based segmentation
   - Video object tracking
   - Hybrid video segmentation

The classification considered to some extent shows the evolution of video segmentation algorithms. Because the core of the proposed semantically video segmentation algorithm is a tracking algorithm, tracking algorithms are discussed more fully while the other approaches are briefly explained.

### 2.9 Region-Based Video Segmentation

This approach divides each frame of the image sequence into different homogeneous regions in term of low level features such as intensity, colour, texture, motion. The

\(^2\)Although more groups than the other classifications is defined, due to too many adhoc video segmentation algorithms in the literature, classifying the algorithms into one of the groups is not trivial.
important application of these algorithms is second generation coding. The region-based coding algorithms remove the block artifact of the traditional block-based video coding algorithms. These algorithms can be classified into three groups, which are explained in the following sections.

**2.9.1 3D Motion Segmentation**

Initially, the video signal is considered as a 3-D signal, and the image segmentation algorithm is extended to the video domain [55, 103]. Although the role of time is not similar to that of the spatial information. The extracted 3-D regions are homogeneous, but in terms of motion the extracted regions are not perfect. These algorithms do not consider the motion information; therefore, the temporal continuity of the label field is not well achieved. A pixel is expected to have the same segmentation label as in the last frame, while in the moving objects it can be different. Due to the different role of the time axis and the importance of motion information, these algorithms have not been extended.

**2.9.2 Motion-Based Segmentation**

In one category of these algorithms motion information are used to segment the video frames, while in another group the motion and segmentation are estimated simultaneously. These two groups of algorithms are further described in the following:

**Segmentation based on motion information only**: In a classical approach to video segmentation, the dense motion vectors of image pixels are computed, and segmentation is then performed at the motion domain. Motion boundaries will detect the objects’ boundaries. Initially, these algorithms used the dense motion field information [175, 176]. However, due to spatially varying motion vectors even within a region, the parametric motion model produces better results. Consequently, later algorithms used the parametric motion model [160, 177, 178]. However, in all the above-mentioned algorithms the motion vector is produced independently and therefore the segmentation field should be extracted from the discontinuity of the motion vector. Furthermore, the temporal continuity of the segmentation field is not considered [3]. These algorithms do not use intensity, colour, texture or other spatial
information. Therefore except for simple scenes, the extracted borders do not coincide with the spatial borders, and they have the problems of motion estimation, such as the occlusion problem.

**Joint motion estimation and segmentation:** The motion and segmentation estimations are interdependent. Accurate segmentation and real border information will result in better motion estimation, and conversely, more accurate motion estimation results in a more accurate segmentation and motion vector field. This is an example of the “chicken and egg” dilemma [174]. To break this cycle, joint motion and segmentation estimation algorithms are proposed [6, 72]. In joint estimation, the algorithm simultaneously estimates the motion and segmentation. Practically, the algorithm alternates between motion estimation and segmentation label estimation.

### 2.9.3 Spatio-Temporal Video Segmentation

Although almost all video segmentation algorithms use motion/temporal information, spatial information can also be used. Spatial segmentation increases the pixel-wise accuracy necessary for the object-based processing. Therefore the spatial-temporal approach combines the spatial segmentation and temporal information to improve the segmentation result. The combined method is an adhoc, open problem, and many algorithms are proposed [6, 160, 179–181]. Although the results often coincide with the moving parts of objects boundary, the image is decomposed into some homogenous regions in terms of motion or grey/colour [181, 182]. Therefore some sort of pre- and/or post-processing are required toward semantic segmentation.

### 2.10 Semantic Video Segmentation

The region-based segmentation algorithms described in the last section are focusing on coding. The extracted regions are useful for region-based coding in terms of compression efficiency and reduction of blocking artifacts. However, with emerging object-based coding algorithms and object-based functionality such as interactivity and manipulation, semantic segmentation algorithms which divide the video into meaningful objects is required.
In these algorithms the motion and spatial features should come together to extract the meaningful objects with visually perfect boundary locations. However the semantic concept does not have enough correlation with homogeneity in ways such as colour, intensity. Therefore, some kind of user intervention is necessary, and fully automatic segmentation is not possible at this stage. However, the user intervention level can be reduced to a minimum such as a high level knowledge about the type of object. In one type of the semi-automatic semantic segmentation algorithms, called the tracking algorithm, the “object-of-interest” is determined in the first frame by some kind of user intervention, and then it is tracked in the subsequent frames [162, 172, 182–186]. In most of the automatic/unsupervised algorithms, the segmentation algorithm detects and track the moving objects without user intervention [5, 165, 181]. The problem with these algorithms is the gradual detection of the moving object’s regions. No region detection is possible if there is no movement. For example if some parts of an object do not move, they will not be detected perfectly. Therefore, in this thesis the extraction algorithms which include some kind of user intervention and tracking of the object in the subsequent frames are emphasised.

Special video tracking algorithms which use specific assumptions for detecting the VOP of interest in simplified or particular applications are not considered. For example, in many algorithms humans or cars are the “objects-of-interest” in a constant background [150, 187, 188]. In this thesis, these algorithms are not interested in, but the major approaches useful for generic object-based applications are analysed.

Another group of algorithms for semantic video segmentation detect the changed area in the frame and divide each frame to the foreground/background areas. Changed detection based algorithms have no pixel-wise accuracy, and need many post-processing such as removing the covered and uncovered regions from the foreground.

Hybrid algorithms track the detected objects in the previous frames and also based on the motion or changed area information, the newly appeared objects are also detected.
2.10.1 Video Object Tracking

There are many applications that benefit from motion tracking, including surveillance cameras, radar, air traffic control systems, security monitoring, etc. This process is a video analysis task corresponding to the segmentation of the detected video objects at previous frames of a video frames sequence. In interactive multimedia applications a user typically selects an object only once, and it is expected that the object is recognised in the subsequent frames by using information about its attributes and behaviour such as motion and colour/grey-level. The tracked objects have arbitrary shape and can change their shape over time. In addition pixel-wise accuracy is needed, therefore spatial segmentation and temporal information must be combined. The procedure can be divided into two levels:

I) Recognising the object in the first frame and II) tracking it through frame sequences using spatial and temporal information.

The first problem arises because it is necessary to recognise the “object-of-interest” in the first frame using only the available spatial information. Ordinary image segmentation methods use only homogeneity criteria, and the result is far from isolating the meaningful objects. Moreover, the homogeneity criteria are not unique and by changing it the result of segmentation changes. Therefore, research on semi-automatic methods requiring human assistance have attracted considerable attention. In semi-automatic methods, a user specifies an “object-of-interest” in the image (first frame). Of course, they try to minimise the intervention of the user. For example, in some works, it is enough that a user determines the object roughly, and then a spatial region/edge-based segmentation will find the correct borders with high precision [171, 189, 190].

Tracking algorithms often use motion/temporal information, and there are general problems with motion such as occluded regions that these algorithms need to overcome. There are some tracking algorithms that try to track several objects in a scene and analyse their behaviour, such as appearance, disappearance, overlap, collision, separation and stopping [161, 183, 185, 191, 192]. However, in this thesis the scope of analysis is limited to the extraction stage, because object motion and behaviour is
discussed often after the objects extraction stage.

Different algorithms for object tracking through frames are reported, and in the following section two major approaches are classified and described. The first approach is based on the edge or contours of the tracked object, and the second one is the region-based segmentation algorithm.

2.10.1.1 Edge-Based Object Tracking

These approaches rely more on the information closer to the boundary of the video object. A general problem of these algorithms is their performance in the cluttered or textured areas, which leaks accuracy and is computationally complex. Some of the outstanding edge/contour-based tracking algorithms are reviewed at the following:

A group of algorithms [17, 193] track the object’s edge pixels in the following frame. However the detected edge pixels do not necessarily make closed contours. Therefore a special algorithm is used to close the contours. This creates two shortcomings: the computational complexity and the pixel-wise inaccuracy of the contour closing process. Meier and Ngan [17, 193] present an edge-based tracking algorithm in two versions. In the first version, they separate the moving objects by considering the deviation from the global motion. After global motion estimation, each object moving differently from the background is a VOP. A morphological motion filter divides the image into the connected moving components. After detection, “moving connected components” smaller than a predetermined size are deleted, which performs a kind of noise filtering. In the next stage the edges of VOP(s) are extracted using the Canny operator. The shifted edges from the object in the last frame are then matched with the edges in the present frame. The Hausdorff distance criterion is used for finding the best match, and by shifting the detected object in the last frame, they find the best match in the current frame. This algorithm finds the location of the object in the new frame. Therefore all edge pixels very close to this shifted object are selected to belong to the object in the present frame. These pixels are related to rigid or slow moving components. However, edge pixels related to non-rigid moving components or to fast moving objects can be further than the shifted object’s edge pixels. For finding these pixels, Meier et al. use the rule that all edge pixels close to moving
components overlapped with the shifted old model belong to the updated model in the present frame. A post-processing step is carried out to produce objects from the unclosed edge pixels. All the pixels between two edge pixels in a row belong to the VOP. This is repeated for columns and once more for rows. Again a post processing is performed to correct the wrong boundaries. Some key parameters are entered by the user, so this is not an automatic procedure. The boundary edge gap is dealt with using Dijikstr’s shortest path [3], however there are still some gaps in the final model that cannot be connected, and this is not explained in the algorithm [174]. The computational complexity of different stages of the algorithm, such as the morphological motion filter and border gap closing, is too high.

A second version of the Meier [172] works is very similar to the Kim et al. work [183] which are more suitable for fast moving objects with stationary background. Kim and Hwang use a method which is based on the frame difference and Canny edge pixels. They extract the edges of the difference of two consequence frames by the Canny operator. All edge pixels from the current frame close to the edge of the difference image are selected as pixels of the tracked objects. To consider stopped objects they add all the edge pixels close to the objects in the last frame, which are not related to the background, as edge pixel of the tracked objects’ area. These pixels are related to stopped objects. For this purpose, the algorithm also tracks the background from the first frame. When the edge pixels of the tracked objects are determined they extract the video object. Because the edge pixels are not necessarily closed contour, each pixel between the first and last pixels in each row and each column are candidates to declare as object pixels. Finally, the real edge and object are separated by a morphological operation from the other candidate pixels.

In the last part of their work, Kim et al. [183] discuss the extraction of each object between several moving objects in the foreground. They have considered the mean pixel of the each object, and the closest means determine the “object-of-interest”. In the situations where the numbers of objects differ in subsequent frames, which means that objects have merged or separated, they rely on the smoothness of the motion vector between frames, and this criterion determines the “object-of-interest” including, its possible disappearance. One of the problem is related to the situation
where new object appears after a merge in the previous frames, which they suppose is a split. However, in some examples, it could be a new object. Therefore more discussion is necessary.

Erdem et al. [194] improve the accuracy of the extracted contour by the active contour model. Mazier et al. [190] try to extract the closed contour by using a mesh-based processing. Then by an active contour model the pixel-wise accuracy of the extracted contour is increased. More details of their works are reviewed in the following.

Erdem et al. [194] present a scalable object tracking algorithm. The algorithm can be adjusted to increase the pixel-wise accuracy or decrease the accuracy and computational complexity. The algorithm includes both open-loop and closed-loop processing. In the first phase, the contour of the object at frame $t$ is divided into sub-contours. For these purposes the algorithm of [195] selects good feature pixels such as pixels with high texture or high curvature, for tracking. Then the closest contour pixels to the best pixels are found. The found pixels divide up the contour. Then using motion information, these contour pixels are tracked to the next frame and for any two consecutive pixels, the transformation matrix between pixels at frame $t$ and $t + 1$ is computed. The pixels between two consecutive boundary pixels are also projected by the calculated transformation matrix. The transformed pixels make a contour at the next frame. At the closed loop boundary correction stage, this contour is refined with the active contour model, considering the colour segmentation, edges and motion information. The number of selected feature pixels is entered into the algorithm.

Another edge approach to object tracking is proposed by Maziere and Chassaing [190] which uses contours obtained from the snake model. In their algorithm a user first defines the exterior of object contours with a standard input device, and then the selected sketch is iteratively refined using a classical active contour model in order to accurately fit the natural edges of the objects. For tracking they define a hybrid model which uses a hierarchical mesh defined on the object. The first level of the mesh is built from the nodes of the contour model on the object boundary. The next level is built according to a node-based sub sampling and an edge constrained Delaunay triangulation. This process is iteratively repeated until a given number
of hierarchical meshes are obtained. The motion estimation is followed by motion compensation using an affine motion model for the triangular meshes which produces the first approximation of the object at the current frame. An active contour model is then built from the finest level of the mesh hierarchy, in order to improve the spatial accuracy of the object contour. This algorithm can extract the contour successfully for the large size objects. This problem comes from the node motion estimation and compensation which needs enough number of nodes. However, it can follow the different movements of internal areas of the objects.

Wang et al. [161] propose a multiresolution approach which decreases the computational complexity. After foreground/background separation, a rule based algorithm determines different objects. First the image is decomposed by wavelet decomposition. Then the global motion is estimated by a camera motion model. By adding the AC bands, the edges of two images at this level are obtained. By using global motion compensation the two edge images are aligned. By subtracting and thresholding the two edge images the foreground and background areas are found. This procedure is repeated for the next higher resolution level. The motion model of the last level is used as an initial value for higher level and only the parts of the image that have been classified as background in the lower resolution level are considered for motion approximation. This procedure continues until the highest resolution level is segmented. Finally the moving regions related to noise are removed. The main criterion is width and length of the region and the peak of edges in the region. If it is not large enough, a region is removed.

The tracking of more than one moving object in a scene is then pursued. It includes complex situations such as new track (new object), ceased track (object stops moving) and possible collisions (objects overlap). A rule-based method to deal with this situation is proposed. To discriminate between objects, they have defined the centroid of the object \( (C_x, C_y) \), the dispersion value, the mean of the grey scale distribution of the object and the texture. Based on the above-mentioned variables, the number of objects and the defined rules, the presence of a new object, collision and stopping can be determined. A more robust camera motion estimation method is needed. The proposed algorithm has not pixel-wise accuracy.
2.10.1.2 Region-Based Object Tracking

The other group of tracking algorithms classifies the objects’ regions. They try to spatially segment the image into several regions and establish the correspondence between different regions in consecutive frames using the spatial and temporal information which results in object extraction in the next frame. Although some proposals are based on temporal information [196], most approaches have tried to use spatio-temporal information. While many of the proposed tracking algorithms have pixel-wise accuracy [162, 164, 184, 189], some track the object’s bounding box with less computational complexity for special applications such as video surveillance, traffic control, autonomous vehicle guidance [186, 197]. For object-based applications such as coding or manipulation and editing, pixel-wise accuracy is necessary. Tracking algorithms use motion [162, 173, 189, 197–199], change detection [200, 201], Kalman filtering [197], the maximum entropy method [202], the hidden Markov model [203], etc., to establish the temporal linkage. In the following some of the outstanding tracking algorithms are reviewed.

In one group of tracking algorithms, the detected object at the current frame is projected to the next frame and the contour is refined [189, 197–199]. The problem behind these algorithms is considering a motion model for the object, while the tracked object can have various motion models. Normally an object is a combination of some segments with different movements. Difficulty in tracking of non-rigid and fast moving objects is the other shortcoming of these algorithms. Gu and Lee [198] propose an algorithm based on object motion estimation and projection from the previous to the current frame followed by object refining. Since the motion estimation near the object contour is known to be inaccurate, therefore, the projected object contour must be refined in order to obtain a more precise boundary. Hence the boundary refinement step is performed as follows. First, all pixels within a small width around the projected object boundary are marked as uncertain pixels. Then region growing is performed to assign the uncertain pixels to the object or background. This boundary refinement step assumes that the true object boundary exists within a threshold width around the projected boundary. There are some problems with this algorithm. Their basic assumption about boundary refinement is not true for all examples. In other
words it cannot be guaranteed that the true object boundary is within a certain width around the boundary.

Vigus et al. [199] use the Kalman filter to estimate the position of the object in the next frame. They track a simple or homogeneous intensity object such as a ball. Kalman estimation is completed by a spatial search and match around the estimated place. If the spatial search is not successful they use a region (split and merge) segmentation to locate the ball in the scene. Their method is very fast and suitable for real time applications. However, tracking only a simple homogeneous object is a shortcoming of this algorithm.

Park et al. [189] present a semiautomatic region-based tracking approach. Initially, they improve the first initialisation method of [198]. First the image is segmented into the homogeneous regions by using the MAP-based segmentation algorithm, and then a user select the regions as the “object-of-interest” by a graphical user interface like a mouse on the screen. Subsequently, tracking starts frame by frame. First the movement of the regions of the object in the last frame is obtained by a novel motion matching method. The histogram of each row and each column of the object bounding box is determined, and then the same vectors on the shifted bounding box at the present frame are calculated. A matching algorithm finds the best match for horizontal and vertical movement \((dx, dy)\) of the bounding box. For horizontal movement \((dx)\) a row vector histogram feature is used, and the same for the vertical columns. Using the obtained motion, the object of the previous frame is projected to the current frame. Finally, the edge has to be refined because of non-rigid movement and different motions of the regions of the object. The refinement is performed by a modified version of the morphological watershed algorithm performed at regions around borders.

Enriquez and Robles [197] propose a Kalman filtering based object tracking algorithm. The algorithm uses two sources of information, the intensity and the infrared image. The two tracking algorithms are performed independently, and the fusion gives the final result. In the first algorithm, the bounding box of the object in frame \(t - 1\) is matched, and the best place for the object in the frame \(t\) is found by the best correlation criterion. The result is further refined by a Kalman filtering algorithm.
the second tracking algorithm, the infrared image sequence is processed. The change detector finds the object’s place in frame $t$. Similarly, Kalman filtering refines the algorithms. In a simple fusion algorithm, the final object estimation is obtained from a simple convex combination of the estimations and covariance matrices [204]. The occlusion error is reduced by checking the difference between the Kalman inputs, in which the object’s place is estimated from motion information processing and the last frame object estimation. If the difference is more than a threshold, the Kalman input is replaced with the object estimation from the last frame. The algorithm is limited to stationary background. It does not have pixel-wise accuracy for object-based applications.

To overcome the above-mentioned deficiencies, in some works, the object is divided into several homogeneous regions, and then after motion estimation for each region, it is projected to the next frame. Finally, borders are refined. In these algorithms the motion vector of each region is separately extracted. Some of the problems of these algorithms are overlap between different projected regions, and increased complexity in extracting different regions motion vector. Some of the outstanding works related to this approach are reviewed at the following.

Lim and Ra [162] propose a forward tracking algorithm. They segment each object into several homogeneous regions, and for each regions a moving vector model is extracted. To accurately estimate an uncertain area, the algorithm uses two predictions for pixels based on the colour statistics in addition to the prediction based on the motion compensation. The pixels around the projected contour are examined. If the inverse motion model for the background pixels or inverse motion model for the projected object pixels assign different categories to a pixel (background or foreground) it is announced as an uncertain area. Similarly, the colour of pixels around the projected boundary is examined if an object pixel colour is similar to that of the background or similarly, a background pixels colour is similar to the object’s colour, the pixels is joined to the uncertain area. In the final step, the uncertain area is allocated to the object or background by using a watershed-based decision algorithm.

Venkateswaran and Desai [173] propose a region-based tracking algorithm. In the current frame, the spatial segmentation divides the image into different regions. The
number of region classes is estimated by minimising the validity criterion, which is the ratio of the average of intra clusters over the minimum of the intra-frame clusters. Then for each foreground region the affine motion model is estimated by minimising the least squares criterion, and the adjacent regions with a similar motion model are merged. Then the regions are projected to the next frame. A similar spatial segmentation at frame $t + 1$ is performed, and any regions with more than 75% projected object pixels are considered as object regions.

To increase the pixel-wise spatial accuracy of the extracted objects, a group of algorithms combine the spatial segmentation with temporal information. They classify the segmented regions. The problem behind these algorithms is computational complexity of spatial segmentation, and the need for global motion estimation and compensation. Some works of this approach are reviewed at the following.

The change detection algorithms do not have pixel wise accuracy. To increase it, the detected object regions are overlapped with the spatial segmentation [200, 201]. In [200] the assumption that the variation of the inter-frame difference of the stationary background is different from that of the foreground is used. It starts with a global motion estimation and compensation step. Spatial segmentation commences with a morphological opening-closing by a reconstruction filter. The morphological watershed algorithm detects the location of the object boundaries. To avoid oversegmentation, regions smaller than a threshold are merged with their neighbours. Finally, a foreground/background decision is made to create the VOP(s). Every region for which more than half of its pixels are marked as changed by the change detection algorithm is assigned to the foreground. At any pixel of the processed region, the hypothesis that the variation is different from the background variation is tested. Frame difference variation at the background is estimated from the area corresponding to the last frame background. The frame difference at the current pixel $s$ is estimated within a window with width $w$ that is centered at $s$. Then based on the variance comparisons the current pixel is classified as foreground or background. To track stopped objects’ regions, the detected object in the previous frame is projected to the current frame, and the segmented regions including object regions above a threshold are added to the object regions in the current frame. This allows tracking
an object even when it stops moving for an arbitrary time. In contrast, the technique in [205] will lose track after a certain number of frames, depending on the size of the group of frames and the memory length.

A region-based object tracking algorithm using a genetic algorithm is proposed by Hwang et al. [206]. Spatial segmentation is performed by a genetic algorithm using chromosomes and can be avoided being trapped at local optimums. Chromosomes corresponding to the object of the last frame are considered, therefore only chromosomes corresponding to moving object parts are evolved. This allows for eliminating redundant computation and facilitating a temporal linkage between two objects in two consecutive frames. Then foreground and background regions are determined. They have used a motion detection method which produces a change detection mask (CDM) according to [200] that dictates the foreground and background. Each region is background or foreground depending on the number of foreground pixels in that region and on comparison with a predetermined threshold. The connected foreground regions make the VOP(s). The genetic algorithm increases the computational complexity. The spatial segmentation part of the algorithm is performed separately without effective considering the temporal information. Therefore, the algorithm cannot assure discrimination between foreground and background for a cluttered background.

In some tracking applications such as surveillance and security control systems, the pixel-wise accuracy is not necessary, and real time performance of the algorithm is more important. Block-based algorithms are suitable for these applications. Two of these algorithms are reviewed at the following.

Lefver et al. [203] propose a semi automatic hidden Markov model (HMM) based object tracking algorithm. At first, the offline object learning is performed. The object is learnt in different sizes, and from different viewpoints and light conditions. Then the object tracking is performed. At the first frame, the object position is determined by the user intervention. At the other frames, the extracted object is simply projected to the next frame by the calculated speed $C$ as the initialisation step. The speed $C$ is equal to the difference of the object center at frame $t$ compared to frame $t - 1$. The bounding box of the object projected by the motion vector is divided
into 64 sub-windows. These sub-windows are examined by the HMM, and the score $P_i$ is calculated for each sub-windows [207]. A $P_i$ bigger than a threshold confirms the presence of the object at the examined sub-window. If the object is not found at any sub-windows, the speed vector is halved (a deceleration) and then doubled (acceleration) and the procedure, including projection and testing of the sub-windows, is repeated to find the object. If the object is not found in the several consecutive frames, the object is lost. If in several sub-windows the object is found, the object center is equal to the average of the sub-window centers. This algorithm does not have spatial segmentation and motion estimation therefore it is a fast algorithm for real time applications such as ball tracking. However, tracking of non-rigid or small objects is not accurate, because the HMM cannot capture the deformation perfectly.

Hariharakrishnan et al. [208] propose a backward block-based object tracking algorithm. The initial mask corresponding to the first frame is assumed to be given to the algorithm. Considering the small motion between consecutive frames, the motion and object are updated every $N$ frames. A block-based and backward motion estimation is used. A $16 \times 16$ block at frame $K + N$ is matched with the reference frame $N$. If the matched block is completely within the object or background area, the corresponding block is labelled as a seed, and otherwise it is labelled as an uncertain block. The uncertain block is divided into smaller blocks and new seed or uncertain blocks are estimated. This procedure continues until a $4 \times 4$ block size is reached. The object mask at frame $K + N$ is estimated by the union of all the blocks in the $K + N$ frame that lie within the object area at the $K$ frame is classified as object area. The extracted object area is refined with an occlusion and disocclusion detection algorithm. It does not have pixel-wise accuracy.

### 2.10.2 Change Detection

This approach divides the image into foreground and background regions. The idea is that for the detection of the moving pixels, the exact value of the motion vector is not important, and non-zero motion can identify foreground pixels in a stationary background image sequence analysis. In other words, the inter-frame differences of features such as luminance can detect foreground pixels. Therefore the change
detector is a simple approach which can detect moving objects/regions. It simply segments a video frame into changed and unchanged regions. The changed regions denote the foreground and the unchanged regions denote the stationary background. The main tool is the frame difference between the current and the last frame.

\[ FD_{k,k-1}(x,y) = I_k(x,y) - I_{k-1}(x,y), \]  

(2.16)

where I is an image feature such as luminance, colour, etc. To support the sequences with non-zero global motion, the global motion estimation and compensation should be performed, and then the frame difference will be computed. In the simplest form the frame difference is subjected to a threshold to partition the image into foreground and background areas:

\[ O_k(x,y) = \begin{cases} 
1 & \text{if } |FD_{k,k-1}(x,y)| > T \\
0 & \text{otherwise} 
\end{cases} \]  

(2.17)

where T is an appropriate threshold. However due to background noise, a simple thresholding algorithm easily creates small holes and many noisy small regions, while some pixels in the background area are also detected as foreground. The covered/uncovered background regions are also detected as foreground. Change detection is a binary foreground and background classification algorithm. Also, some movements are not equivalent to changes. For example, the interior of a homogeneous region may not be correlated with the detected changed region [2].

To reduce the above mentioned problems of the change detection algorithms different algorithms have been proposed. The most important modification relates to frame difference comparison. To increase the accuracy of the algorithm a statistical model is often successfully used to model the background, and a statistically-based comparison is more accurate [200, 205, 209]. For example in [200] a group of frames is first selected and the frame differences of these frames with respect to the first frame are computed. Then a fourth-order static test of the frame difference is performed to detect the changed area. The motion vector for the changed area is analysed to remove the uncovered background area and open/close morphological processing removes the small holes.
In the other methods, the multi-dimensional Gaussian probability density function models the background area [210, 211]. The model is fitted using the regions corresponding to the background of the last frame. Any feature such as grey-level or colour can be used. If the probability of the current pixel is over a threshold it is classified as foreground, and otherwise it is classified as background. If it is classified as a background pixel, the probability distribution function of the background model is updated.

2.10.3 Hybrid Video Segmentation

In a generic framework, these algorithms detect newly appearing objects in the scene and track already extracted objects in the video signal to segment the video signal. Most of these algorithms are an intelligent combination of motion-based or change detection segmentation with a tracking mechanism [5, 163–165, 181]. To be automatic and unsupervised, most of these algorithms track and detect only moving objects/regions [5, 165, 181, 182]. A small number of these algorithms present a semi-automatic approach to the full extraction of the semantic objects [163, 164]. There are some different approaches which are designed for special cases. For example, for a fully automatic object segmentation/tracking extract objects using blue screening (chroma keying), which requires video-object apparatus [194]. The other group uses 2-D shape information through training [212] onto the shape space to estimate the most likely object boundary at a certain frame [194]. Some of the outstanding works are reviewed in the following.

Patras et al. [181] introduce a MRF-based region labelling for video segmentation. The algorithm developed the traditional pixel-based MRF-based video segmentation to a region-based approach. It detects and tracks moving regions. The algorithm alternates between motion and labelling estimation. At first the image is segmented by the watershed algorithm, which results in a conservative over-segmentation of the image. The basins then are classified by the optimization of the MRF-based objective function. The objective function includes three terms. The first term expresses how well the current motion and label conform with the image intensities. The second term expresses the temporal constraint, and the third term expresses the spatial con-
The spatial constraint is controlled by a coefficient which denotes the length of the common borders between the two neighbouring regions. This coefficient gives another feature so that for large regions the emphasis is over the temporal behaviour while for smaller segments the emphasis is put on the spatial constraint. The optimization alternates between label estimation and motion estimation. Due to good initial estimation from the last frame, deterministic optimization methods are used. Labels are estimated using the ICM approach, and motion is estimated in a gradient based estimation. The main problem with the algorithm is the number of objects that need to be entered into the algorithm. The occlusion treatment should be considered and high computational complexity is the other problem. This algorithm tracks the moving regions. Stationary regions are not detected before their movement. To solve this problem, Patras et al. [163] introduce a semi-automatic video segmentation algorithm in which user intervention determines the semantic objects in the first frame, which is tracked in the next frames.

Tsiag and Averbuch [5] present a video segmentation algorithm for extracting moving objects from an image sequence. First a global motion estimation and then compensation is performed. Then the presence of a scene cut is tested. In the first frame of a video shot the algorithm is reset. A spatial segmentation, by the watershed algorithm over the colour gradient image, is performed. The watershed over-segmentation is reduced by a region merging algorithm. The merging criterion considers the spatial constraint as well as the temporal constraint. The temporal constraint prevents the merging of foreground regions to background and vice versa. A change detection algorithm detects the candidate foreground regions. A region is classified as a foreground candidate if more than 10% of its pixels are marked as changes, otherwise it is marked as background. A hierarchical motion estimation and validation finds and deletes the occlusion area. Then a MRF-based optimization over the candidate foreground regions determines the foreground regions. The MRF includes three terms. The first term contains a low potential term (negative) for the moving regions which are declared as foreground or non-moving regions which are known as background. The second term is a temporal continuity term which allows consideration of the segmentation of prior frames. If a region has been classified as foreground several times in the past, a low potential value is considered for the
region classification as foreground. The last term considers spatial continuity. A low potential value corresponds to the two adjacent regions with close average and similar foreground/background classification. The MRF objective function is optimized by a HCF method. The high computational complexity of the algorithm, especially for the global motion estimation and MRF optimization are some of the algorithm’s problems. Semantic regions are not detected before movement of the region.

**2.11 Conclusion and Research Direction**

In this chapter, a brief review of the various image and video segmentation algorithms, including low level and high level (semantic) stages, were presented. In low level image segmentation, the edge-based and region-based segmentation algorithms were described. The major approaches in region-based segmentation, including morphological and Bayesian based approaches, were explained. The focus was placed on multiresolution low level image segmentation and semantically-based image segmentation including “object-of-interest” extraction. Video segmentation approaches were discussed and the video object tracking algorithms in the literature were reviewed.

The aim of segmentation is partitioning the image into semantic object(s)/region(s) for further processing. Any general object extraction and recognition needs high level knowledge [24], but acquisition, processing, extending, applying and presenting the general low and high level information and knowledge, similar to the human vision and knowledge systems, is a very difficult task at this stage. As the literature review in Section 2.5.2 shows, the existing scene segmentation algorithms have many limitations and include many simplified assumptions about the objects that exist in the scene. Perfect and effective segmentation of a scene is, far from reality at this stage [140, 141] and the present algorithms can effectively segment only simple images. Therefore “object-of-interest” extraction has been the topic of much research in recent years [15, 138, 142, 144, 152].

As mentioned in Section 2.5, a comprehensive solution for object detection and extraction includes both low and high level segmentation. Both low and high level
stages of segmentation are active topics of research. This is due to the significance of segmentation in many applications and also the lack of a dominant segmentation solutions for general segmentation applications. As in the introduction chapter explained, there are three main areas of focus in this thesis which the existing literature does not offer effective solution. In the following the goals pertaining to these challenges are briefly reviewed and the selection of the proper approaches to achieve the goals are explained.

The first goal is proposing an effective low level multiresolution image segmentation which extracts and presents objects/regions at different resolutions, which are useful for spatially scalable object-based coding applications. Existing multiresolution segmentation approaches in the literature as mentioned in Section 2.4 are progressive, and low resolution results are refined at higher resolutions; therefore, the higher and lower resolution segmentations could be different. In other words, the refining of the result at higher resolutions has no effect on lower resolutions. In the best cases the algorithms consider the interscale correlation between the last or the next resolutions. These approaches are not effective solutions for multiresolution object-based applications such as scalable object-based coding algorithms. Therefore an effective low level multiresolution segmentation algorithm which maintains the similarity/scalability of the extracted objects/regions at different resolutions is necessary. This calls for a multiresolution refinement and interaction similar to the HVS mechanism. The HVS starts from low resolution, so that at first the global objects/regions at lower resolutions are detected, and then the detailed information at finer resolutions is extracted. To refine the lower resolution segmentations, there is also a feedback from finer resolutions to lower resolutions. The refined low resolution information again refines the high resolution perception. This progressive refinement from low to high and high to low resolution feedback continues iteratively until convergence. Therefore, as well as the traditional low to high segmentation refinement in the multiresolution segmentation algorithms, the high to low feedback to correct and optimize the segmentation at different resolutions is necessary.

The two main categories in the segmentation are edge-based and region-based algorithms. Edge extraction, has lower computational complexity than region-based
segmentation. However the necessary pre- and post-processing in edge-based segmentation such as removing useless short length edges and edge linking to create the closed contours are complex. They are very sensitive to noise, and to prevent over/under-segmentation, some parameter tuning is often done manually. In contrast the region-based segmentation approaches have more computational complexity, better noise tolerance, and more flexibility for imposing different constraints. In particular, region-based multiresolution segmentation algorithms can be implemented effectively. Therefore region-based multiresolution segmentation approaches are employed. As explained in Section 2.2.2 there are many different region-based approaches, but morphological and Bayesian based approaches are more effective and give satisfactory results. Morphological functions can capture the geometry of shapes/regions and Bayesian approaches capture the statistics of the image, Bayesian approaches in particular are very well suited to multiresolution approaches. Therefore, two morphological and Bayesian based multiresolution segmentation approaches are proposed and analysed. The proposed segmentation algorithms are extended to segment colour images.

The next goal is to enhance the visual quality of the segmentation. Multiresolution object extraction and resolution scalability extends the visual quality to multiresolution. Visual quality definition is a challenging area, and in this work the borders smoothness is suggested as a criterion which has correlation with the visual quality. The smoothness criterion should be imposed to the segmentation algorithm. This is required for both image and video segmentation algorithms.

The next goal is to extract the image “object-of-interest”. Using the region-based segmentation approach, the segmented regions are examined to extract the “object-of-interest”. An object can include several regions. Therefore the combination of regions should be examined. For region combination examination, a shape comparison algorithm is needed. The comparison should be translation, scaling and rotational invariant. Search over all possible region combinations, extracts the “object-of-interest”. However, exhaustive searches over the entire image, has high computational complexity, and there are not many effective algorithms for search. Therefore an effective search in the image for the “object-of-interest” detection is a topic which
needs more research. The suggestion is to perform a directed search in the image through defining a hierarchy of the objects/regions to be examined. In this regard, multiresolution search through a pyramid can be an effective solution. Therefore, effective low level multiresolution image segmentation and priority-based search over the segmented pyramid are the selected approaches to achieve the goal. Lack of sufficient information at low resolution is the problem of multiresolution approaches.

To extend the proposed approaches to the video domain, a region-based multiresolution video object tracking and extraction algorithm is proposed that extracts objects at different resolutions with scalability and smoothness as two constraints. Region-based approach increases the spatial accuracy of the segmentation. In addition the number of regions is much less than the number of pixels, which decreases the computational complexity, a critical problem for the image sequence segmentation algorithms. To remove/correct the invalid motion vector corresponding to the regions to be covered, the backward tracking is selected. To detect the newly appearing objects, the global motion compensation followed by local motion detection is used. The smoothness criterion will be imposed on the regions classifications. For regions classification/decision the MRF-based optimization is considered which is very flexible to impose different constraints such as visual quality, spatial and temporal continuity, etc.

Finally, reducing the computational complexity of the proposed algorithms is an important goal. The proposed multiresolution frameworks are considered to reduce the computational complexity. Hierarchical search over the image to extract the “object-of-interest” and replacing the pixel-wise processing to region-based processing in video segmentation algorithm significantly decrease the computational complexity.
Chapter 3

Towards Scalable Multiresolution Image Segmentation

3.1 Introduction

The major challenges in the multiresolution scalable segmentation process is ensuring similar segmentation patterns for different objects/regions at different resolutions. This requirement is essential for scalability. Many recently developed scalable object-based coding schemes need well segmented objects at different resolutions. Traditional multiresolution segmentations fail to achieve this requirement, and the segmented objects at lower resolutions suffer from distortions. To overcome this problem, a novel bidirectional projection is proposed.

As discussed in the literature review chapter, morphologic and Bayesian based segmentation algorithms are two major region-based image segmentation approaches. Therefore in this chapter two novel morphological and Bayesian based multiresolution segmentation algorithms are proposed. The results are compared with multiresolution segmentation algorithms in the literature. The flexibility of the segmentation algorithms to allow compatibility with the scalability constraint is surveyed for further algorithm development.

The proposed morphological segmentation method improves the segmentation results in terms of noise tolerance and computational complexity as well as overcoming
the problem of over-segmentation. This method is similar to traditional multiresolution segmentation algorithms in terms of progressive projection and refinement from low to high resolution segmentation. The shortcoming of the proposed morphological and traditional hierarchical multiresolution segmentation methods to fully meet the scalability requirement is highlighted by the presented experimental results. However, the second proposed algorithm (Bayesian) improves the segmentation results while meeting the scalability criterion.

Section 3.2 explains the scalability concept and the down-sampling relation constraint between object masks at the different resolutions necessary for scalable object-based wavelet coding algorithms. Section 3.3 describes a morphology-based multiresolution image segmentation algorithm. It includes a single level segmentation algorithm for the lowest resolution. The algorithm is then developed for the segmentation of the higher resolutions. It includes the projection of a lower resolution segmentation to the next higher resolution and detection of the new objects/regions at the higher resolution. In Section 3.4 some simulation results are presented and the algorithm’s advantages/disadvantages and capability to satisfy the scalability constraint discussed. In Section 3.5 the development of a single resolution MRF-based algorithm to a novel MMRF segmentation is introduced. The section also includes the statistical image modelling and optimisation processes. Some experimental results are presented in Section 3.6. Finally, conclusions are drawn in Section 3.7.

### 3.2 Object-Based Wavelet Coding Scalability

Scalability is known as an efficient feature in decoding the compressed data at different data rate [10, 11]. This feature enables the decoder to decode parts of the bitstream in order to meet certain requirements such as resolution, and quality. It is useful for image/video communication over heterogeneous networks which require a high degree of flexibility from the coding system. In this heterogeneous structure, users with low performance requirements are only able to receive low quality and/or low resolution images and videos, while users requiring higher performance should be provided with higher quality and/or resolution of visual information.
Towards Scalable Multi-resolution Image Segmentation

There are three possible methods to achieve scalability; Signal-to-Noise Ratio (SNR) scalability, spatial scalability and temporal scalability. In the SNR scalability, a feature is defined in the encoded bitstream which allows the decoder to reconstruct the main parts of the encoded bitstream at lower frame rates. In spatial scalability the shapes and their texture information are decoded on the basis of a specific resolution. In this case, the resolution is determined in correspondence with the end user’s capabilities such as bandwidth, display resolution and so on. Figure 3.1 shows the spatial scalability concept of a scalable object-based codestream. In this figure the bitstream supports three levels of spatial scalability, and a scalable decoder would be able to reconstruct the object at any of these three spatial resolutions. The first part of the bitstream ($S_1$) is needed for decoding a low resolution version of the original object. By adding the parts $S_2$ and $S_3$ to the first part, two higher resolution levels of the image are achieved. Scalable image/video coding is used in different applications such as image/video database retrieval, video telephony, web browsing or low-bandwidth image communication systems such as telebrowsing and teleshopping where progressive coding enables the user to make a quick accept or reject decision.

Due to the multi-resolution signal representation offered by wavelet transforms, wavelet-based coding schemes have the potential to support SNR, spatial and temporal scalability. Over the past decade wavelet-based image/video compression schemes have become increasingly important and gained widespread acceptance. An example is the new JPEG2000 still image compression standard [213].

In multi-resolution image analysis and segmentation frameworks, the wavelet transform provides a scale-space analysis, and wavelet-based image decomposition provides a sequence of similar images at different resolutions which is useful for scalable multi-resolution object extraction and coding. This is due to the short length of the wavelet’s filters [214], which makes a low pass band (LL) image at a lower resolution, but similar to the main shape. This feature is called the self-similarity of the wavelet transform. In this work an odd length wavelet filter (e.g. 9/7) is used, where all shape pixels with even indices\(^1\) are down sampled for the (LL) low pass band [12]. For other filters with different down-sampling styles, the algorithms can

\(^1\)Supposing indices start from zero or an even number.
Towards Scalable Multiresolution Image Segmentation

<table>
<thead>
<tr>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
</tr>
</thead>
</table>

Figure 3.1 Decoding of an object-based spatially scalable codestream. Depending on the channel, monitor, etc., a resolution is selected and the proper codestream is sent to the decoder.

be adapted. Figure 3.2 further illustrates the wavelet decomposition of arbitrarily shaped objects when using an odd-length filter. It includes two horizontal and vertical decompositions. Horizontal decomposition examines any pair of pixels such as $(U, V)$ where $U$ and $V$ are horizontally neighbours. If $U$ is sited in an odd column index then $U$ is down sampled to the $(L)$ low pass sub-band and $V$ to the $(H)$ high pass band. Therefore in the first step, the original image shown in Figure 3.2(a) is decomposed into the two low pass and high pass sub-band shown in Figure 3.2(b) and (c). Then in a similar procedure but in the vertical direction, $L$ and $H$ are decomposed to the four sub-bands $LL$, $LH$, $HL$ and $HH$ depicted in Figure 3.2(c). As a result, if the LL sub bands are considered as the figures at different resolutions, every shape pixel has a corresponding pixel at the higher resolution, however, only pixels with even indices have corresponding pixels at the lower resolution. Therefore only $1/4th$ of the current resolution pixels have corresponding pixels at the next lower resolution. By considering the self-similarity of the wavelet transform, it is
Figure 3.2 Decomposition of a non-rectangular object with odd-length filters: (a) the object, shown in dark grey; (b) the decomposed object after horizontal filtering; (c) decomposed object after vertical filtering. The letters “E” and “O” indicate the position (even or odd) of a pixel in the horizontal and vertical dimensions.

straightforward to suppose that the pixels of a shape with even indices have the same segmentation classifications as the corresponding pixels on the next lower level.

The wavelet self-similarity extends to all low pass sub-band shapes at different levels. Therefore the above relationship between corresponding pixels is extended to shapes at different resolutions. Each pixel has corresponding pixels at all the higher resolutions and pixels with indices that are multiples of $2^n$ in both dimensions are down sampled to the next $n$ lower resolutions. A pixel and its corresponding pixels at the lower and higher resolutions form a set called corresponding pixels. The length of the corresponding pixel set depends on the pixels’ indices and it can be $1, 2, \cdots, n$ where $n$ is the number of levels in the pyramid. Due to the self-similarity of the wavelet transform, corresponding pixels at different resolutions have the same segmentation label. Figure 3.3 shows a 4th level pyramid, and some corresponding pixels are shown.

### 3.3 Morphology-Based Segmentation

MRF-based segmentation algorithms would usually result in a local optimum while finding the global optimum would be computationally expensive. Other problems
Figure 3.3 A 4 level pyramid where some of the corresponding pixels are shown by the similar colour and the dashed lines also connect them. Set $V_1$ shows 4 corresponding pixels at different resolutions. $V_3$ shows 3 corresponding pixels, and $V_2$ shows 2 corresponding pixels. $V_1$ is a pixel at highest resolution which has no corresponding pixel at the other resolutions. The number of corresponding pixels depends on the pixels indices.

such as the need for an initial segmentation estimation and proper capturing of the region edges should also be considered. Morphology-based segmentation algorithms do not have the problems associated with statistical segmentation algorithms, and they can also capture the geometry of the image. However, they suffer from over-segmentation and sensitivity to noise [57]. The over-segmentation is reduced by the region merging algorithms [61,118]. However further reduction of over segmentation and noise sensitivity can be achieved by multiresolution image segmentation. As the literature review chapter shows, there are not many morphological multiresolution image segmentation algorithms in the literature.

The aim of this section is to present a morphology-based multiresolution image
Figure 3.4 Flow chart of the proposed multiresolution segmentation algorithm.

The proposed algorithm removes the problems of over-segmentation, sensitivity to noise and also computational complexity. Furthermore, watershed contours match with the natural object/region border to obtain very well located and smooth borders. Ultimately the extracted objects/regions at different resolutions could be used for general image analysis applications.

The image is first decomposed by a wavelet transform using 9/7 tap filters. Initially, the lowest level of decomposition is segmented by a single resolution image segmentation algorithm followed by a hierarchical procedure where the low resolution segmentation is projected to the next higher resolution and then it is refined to match the object/region border. The new detectable objects/regions at the higher resolution are also segmented. The procedure continues iteratively until the highest resolution is segmented. In Figure 3.4 the flow chart of the whole segmentation algorithm can be seen. The next two sections describe the segmentation algorithm in details.
3.3.1 Morphological Single Level Segmentation

A gradient operator is applied to the low pass sub-band image of the lowest resolution of decomposition followed by a morphological watershed operator on the gradient image. The adjacent regions with dissimilarity less than the defined threshold are merged [61]. The dissimilarity criterion is the absolute value of the grey mean difference of the two adjacent regions which results in a homogeneous grey-level region. In order to further decrease the number of regions, two adjacent regions with slowly varying grey-levels around their common borders are also merged. Such regions have no valid edges between them. The existence of edges between two regions is tested by using a function of wavelet coefficients with the following formula [215]:

\[ M(x, y) = \sqrt{|W_{lh}(x, y)|^2 + |W_{hl}(x, y)|^2} \] (3.1)

where \( W_{lh} \) and \( W_{hl} \) are wavelet coefficients related to point \((x, y)\) in the horizontal and vertical \((LH \text{ and } HL)\) sub-bands on that scale. The maximum value of \( M(x, y) \) in the direction of the gradient at the point \((x, y)\) will determine an edge [214]. The mean of the \( M(x, y) \) across the common borders is calculated and if it is less than a threshold, the two regions are merged. A good value for the threshold is the minimum value of \( M(x, y) \) along the Canny edge pixels. This merge can produce inhomogeneous regions in a special case: inhomogeneous regions with slowly varying grey-levels are well detected as an object/region. Eventually, regions with sizes smaller than a threshold are deleted.

3.3.2 Hierarchical Morphology-Based Segmentation

In an iterative procedure, starting from the coarsest level, the segmentation of a lower resolution is projected to the next higher resolution. New detectable objects/regions at the higher resolutions are also identified. This procedure continues until the highest resolution level is segmented.

3.3.3 Projection to the Next Level

Using this algorithm, a lower level segmentation is projected to the next level. Each pixel could simply be projected to four pixels on the next level. However this simple
method creates coarse regions with poor results in the next level. The result could be improved by a complex post-processing algorithm such as is done in statistical approaches. However a low complexity projection procedure is interested. This is achieved by adapting the projection procedure to watershed basin regions in such a way that every pixel of a basin belongs to the same object/region. Therefore the borders of the objects/regions at the new higher resolution match the contours of the watershed. Producing thin, smooth and well located borders of regions is another advantage of matching with the watershed contours. To this end, the projection is carried out with a fast merging of the regions obtained from the watershed algorithm. The following description highlights the technique.

The catchment basins of the image at higher resolution are obtained by a watershed algorithm applied on the gradient image. Every pixel inside the regions of lower resolution is then projected onto 4 pixels at the higher resolution. In each catchment basin of the higher level, the number of projected pixels with the same label is counted. If the number of projected pixels with the same label is more than a predefined threshold, such as 50 percent of the region’s size, the region is labelled the same as the pixels; otherwise the basin is labelled as unknown. It is interesting that only basins corresponding to lower resolution pixels which are close to the borders of segmented regions can have more than one type of projected pixel label. Subsequently, regions with the same label are simply merged and regions labelled as unknown are merged with one of their neighbouring regions according to the least dissimilarity criterion. The adjacent regions can be inhomogeneous and the dissimilarity criterion should be applied only to the pixels that are near to the borders between two regions.

3.3.4 Projection Complexity Reduction

The computational complexity of the projection procedure can be decreased by producing a lower number of catchment basins using the watershed algorithm. The simple 1 to 4 projection and labeling is precise for the higher resolution pixels corresponding to lower resolution pixels inside the regions and away from the borders. Labelling other pixels is obviously more uncertain. Therefore the pixels corresponding to internal pixels at the low resolution do not need a complex process and could
Towards Scalable Multiresolution Image Segmentation

be simply projected onto 4 pixels at the higher resolution. Using the following modification to the projection algorithm, the processing is limited to uncertain area pixels and basins. Each border pixel at the low level is projected onto an $n \times n$ area, with $n > 2$, such as $n = 4$, to create uncertain areas at the next level. In certain pixels, the gradient is replaced with zero, and the watershed on the gradient image is then applied. The other stage of the projection is the same as before. This results in a big catchment basin inside certain areas while other small basins are around the projected borders or in the uncertain areas. The number of basins depends on the size of the uncertain areas and typically decreases to less than 25 percent, which results in a large reduction in the complexity of the projection process. Both cases of the normal and reduced number of basins can be seen in the Lena segmentation example in Section 3.4.

3.3.4.1 Detecting New Objects/Regions

Low pass texture filtering and resolution reduction in the wavelet pyramid representation result in some of the small and low contrast objects/regions not being accurately detected at lower resolutions. Therefore, by increasing resolution, new objects/regions could be detected or segmented. To consider this issue, regions obtained from the projection of the last level are re-segmented separately.

Regions are re-segmented into two or more regions by an algorithm similar to the one used for segmentation at the coarsest level. The complexity of the algorithm for each region depends on the size of the regions. If each region’s size is much smaller than the entire image at the corresponding resolution, a re-segmentation of all projected regions has much lower complexity than provided by the normal single level segmentation algorithm for the image at that resolution. In addition, since each region is segmented separately, they can be segmented in parallel. The newly detected regions can be in the neighbourhood of several other regions, and their similarity to the other regions should be examined. If the dissimilarity is less than a threshold, the two regions are merged. For example in Figure 3.5, suppose the subregion $D$ in the region $A$ is detected at the current resolution, while due to the small size of region $D$, the filtering effect of the wavelet transform and the selected parameters
value, it is not detected at the lower resolutions. Also assume that the features of this subregion are more similar to the features of regions B and C than to region A. In the merging process, D is merged with regions B and C, therefore B and C are also merged. Actually, region D connects regions B and C appropriately so they are merged. These mergings of the regions decrease the number of regions at the highest resolution of the processed image. The merged regions have different parents and the parent-child relationship between resolutions cannot be fitted within the quad tree structure, which is used often in the multiresolution algorithms [84, 88, 98, 106].

### 3.4 Simulation Results

In this section, using the proposed algorithm, the Lena image and the 5th frame of the Table_Tennis SIF sequence are segmented. Threshold selection affects the segmentation results. Large thresholds result in under-segmentation and small thresholds produce over-segmentation. Selecting proper threshold values is often a research subject. In this example, considering the texture areas of Lena’s hair and the wall background of the Table_Tennis images, the selected thresholds are 30 for dissimilarity of regions and 8 for the mean value of \( M(x, y) \) on the borders to test the edge validity. These are relatively large thresholds.

At first the projection procedure is explained by the projection of 128 × 128 Lena im-

---

**Figure 3.5** (a) The projected segmentation at the current resolution; (b) sub-region D in region A is detected; (c) regions B, C and D are merged.
Towards Scalable Multiresolution Image Segmentation

Figure 3.6 Projection to catchment basins of Lena $256 \times 256$: (a) Lena image segmentation at $128 \times 128$; (b) Catchment basins imposed on the Lena image at $256 \times 256$; (c) The projected basin by the label of corresponding pixels of the lower level. The thick borders are the results of lower level borders projection to higher level.

Figure 3.7 (a) The reduced number of catchment basins projected from the lower level segmentation pixels; (b) merge of basins with the same label imposed on the main image, unlabelled basins in projection are shown with white colour; (c) the final projected segmentation to $256 \times 256$ level.

Image segmentation to $256 \times 256$ image in Figures 3.6 and 3.7. The segmentation of the lower resolution at $128 \times 128$ pixels is shown in Figure 3.6(a). The extracted catchment basins for $256 \times 256$ are shown in Figure 3.6(b). Every pixel of the catchment basins is labelled by the class of the corresponding pixel at the lower level resolution segmentation. Figure 3.6(c) shows the labelled basins. Each grey-level shows one of the lower level segmentation regions projected to the higher level. The same procedure can be followed with the reduced number of basins and the projection from
Figure 3.8 Segmentation of Lena in three resolutions: (a) $64 \times 64$; (b) $128 \times 128$; (c) $256 \times 256$ (d) The Lena Image at $256 \times 256$ resolution.

Figure 3.9 Segmentation of the 5th frame of SIF sequence Table_Tennis in three resolutions: (a) $60 \times 88$; (b) $120 \times 176$ and (c) $240 \times 352$ segmentation, (d) the original image of the 5th frame of SIF sequences table tennis.

the low level segmentation classes to reduce the number of basins as shown in Figure 3.7(a). In Figures 3.6(c) and 3.7(a), the thick black lines are the pixels projected from the border pixels at the lower level segmentation in Figure 3.6(a). It can be seen that in Figure 3.7(a) most of the catchment basins are around the projected borders of the low resolution image, or in other words, in the uncertain areas. In this example the number of basins has decreased from 5062 to 1548, which is about a 70 percent decrease in the number of catchment basins. The result of merging the similarly labelled regions (shown with the same grey) in Figure 3.7(a) is seen in Figure 3.7(b). The unlabeled basins do not take part in the merging process. They have been shown clearly in white in this figure. They are merged with their neighbouring regions us-
ing the least dissimilarity criterion to create the final projection. Figure 3.7(c) is the image of the final projection of the $128 \times 128$ segmentation onto the next higher level at the $256 \times 256$ resolution. The detection of new objects/regions will create the final segmentation. The final segmentations for the three levels at $64 \times 64$, $128 \times 128$ and $256 \times 256$ are seen in Figure 3.8. In this figure the pixels of each region are replaced by the mean of the grey-level values of that region. The numbers of regions are 25, 39, and 46 at the 3 resolutions.

It can be seen that, through edge validity examination in the algorithm, some inhomogeneous regions with slowly changing grey-levels such as Lena’s shoulder are detected as a region. As shown, some regions are detected only at higher resolutions. For example, the left eye is not detected in the lowest resolutions, but it is detected in the higher resolutions. Lena’s hair and the wooden frame in the upper, right area of the background are separated from the background only at the highest resolution. Therefore, the low resolution segmentation maps are not the same as those at higher resolutions, rendering the algorithm insufficient for scalable segmentation.

The computational time for the proposed multiresolution segmentation algorithm and single level segmentation of the highest level resolution has a ratio of 1 to 4.5, which represents a big reduction in computational complexity. It should, however, be mentioned that a large proportion of this time is spent in detecting new objects/regions in higher resolutions, and if this stage be deleted the complexity reduction is about 12 times.

In the next example, the 5th frame of the Table_Tennis SIF sequence is segmented. The segmentation results are shown in Figure 3.9. At the lowest resolution $60 \times 88$ the ball, hand and edges of the table are not well detected, in the $120 \times 176$ the ball and most of the table edges are detected, but there is still a problem in detecting the fingers and some parts of the table edges. Finally at the highest resolution all objects/regions including the ball, tennis paddle, table, hand and arm are accurately detected. The numbers of regions at the three spatial resolutions are 8, 21 and 12 regions. It is interesting to note that the number of regions at full resolution is less than in its lower resolution segmentation. This is due to a merging routine related to some newly detected regions at the $240 \times 352$ resolution, which are merged as
explained in Section 3.3.4.1.

At the $120 \times 176$ resolution, the white border area of the table is not well detected as one region and some small parts of it are detected as different regions. However, at the $240 \times 288$ resolution all areas of the table’s border are well detected and the small regions of the table’s border, which were projected from the lower resolution, are merged with the other regions of the table’s border to display the white border area of the table as only one region.

The implementation result shows that the proposed algorithm solves the over-segmentation, noise sensitivity and computational complexity problems by region merging at the lowest resolution of the pyramid decomposition at the lowest resolution and a hierarchical segmentation projection algorithm for the segmentation of the other levels. However the best results are achieved at the highest resolution, and the lower resolution segmentation pattern is different from the higher resolution segmentation. Therefore the proposed algorithm is not useful for scalability applications. The spatial scalability has a pixel-wise definition, while the proposed algorithm classifies regions which are combination of different basins. Modifying the algorithm to choose inter-scale correlation and resolution scalability is very challenging and requires major modifications of the watershed segmentation algorithm. Since almost all multiresolution segmentation algorithms in the literature are progressive from low to high resolution, they provide the best results only at highest resolution. Therefore, generally they are not useful for multiresolution object extraction and application. In the next section, a novel MMRF-based segmentation algorithm with pixel-wise accuracy and more flexibility for scalability constraints is introduced.

### 3.5 MRF Based Scalable Multiresolution Image Segmentation

Markov Random Field statistical modelling is used in many image processing applications. In order to solve an image processing problem by the MRF technique, a statistical image model has to be fitted to the application which captures the intrinsic character of the image in a few parameters. Image/video processing problems,
including all uncertainties and constraints, can therefore be converted to a mathematical parameter optimisation problem [41].

3.5.1 **Statistical Image Model**

The main challenge in multiresolution image segmentation for scalable wavelet-based object coding is to keep the same relation between extracted objects at different resolutions as it exists between the decomposed objects at different resolutions in the shape adaptive wavelet transform which was described at Section 3.2. To meet these challenges, Markov random field modelling is selected as it includes low level processing at the pixel level and has enough flexibility in defining objective functions matched with the problem at hand [41].

First, the wavelet transform is applied to the original image and a pyramid of decomposed images at various resolutions is created. Let \( Y \) be the grey-levels of this pyramid’s pixels. The segmentation of the image into regions at different resolutions will be denoted by \( X \).

As mentioned earlier, considering scalability, a pixel and its corresponding pixels at all other levels have the same segmentation label. Therefore they can only change together during segmentation. To change the segmentation label of a pixel, the pixel and all its corresponding pixels at all other levels have to be analysed together. As a result, an analysis of a set of pixels in a multidimensional space instead of a single resolution analysis needs to be used. Instead of speaking of a set of pixels, in the multidimensional space the word “vector” is used for convenience. A vector includes corresponding pixels at different resolutions of the pyramid. A symbol \( \{s\} \) shows a vector which includes pixel \( s \). The dimension of the vector is equal to the number of corresponding pixels at different resolutions, which depends on the index of the pixels, and it can be 1, 2 or more. The variables in the segmentation procedure, such as intensity and segmentation label are easily extended to the defined vector space. However some basic definitions and functions such as the neighbourhood system, the clique function and the energy function need to be developed and modified to match with the multidimensional domain. The necessary developments are explained as follows:
Towards Scalable Multiresolution Image Segmentation

Two vectors \( \{s\} \) and \( \{r\} \) are neighbours if they have the same dimension and at any resolution, the pixels of \( \{s\} \) and \( \{r\} \) are also neighbours. This definition extends 4 or 8 neighbourhood system to the vector space. Figure 3.10 shows two neighbouring vectors.

In the next step, clique definitions are extended to vector space. Regular cliques include two pixels. Therefore the extended cliques include two vectors. Figure 3.11(a) shows regular one and two pixel clique sets. In Figure 3.11(b), the extension of one of these cliques to the array mode can be seen.

The extension of clique functions is achieved through the following steps: equation (2.7), as described in Section 2.3.2.3 in the literature review chapter, is used for cliques with length two at a resolution where pixels \( s \) and \( r \) are two neighbouring pixels on the same resolution level. Equation (3.2) below is defined for multiple levels, where \( \{s\} \) and \( \{r\} \) are vectors corresponding to two neighbouring pixels \( s \) and

\[
\vec{V}_1 = \{A_1, A_2, A_3\}
\]
\[
\vec{V}_2 = \{B_1, B_2, B_3\}
\]
Towards Scalable Multiresolution Image Segmentation

Figure 3.11 (a) Normal one and two pixels cliques sets defined at a single resolution; (b) A clique of two vectors with the vector dimensions equal to three.

r. The neighbouring pixels of the two vectors \( \{s\} \) and \( \{r\} \) at level \( k \) are denoted as \( s_k \) and \( r_k \). The lowest resolution on the vector \( \{s\} \) is given by \( M \), and its dimension is denoted as \( N \). A positive value is assigned to the parameter \( \beta \), so that two neighbouring pixels on the same level are more likely to belong to the same class than to different classes. Increasing the value of \( \beta \) decreases the sensitivity to grey-level changes [4].

\[
V_c(\{s\}, \{r\}) = \left( \frac{1}{N} \right)^M \sum_{k=M}^{M+N-1} (-1)^{L_k \cdot \beta} ,
\]

(3.2)

\[
L_k = \begin{cases} 
1 & \text{if } X(s_k) = X(r_k) \\
0 & \text{if } X(s_k) \neq X(r_k)
\end{cases} \quad s_k \in \{s\}, \ r_k \in \{r\}, \ r_k \in \partial s_k \quad (3.3)
\]

It should be noted that the clique definition is extended to vector space or to multiresolution mode.

After the development of the neighbourhood system and the clique function, the procedure which extracts the Bayesian based single resolution segmentation described in Section 2.3.2.3 in the literature review chapter can similarly be used in the defined vector space to extract the objective function of the scalable segmentation algorithm. For example, the Bayes formula \( P(X|Y) \propto P(Y|X)P(X) \), the Gibbs distribution assumption of \( P(X) \) and the white Gaussian stochastic modelling for estimation of \( P(Y|X) \) are all similarly correct and applicable. Therefore all stages for the MAP
estimation of $P(X|Y)$ and the objective function extraction procedure are very similar. To prevent repetition, only the final extracted objective energy function is shown in the following formula:

$$E(X) = \sum_{\{s\}} \left\{ ||Y(\{s\}) - \mu_X(\{s\})||^2 + \sum_{\{r\} \in \partial\{s\}} V_c(\{s\}, \{r\}) \right\}$$ (3.4)

The first summation is over the pyramid’s vectors while the second summation is over all neighbourhood vectors of vector $\{s\}$. The two vectors $\{s\}$ and $\{r\}$ are neighbours if pixels of $\{s\}$ and $\{r\}$ located at the same resolution are also neighbours. The grey-levels of pixels in set $\{s\}$ form a vector $Y(\{s\})$, and similarly $\mu(\{s\})$ and $X(\{s\})$ are the mean and segmentation label vectors, respectively. The extracted objective function should be optimised by one of the optimisation algorithms explained in the literature review chapter.

### 3.5.2 MAP Estimation

The Iterated Condition Mode (ICM) optimisation method [67] is used to minimise the objective function (3.4). After initial segmentation with the k-means clustering algorithm the segmentation estimation is improved using ICM optimisation [67]. In single resolution image segmentation, ICM optimises the objective function pixel by pixel in a raster scan order until convergence is achieved. At each pixel, the segmentation of the processed pixel is updated given the current $X$ at all other pixels. Therefore only the terms in the objective function related to the current pixel need to be minimised:

$$E(X(s)) = (Y(s) - \mu_X(s))^2 - \sum_{r \in \partial s} V_c(s, r)$$ (3.5)

ICM, as used in the single level segmentation algorithm by Pappas [4], has been changed to adapt to the scalable multiresolution segmentation algorithm. The objective function term corresponding to a vector of pixels is optimised given the segmentation of all other vectors of the pyramid. The resulting objective function terms related to the current vector are:
Towards Scalable Multiresolution Image Segmentation

\[
E(X\{s\}) = ||Y(\{s\}) - \mu_X(\{s\})||^2 + \sum_{\{r\} \in \partial(\{s\})} V_c(\{s\}, \{r\})
\] (3.6)

For each pixels \(s\) of a set \(\{s\}\), \(\mu_i(s)\) is estimated by averaging the grey-levels of all pixels that belong to the region \(i\) and are inside a window with width \(w\) centered at pixel \(s\). In the next higher resolution the window size \(w\) is doubled. The average of any pixel \(s\) and its associates at all other levels in \(\{s\}\) are used to classify the pixels of \(\{s\}\) with a label which minimises equation (3.6).

The overall algorithm is as follows:

- The initial segmentation of the pyramid is obtained by the k-means clustering algorithm.
- All the pixels in the pyramid’s pixels are processed progressively from lower to higher resolutions.
- At each resolution the intensity \(\mu_i(s)\) at each pixel \(s\) for all possible classes \(i\) with a pre-determined window size \(w\) is estimated.
- The estimation of \(X\) is updated pixel by pixel in a raster scan order until convergence is achieved. At the same time all the pixels belonging to \(\{s\}\) over all the resolutions related to \(\{s\}\) are updated.
- The algorithm then moves to the next higher resolutions and updates the estimates of \(\mu\) and \(X\) at this resolution and so on, until all resolutions are processed.
- The process of updating the segmentation labels from lowest to highest resolution is repeated until convergence is achieved.

The convergence criterion is the number of pixel labels updated at each resolution, which should be below a pre-defined threshold. Other convergence criteria can also
be used. The whole procedure may be repeated with a smaller window size until the minimum window size at the lowest level is reached.

Scalability and multi-dimensional analysis tie high and low resolution pixels together, so that high resolution refinement influences low resolution refinement, too. On the other hand, optimisation includes several stages of refinement from low to high resolution with decreasing window size. Therefore the proposed segmentation algorithm with its objective function and the optimisation routine performs repeated low to high resolution segmentation refinement and feedback from high to low resolution segmentation until convergence of the segmentation algorithm is reached.

The combination of the proposed objective function and the optimisation method provide effective low to high resolution and high to low resolution effect and interaction. These inter-scale interactions continue iteratively until convergence of the optimisation algorithm. Therefore the proposed objective function and optimisation method, together provide a reliable and scalable segmentation algorithm.

### 3.6 Experimental Results and Discussion

The proposed algorithm is tested using frame 5 of Table_Tennis SIF sequence, frame 15 of the Clair CIF sequence and the 256 × 256 Lena image. The results are compared with a regular single and multiresolution segmentation algorithms [4], and the proposed morphology-based segmentation algorithm presented in this chapter. At the first step, the image is decomposed into different resolutions. Then using the segmentation algorithm the regions are extracted for the higher level processing.

As the first example, the 5th frame of the Table_Tennis SIF sequence sequence is segmented. Figure 3.12 represents the original image and the result achieved by the Bayesian based single resolution segmentation algorithm [4].

Segmentation by the hierarchical multiresolution segmentation algorithm [4] is shown in Figure 3.13. Segmentation by the morphology-based multiresolution image segmentation algorithm introduced in this chapter is presented in Figure 3.9 in Section 3.4. Finally, segmentation achieved by the proposed scalable segmentation
Towards Scalable Multiresolution Image Segmentation

Figure 3.12 Table_Tennis image segmentation with $k = 6$ clusters and $\beta = 100$: (a) the main image; (b) single level segmentation.

Figure 3.13 Multiresolution segmentation of Table_Tennis image with $k = 6$ clusters and $\beta = 100$: (a) $240 \times 352$ segmentation; (b) $120 \times 176$ segmentation; (c) $60 \times 88$ segmentation.

Figure 3.14 Scalable multiresolution segmentation of Table_Tennis image with $k = 6$ clusters and $\beta = 100$: (a) $240 \times 352$ segmentation; (b) $120 \times 176$ segmentation; (c) $60 \times 88$ segmentation.

algorithms is shown in Figure 3.14.
Table 3.1 Number of regions in Table_Tennis segmentation.

<table>
<thead>
<tr>
<th>Seg. algorithm</th>
<th>60 × 120</th>
<th>120 × 176</th>
<th>240 × 352</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi Resolution</td>
<td>19</td>
<td>55</td>
<td>164</td>
</tr>
<tr>
<td>Scalable</td>
<td>42</td>
<td>83</td>
<td>184</td>
</tr>
<tr>
<td>Single level</td>
<td>19</td>
<td>73</td>
<td>314</td>
</tr>
</tbody>
</table>

In the proposed algorithm the impact of higher resolutions on low resolution, decreases the under-segmentation phenomena which regular multiresolution segmentation algorithms suffer from and results in the detection of objects/regions that are not detectable otherwise. In other words, the sensitivity to grey-level changes is increased, resulting in a better detection of small or low-contrast objects especially at low resolutions. Table 3.1 shows the number of detected regions at three resolutions by single, regular multiresolution and scalable segmentation algorithms. The proposed scalable segmentation detects more relevant regions than the regular multiresolution algorithm. For example, consider the segmentation of the textured wall and detection of the ball in the Table_Tennis image. The single-level segmentation detects the ball, but it also detects a number of spurious regions due to the textured background, as the number of regions shows in Table 3.1. This drawback is called over-segmentation. The regular multiresolution algorithm misses the ball at different resolutions. The proposed algorithm, however detects the ball, altogether as well as avoiding unsightly segmentation of the textured background. The proposed scalable segmentation algorithm produces reliable and scalable segmentation results at different resolutions. For example, the ball is detected at the lowest resolution by the proposed algorithm, while it is not detected by regular Bayesian based and morphological multiresolution image segmentation algorithms.

In the second example, the first frame of the Clair CIF sequence is segmented. The original image and the single resolution segmentation of the Clair image are shown in Figure 3.15. Figures 3.16 and 3.17 show the segmented Clair image produced by the proposed scalable segmentation algorithm and regular multiresolution image segmentation.

Table 3.2 shows the number of detected regions of the Clair image at the three spatial
Towards Scalable Multiresolution Image Segmentation

Figure 3.15 Clair image segmentation with $k = 5$ clusters and $\beta = 50$: (a) the main image; (b) regular single resolution segmentation.

Figure 3.16 Clair image segmentation by the scalable segmentation algorithm with $k = 5$ cluster and $\beta = 50$: (a) $288 \times 352$ segmentation; (b) $144 \times 176$ segmentation; (c) $72 \times 88$ segmentation.

resolutions for different segmentation algorithms. The proposed scalable segmentation detects more relevant regions than the multiresolution method, and nearly the same number as single level segmentation. To test the scalable segmentation algorithm on noisy images, first a uniform noise signal in the range $(-30, +30)$ is added to the Clair and Table_Tennis images, then different segmentation algorithms are performed. The number of misclassified pixels for the Clair object including the head and shoulders (70553 pixels in high resolution of scalable segmentation) and the Table_Tennis object pixels including the arm, racket and ball (11033 pixels) are counted as well as the number of pixels in the entire image. The results in Table 3.3 confirm that the proposed algorithm can deal with noisy images as effectively
Figure 3.17 Clair image segmentation by the regular multiresolution segmentation algorithm with $k = 5$ cluster and $\beta = 50$: (a) $288 \times 352$ segmentation; (b) $144 \times 176$ segmentation; (c) $72 \times 88$ segmentation.

Table 3.2 Number of regions in Clair image segmentation.

<table>
<thead>
<tr>
<th>Seg. algorithm</th>
<th>$88 \times 72$</th>
<th>$176 \times 144$</th>
<th>$352 \times 288$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi Resolution</td>
<td>46</td>
<td>71</td>
<td>93</td>
</tr>
<tr>
<td>Scalable</td>
<td>72</td>
<td>98</td>
<td>116</td>
</tr>
<tr>
<td>Single level</td>
<td>46</td>
<td>94</td>
<td>138</td>
</tr>
</tbody>
</table>

Figure 3.18 (a) Noisy image of Clair, with a uniform noise in the range $(-30, +30)$ added to the images; (b) single resolution image segmentation; (c) scalable multiresolution image segmentation.

As multiresolution segmentation and much better than single level segmentation. It is significant to note that while maintaining noise tolerance, this algorithm has improved sensitivity to grey-level variation. Figure 3.18 shows the noisy version of the Clair images and its single and scalable segmentation results.
Table 3.3 Misclassified pixels in noisy images.

<table>
<thead>
<tr>
<th>Image</th>
<th>Clair</th>
<th>Table Tennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm</td>
<td>MRSes</td>
<td>Scalable</td>
</tr>
<tr>
<td>Object.</td>
<td>%11.98</td>
<td>%9.8</td>
</tr>
<tr>
<td>Image</td>
<td>%17.34</td>
<td>%17.8</td>
</tr>
</tbody>
</table>

Table 3.4 Number of misclassified pixels for Clair and Table_Tennis image segmentation.

<table>
<thead>
<tr>
<th>image</th>
<th>Clair</th>
<th>Table_Tennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolution</td>
<td>72 \times 88</td>
<td>144 \times 176</td>
</tr>
<tr>
<td>Down sample.</td>
<td>%7.59</td>
<td>%4.85</td>
</tr>
<tr>
<td>Scalable</td>
<td>%4.95</td>
<td>%2.45</td>
</tr>
</tbody>
</table>

The proposed segmentation algorithm detects more object regions than regular multiresolution segmentation and less than single resolution segmentation. Therefore it achieves a good balance between under- and over-segmentation. In confirming this balance, different pixel labelling results achieved by the three segmentation algorithms are compared. To this end, the segmentations by the regular multiresolution algorithm at different resolutions are considered as a basis for comparison. Variations in segmentation by the proposed scalable and down-sampled single resolution methods are counted. Table 3.4 shows the comparison result for the Clair and Table_Tennis image segmentation results. It confirms that the scalable segmentation (SSeg) has less variation than the single resolution image segmentation (SRSeg). In other words, the similarity of scalable segmentation to multiresolution image segmentation (MRSeg) is greater than the similarity between the multiresolution and single resolution segmentation algorithms. This confirms that the proposed algorithm sits between the regular and single resolution image segmentation algorithms and has the good features of both approaches.

The proposed segmentation can be used in general segmentation applications. However, it is especially suited to scalable wavelet-based image object coding which allows us only the pixels belonging to an arbitrarily shaped object to be coded [80,216]. To facilitate “object-of-interest” extraction, in a semi automatic procedure the user determines the rough boundary of the “object-of-interest” through a graphical user interface (GUI). Subsequently, all the regions with a predetermined percentage of
Figure 3.19 Clair object extraction: (a) First selection; (b) object at 288 × 352; (c) object at 144 × 176; (d) object at 72 × 88.

their area inside this closed contour are selected as the regions belonging to the “object-of-interest”. Combining all the selected regions creates the final object. As an example, a user has roughly determined the “objects-of-interest” in Figure 3.19(a). The algorithm then determines the exact borders of the object in different resolutions as shown in Figure 3.19. All regions including the concave regions are detected well, overcoming a weakness some object detection algorithms such as the snake active contour model suffer from [217]. The extracted image object, Clair’s head and shoulder, can be coded by scalable object-based coding algorithms [80].

Integrating the high and low resolution information in the proposed scalable segmentation algorithm often results in a better capturing of the image structure than regular multiresolution image segmentation. For example the grey-level image of Lena at 256 × 256 resolution is segmented by the proposed scalable and regular hierarchical multiresolution segmentation algorithm [4]. Segmentation results are shown in Figure 3.20 and segmentation by the proposed morphological segmentation can also be seen in Figure 3.8 in Section 3.4 of this chapter. Considering the different regions in the image, especially the regions over the hat, it is clear that the proposed algorithm better captures the structure of the image, although due to the low contrast between the hat and the background over the hat, it is not fully segmented from the background. Better separation is possible by considering colour information which is explained in the next chapter.

In the last example three images are segmented. The original images and their seg-
Towards Scalable Multiresolution Image Segmentation

Figure 3.20 Multiresolution Clair image segmentation with \( k = 5 \) clusters: (a) original Image; (b) regular multiresolution segmentation [4]; (c) proposed scalable multiresolution segmentation.

mented image are shown in Figure 3.21. The first image is the first frame of the QCIF sequence Carphone. The original image and its segmentation are shown in Figure 3.21 (a), (b). It is segmented to 152 regions. The second image is the frame 50 of CIF sequence Hall Monitor. The original image and its segmentation are shown in Figure 3.21 (c), (d). The most parts of the walking man are successfully discriminated from background. However, due to the low contrast, the small areas around the left leg are mixed with the background. in the last example, the \( 256 \times 256 \) size Barbara image is segmented. The original image and its segmentation are shown in Figure 3.21 (e), (f). The textured areas in the image are divided to many regions. This shows that for the highly textured areas such as Barbara’s scarf, the texture segmentation is necessary.

3.7 Conclusions

In this chapter, two multiresolution segmentation algorithms were proposed. The first one is based on a morphological algorithm using a watershed operator and region merging. The proposed algorithm solves the over-segmentation, noise sensitivity and computational complexity problems by region merging at the lowest resolution coupled by a hierarchical segmentation projection over all other resolutions. Al-
Figure 3.21 Three images segmentation: (a) first frame of QCIF sequence carphone image; (b) carphone segmentation with $k = 7$ and $\beta = 50$; (c) frame 50th of the CIF size sequence Hall_Monitor; (d) Hall_Monitor segmentation with $k = 6$ and $\beta = 50$; (e) $256 \times 256$ Barbara image; (f) Barbara image segmentation.
though the algorithm produces good results at the highest resolution, however it fails to produces similar segmentation maps over all resolutions rendering it ineffective for scalable object-based coding. The experimental results in this chapter confirm the deficiency of the hierarchical segmentation algorithms to provide effective and reliable results for multiresolution object-based applications such as scalable object-based coding algorithms.

The second algorithm is based on discrete wavelet transform and multiresolution Markov random field (MMRF) modelling. To consider the correlation of “objects-of-interest” at different resolutions of the wavelet pyramid, with the scalability constraint, a multi-scale analysis is developed and incorporated into the objective function of MMRF segmentation algorithm. The multi-scale analysis integrates low and high resolution information together. It more effectively considers inter-scale correlation and captures the structure of the image compared to the regular multiresolution segmentation algorithm which often confronts the over-segmentation problem. The proposed algorithm improves the segmentation result, especially at lower resolutions of the decomposition, over regular multiresolution segmentation in both objective and subjective tests, yielding an effective segmentation that supports scalable wavelet-based object coding. The major contribution of this algorithm is to match the multiresolution segmentation results with the spatial scalability feature required by the wavelet-based object coding algorithm. In the next chapter the proposed MRF segmentation algorithm is further developed in term of the smoothness criterion to present visually pleasing objects/regions. The algorithm will be also extended to segment colour images, resulting in a better separation of the foreground and background regions.
Chapter 4

Further Development of the Scalable Segmentation: Smoothness and Colour

4.1 Introduction

In this chapter the proposed MRF-based scalable segmentation algorithm is further developed. The first improvement proposed is a new criterion for border smoothness to be incorporated into the objective function of the segmentation algorithm. Allowing for smoothness terms in the objective function at different resolutions improves border smoothness and creates visually more pleasing objects/regions at different resolutions. The second proposed modification is to extend the algorithm to cater for colour images. Colour information increases the discrimination and separation capabilities over intensity only segmentation. Examining the corresponding pixels at different resolutions simultaneously enables the algorithm to directly segment the images in the YUV or similar colour spaces where luminance is in full resolution and chrominance components are at half resolution. In these further developments, scalability is maintained as a constraint.

This chapter is organised as follows. In Section 4.2 the concept and estimation of the smoothness criterion is explained, and the development of the objective function integrating this new criterion is also considered. In Section 4.3, the proposed scalable
Further Development of the Scalable Segmentation: Smoothness and Colour

colour image segmentation algorithm, which includes statistical image modelling and optimisation processes, is discussed. Some experimental results and discussion are presented in Section 4.4, and finally, conclusions are drawn in Section 4.5.

4.2 Smoothness

Objects borders are one of the most important properties for visual perception. Many natural objects exhibit smooth borders/edges. Hence, to some extent there is a correlation between visually pleasing objects and edge/border smoothness. Psychologically, the smoother edges/borders increase the influence and visual quality effect of the segmentation result. Therefore in some edge/contour-based segmentation algorithms such as the active contour model and the “Canny” edge extraction algorithm, the extracted objects/regions edges or borders are smoothed [23, 212, 217].

Traditionally, in region-based image/video segmentation algorithms, the image features such as pixels grey-level or colour have been considered. In most of these approaches, emphasis is put on the accuracy of segmentation. However the shape delineation of objects/regions, and producing a well-pleasing objects/regions shape have not attracted enough attention. On the other hand, perfect segmentation, if not impossible, is very difficult and distortions created by wrong segmentation in region-based approaches can result in incorrect, rough and unpleasing borders/edges. For example in pixel-wise segmentation algorithms such as MRF-based algorithms, the segmentation algorithm sometimes cannot capture the object/region structure very well, especially in low contrast areas which can result in border fluctuation and unpleasant object/region extraction. Therefore, in the proposed region-based segmentation algorithm, a smoothness criterion is incorporated into the objective function, which improves the visual quality of the segmentation process.

Due to multiresolution object extraction applications such as scalable coding the smoothness constraint is emphasised by considering it in multiresolution analysis. At high resolutions, the large number of pixels ensure more visual quality for the segmentation. However, at lower resolutions the visual quality can suffer due to insufficient information and down-sampling distortion. Down-sampling distorts shapes
and cannot necessarily preserve their topology at lower resolutions for all possible shapes [218]. This is more critical for complex shapes in terms of number of perimeter to area pixels. For example in Figure 4.1 down-samplings of two digital circles are compared. It can be seen that down sampling of the better approximation of the digital circle at high resolution can result in a worse shape at lower resolution. Therefore, achieving visually pleasing objects/regions at higher resolutions does not necessarily ensure similar quality at lower resolutions. Hence, it is necessary to enhance smoothness at all resolutions.

The proposed smoothness definition is based on the border’s curvature, which is the rate of the angle change between a curve and the tangent line to the curve [212, 217, 219]. In a digital environment an estimation of curvature can be used. The estimation is explained in Figure 4.2. Minimising the proposed estimation of smoothness prevents visually unpleasing fluctuations in the border pixels. The multiresolution smoothness analysis is realised by different coefficients for different resolution smoothness terms in the objective function of the segmentation algorithm. Therefore the objective function in equation 3.4, extracted in Chapter 3, is further extended to includes the smoothness constraint as following:

$$E(X) = \sum_{\{s\}} \left\{ (Y(\{s\}) - \mu_X(\{s\}))^2 + \sum_{\{r\} \in \partial(\{s\})} V_c(\{s\}, \{r\}) + \sum_{q \in \{s\}} l_{res(q)} \ast \nu(q) \right\}$$

(4.1)

where $Y$ is the grey-intensity function and $\mu$ is the grey-intensity average function. $\nu(q)$ shows the curvature estimation of pixel $q$, a pixel of vector $\{s\}$, and $l_{res(q)}$ is a coefficient which decreases when resolution increases. Therefore for ICM optimisation, the objective function at vector $\{s\}$ is equal to:

$$E(X\{s\}) = (Y(\{s\}) - \mu_X(\{s\}))^2 + \sum_{\{r\} \in \partial(\{s\})} V_c(\{s\}, \{r\}) + \sum_{q \in \{s\}} l_{res(q)} \ast \nu(q)$$

(4.2)

The proposed smooth object extraction is different from the simple border smoothness proposed in [220], which is a filtering of the extracted video object’s shape to remove small elongation introduced during the segmentation process, in the following areas:
Our smoothing process is an integrated part of the segmentation algorithm and effects the segmentation outcome.

With sufficient contrast, the proposed algorithm produces borders that are more faithful to the region’s shape.

On some occasions, some background pixels are added to the foreground regions to produce better looking shapes, especially at low resolution.

The smoothness factor could be adjusted for different resolutions to produce visually pleasing shapes at different resolutions with scalability as a constraint.

Although, smoothness somehow modifies the borders to be more visually pleasing, on the other hand, considering the smooth edges/borders of real objects, the proposed criterion, allows the segmentation results to better capture the natural borders of the existing objects/regions in the image, especially in low contrast areas of the image. However, even if it can’t capture the real border, the extracted borders are much less visually annoying to the user. The experimental results in Section 4.4 confirm that the extracted region borders are more favorable than the other regular segmentation algorithms. Practically smoothness also removes the small and low contrast regions in the segmented image. In other words, it decreases the number of regions. Therefore too much emphasising of this criterion by considering large smoothness coefficients can result in semantic distortion. The suitable coefficients should be entered to the algorithm. Considering the smoothness criterion, requires measuring the smoothness function at each pixel which increases the computational complexity of the optimisation algorithm.

As an example of the smoothness effect in spatial segmentation, consider the circle in Figure 4.3(a). It has two grey-levels, 100 in the background area and 200 in the foreground area. A uniform noise in the range $(0, 50)$ is added to the background and subtracted from the object intensity. This noise changes the image from binary to grey-level and reduces the pixel intensity variation of the foreground to the background pixels. The image is segmented by the proposed algorithm at two resolutions $20 \times 20$ and $10 \times 10$. The lower resolution smoothness is augmented by decreasing the smoothness coefficients to zero for the highest level and increasing the smoothness
coefficients for lower resolution. The results are shown in Figure 4.3(c) and (d). In this example, the smoothness criterion has deleted some pixels of the shapes at different resolutions. The results can be compared with Figure 4.1(a) and (b), which can be assumed as regular segmentation results at two resolutions, $20 \times 20$ and $10 \times 10$. The proposed segmentation method extracts a more pleasing shape at lower resolutions, albeit sometimes adding some distortion at higher resolution. However, the large number of pixels at higher resolutions ensures more smoothness and visually pleasing objects.

### 4.3 Colour Image Segmentation Algorithm

Colour image has more information than grey-level images, which results in more reliable separation of foreground regions from background in colour image segmentation algorithms. In this section, the proposed scalable segmentation algorithm is developed to segment colour images. At the initial step, the MRF objective function for the colour image segmentation at single resolution is extracted, and then it is extended to multiresolution scalable mode. It has been recognised that selection of an appropriate colour space produces more perceptually effective segmentation results [19, 221]. In particular, segmentation in YUV or LUV spaces often produces more favorable results than in RGB space [19, 93, 221]. Many of the images and image sequences in the databases are in YUV format where Y is in full resolution while the U and V components are in half resolution. The fact that the Y, U, and V channels are presented at different resolutions is not considered in any of the existing regular single or multiresolution colour image segmentation algorithms. However, this fact calls for a specially fitted multiresolution algorithm to perform the segmentation task effectively. The proposed algorithm has enough flexibility to directly segment this format of colour images.

#### 4.3.1 Statistical Colour Image Model

Considering the high flexibility of MRF modelling in solving different problems in image processing, the task is formulated by MRF modelling and a MAP criterion for
segmentation estimation. For simplicity, the statistical model of the single resolution colour image segmentation is first explained, and then it is developed to the scalable multiresolution segmentation mode\(^1\). The desired segmentation is denoted by \(X\), and \(Y\) is the observed colour image with three channels shown by a three dimensional vector \(Y = [Y_1, Y_2, Y_3]\). According to Bayes rule, the a posteriori probability density of the segmentation variables can be written as

\[
P(X|Y) \propto P(Y|X)P(X) ,
\]

where \(P(X|Y)\) represents the conditional probability of the segmentation label, given the observation \(Y\). By assuming the conditional independence of the channels given the segmentation field \([19, 102, 222]\), the probability of \(P(Y|X)\) is given by:

\[
P(Y|X) = P(Y_1|X)P(Y_2|X)P(Y_3|X)
\]

Then the conditional probability in equation (4.3) becomes

\[
P(X|Y) \propto P(Y_1|X)P(Y_2|X)P(Y_3|X)P(X)
\]

If a region is defined as the union of connected pixels with the same label from three colour channels, a unique label for each region can be considered. Therefore the label field \(X\) can be modelled by a regular MRF stochastic variable with one dimension. Using a four or eight pixel neighbourhood system considering only pairwise cliques, \(P(X)\) is then a Gibbs distribution [4] and is defined by its energy function \(U(X)\) such that

\[
P(X) = \frac{1}{Z} \exp \left( -\frac{1}{T} U(X) \right) , \quad U(X) = \sum_{c \in C} V_c(X) ,
\]

where \(C\) is the set of all cliques, and \(V_c\) is the clique potential function, as described in Section 2.3.2.2 in the literature review chapter, which encourages adjacent pixels to have the same segmentation label. If the statistical independence of the different pixels is assumed \([4, 19, 93, 172]\) then

\[
P(Y_i|X) = \prod_{s} P(Y_i(s)|X) , \quad i = 1, 2, 3
\]

\(^1\)After developing the clique function and neighbourhood system, the scalable colour segmentation algorithm can be directly extracted with the same procedure as the extraction of single resolution colour segmentation algorithm.
Further Development of the Scalable Segmentation: Smoothness and Colour

where \( s \) indicates pixels of the image. The conditional probability density \( P(Y_i(s)|X) \) is modelled as a white Gaussian process, with mean \( \mu_{X,i}(s) \) and variance \( \sigma_i^2 \) for channel \( i \). Each region is characterized by a mean vector \( \mu_X(s) = [\mu_{X,1}(s), \mu_{X,2}(s), \mu_{X,3}(s)] \) which is a slowly varying function of \( s \). Therefore \( P(Y|X) \) can be described by the following equation:

\[
P(Y|X) \propto \exp \left\{ -\sum_s \left( \sum_{i=1}^3 \frac{1}{2\sigma_i^2}(Y_i(s) - \mu_{X,i}(s))^2 \right) \right\} \tag{4.7}
\]

Considering equations (4.3), (4.5) and (4.7), the conditional probability density of the segmentation variable becomes:

\[
P(X|Y) \propto \exp \left\{ -\sum_s \left( \sum_{i=1}^3 \frac{1}{2\sigma_i^2}(Y_i(s) - \mu_{X,i}(s))^2 + \frac{1}{T} \sum_{r \in \partial_s} V_c(s,r) \right) \right\} \tag{4.8}
\]

Similar to single resolution segmentation, the parameters \( \sigma_i, i = 1, 2, 3 \), \( T \) and \( \beta \) in the clique function are interdependent. Therefore, to simplify the expression, the parameters \( 2\sigma_i^2, i = 1, 2, 3 \) and \( T \) are set to one, and the segmentation result is controlled by the value of \( \beta \) in the \( V_c \) function\(^2\). According to the MAP criterion the probability \( P(X|Y) \) should be maximised, which is equivalent to minimising the negative value of the argument of the exponential function in equation (4.8). This results in the following cost or objective function which has to be minimised with respect to \( X \):

\[
E(X) = \sum_s \left( \sum_{i=1}^3 (Y_i(s) - \mu_{X,i}(s))^2 + \sum_{r \in \partial_s} V_c(s,r) \right) \tag{4.9}
\]

To obtain the final segmentation, this objective function is minimised by one of the several MRF objective optimisation methods\(^4\). The rest of the procedure is similar.

\(^2\)The coefficients \( 2\sigma_i^2 i = 1, 2, 3 \) can be kept, but should be estimated in the segmentation algorithm. The rest of the procedure is similar.
Further Development of the Scalable Segmentation: Smoothness and Colour

resolutions will be denoted by \( X \). Similar to scalable grey-level image segmentation, considering resolution scalability, an analysis of pixels in a multidimensional space needs to be used. The term “vector” is used to refer to multidimensional space. The symbol \( \{s\} \) indicates a vector which includes pixel \( s \) and its corresponding pixels at different resolutions. The necessary development of a neighbouring system and clique function for this new vector space were explained in Section 3.5.1 in Chapter 3 for the scalable grey-level image segmentation algorithm.

As a result of the clique extension to multiresolution space, segmentation processing will continue in the vector space, therefore, intensity average and segmentation label functions are also extended to vector space. The intensity of pixels in different channels in set \( \{s\} \) form three vectors \( Y_i(\{s\}), i = 1, 2, 3, \) and \( Y(\{s\}) = [Y_1(\{s\}), Y_2(\{s\}), Y_3(\{s\})] \) defines the intensity \( n \times 3 \) matrix. Similarly, \( \mu_i, i = 1, 2, 3 \) and \( \mu(\{s\}) = [\mu_1(\{s\}), \mu_2(\{s\}), \mu_3(\{s\})] \) defines the mean vectors and the mean matrix. Therefore by a similar procedure which extracts the objective function of single resolution colour segmentation in equation 4.9, the objective function of scalable colour segmentation in the vector space is extracted as follows

\[
E(X) = \sum_{\{s\}} \left\{ \sum_{i=1}^{3} ||Y_i(\{s\}) - \mu_{X,i}(\{s\})||^2 + \sum_{\{r\} \in \partial\{s\}} V_c(\{s\}, \{r\}) \right\} \tag{4.10}
\]

The outer summation is over vectors, while the first inner summation is related to the distances of the pixel intensities from the estimated average for each channel of colour images. The second inner summation is over all neighbourhood vectors of vector \( \{s\} \). Considering the smoothness constraint at different resolutions, the equation is developed to the following:

\[
E(X) = \sum_{\{s\}} \left\{ \sum_{i=1}^{3} ||Y_i(\{s\}) - \mu_{X,i}(\{s\})||^2 + \right. \\
\left. \sum_{\{r\} \in \partial\{s\}} V_c(\{s\}, \{r\}) + \sum_{q \in \{s\}} l_{res(q)} \ast \nu(q) \right\} \tag{4.11}
\]

where, \( l_{res(q)} \) is a resolution dependent coefficient for \( \nu(q) \) the smoothness estimation function at pixel \( q \), where \( q \) is a pixel of the border \( \{s\} \).
For segmentation of colour images in YUV or similar colour formats, only the available components of colour data at different resolutions are used to classify the vector under one of the segmentation labels. In other words, if chrominance components such as U and V are in half resolution, the terms related to the chrominance components at the highest resolution are deleted from the objective function in equation 4.11, but tying the pixels together at different resolutions classifies the vector and pixels at different resolutions successfully. Considering the same argument, the objective function can be simplified to segment the grey-level image.

### 4.3.2 MAP Estimation

The segmentation is initialised with the k-means clustering algorithm for each colour channel separately. Then neighbouring pixels with the same equal labels for all three colour channels form a region. Similar to the scalable segmentation of grey-level images described in the previous chapter, the Iterated Condition Mode (ICM) optimisation method [67] is used to minimise the objective function and improve the segmentation estimation. Any other optimisation method can also be used.

Considering the ICM optimisation, the objective function terms corresponding to the current vector are optimised, given the segmentation at all other vectors of the pyramid. The resulting objective function terms related to the current vector are:

\[
E(X\{s\}) = \sum_{i=1}^{3} ||Y_{i}(\{s\}) - \mu_{X,i}(\{s\})||^2 + \\
\sum_{\{r\} \in \partial(s)} V_{c}(\{s\}, \{r\}) + \sum_{q \in \{s\}} I_{res(q)} \ast \nu(q) \quad (4.12)
\]

The same optimisation technique as described in Section 3.5.2 of the Chapter 3, for the grey-level scalable segmentation optimisation is used. Here, the difference is that during the optimisation process for each vector \(\{s\}\), 3 terms \(\mu_{i}\{s\}, i = 1, 2, 3\) for each colour channel are estimated separately.

To reduce computational complexity, it is sufficient to consider only labels of \(\{s\}\) and its neighbouring vectors to select the best label by the energy minimisation through equation (4.12). Therefore for the pixels inside a region, there is no computation, and
the region’s border is gradually refined\textsuperscript{3}. Furthermore, this border processing prevents isolated noise pixels from becoming a new cluster, resulting in fewer wrongly detected boundaries [79].

### 4.4 Experimental Results and Discussion

In this section, experimental results obtained from considering the smoothness criterion and using the algorithm introduced in Section 4.3 are presented. The results are compared with regular single-level and multiresolution segmentation algorithms [4, 19]. First, the image is decomposed into different resolutions, using the (9/7) wavelet filter. Then at each level of the decomposition, the image is segmented while scalability between regions at different resolutions, as required for the arbitrary shape wavelet transform, is achieved with the proposed algorithm. In the first three examples, the visual quality of the segmentation is analysed and discussed. In the next three examples, the colour segmentation where chrominance components are in half resolution is performed and discussed. In the last three examples, typical segmentations of natural images by the proposed segmentation algorithm are presented.

As the first visually pleasing segmentation example, the proposed algorithm is tested using frame 5 of the CIF colour images sequence Miss America. The image is in YUV colour format. At the first segmentation solution, considering only the available luminance components $Y$ at the highest resolution, a single resolution grey image segmentation is performed at the highest resolution. The original grey image and its regular single resolution segmentation are shown in Figure 4.4 (a) and (b). Some regions such as part of the hair are not detected, and border fluctuations occur in some areas of the image. Scalable segmentation without smoothness constraint is shown in Figure 4.4 (c). Multiresolution analysis decreases the border fluctuation, but there is still some degree of fluctuation. The scalable segmentation with the smoothness constraint at the highest resolution can be seen in Figure 4.4(d). The fluctuations and the unpleasant segmentation problems are removed, but for perfect extraction of the

\textsuperscript{3}This technique is not useful for grey image segmentation. In colour image segmentations, due to more available information from the different colour channels, there is over-segmentation. Therefore in colour segmentation detecting new regions is not of interest. However, in grey image segmentation, often there is under-segmentation and extracting new regions can be useful.
objects/regions the colour information is necessary.

To perform colour segmentation the chrominance components at $144 \times 176$ resolution are doubled ($1 : 4$ transform) and projected onto the $288 \times 352$ resolution. The original colour image is shown in Figure 4.5(a). The regular single resolution segmentation of the colour image in the YUV colour space is shown in Figure 4.5(b). The regular single resolution segmentation in the RGB colour space segmentation is seen in Figure 4.5(c) and the segmentation result of the proposed scalable segmentation algorithm in the YUV space is shown in Figure 4.5(d). The number of regions for different segmentation algorithms and the smoothness at different resolutions for both grey and colour segmentation are shown in Tables 4.1, 4.5, 4.3 and 4.4. These tables and a subjective test of the segmentation results confirm that the proposed scalable segmentation with the smoothness constraint extracts objects/regions with better visual quality than the other segmentation algorithms. The borders/edges of segmented regions defined by different segmentation algorithms are extracted by the Canny edge extraction algorithm and are shown in Figure 4.6. These edges coincide with the regions border. A subjective test of these edge images clearly confirms the efficacy of the proposed algorithm in the extraction of visually pleasant regions. Comparing the number of regions segmented also confirms the superiority of the proposed algorithm. While most meaningful regions are detected, the number of regions is less than with the single resolution segmentation algorithm.
Table 4.3 Miss America grey segmentation smoothness

<table>
<thead>
<tr>
<th>Segmentation</th>
<th>88 × 72</th>
<th>144 × 176</th>
<th>288 × 352</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single resolution seg</td>
<td>21.15</td>
<td>22.14</td>
<td>20.19</td>
</tr>
<tr>
<td>Scalable Seg</td>
<td>19.47</td>
<td>19.11</td>
<td>16.98</td>
</tr>
<tr>
<td>Improvement</td>
<td>7.95%</td>
<td>13.68%</td>
<td>15.86%</td>
</tr>
</tbody>
</table>

Table 4.4 Miss America colour segmentation smoothness

<table>
<thead>
<tr>
<th>Segmentation</th>
<th>88 × 72</th>
<th>144 × 176</th>
<th>288 × 352</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSeg with smoothness</td>
<td>21.24</td>
<td>19.96</td>
<td>16</td>
</tr>
<tr>
<td>Single resolution seg (RGB)</td>
<td>26.68</td>
<td>24.94</td>
<td>21.05</td>
</tr>
<tr>
<td>Improvement</td>
<td>20.40%</td>
<td>19.45%</td>
<td>23.96%</td>
</tr>
<tr>
<td>Single resolution seg (YUV)</td>
<td>22.35</td>
<td>21.76</td>
<td>17.8</td>
</tr>
<tr>
<td>Improvement</td>
<td>4.97%</td>
<td>8.24%</td>
<td>10.07%</td>
</tr>
</tbody>
</table>

Table 4.5 Number of regions for segmentation of grey Guitar

<table>
<thead>
<tr>
<th>Segmentation</th>
<th>64 × 64</th>
<th>128 × 128</th>
<th>256 × 256</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRSeg</td>
<td>82</td>
<td>189</td>
<td>447</td>
</tr>
<tr>
<td>MRSeg</td>
<td>82</td>
<td>98</td>
<td>157</td>
</tr>
</tbody>
</table>

In the second visually pleasing segmentation example, the 256 × 256 Guitar image is segmented. The original grey-level image and the single resolution segmentation are shown in Figure 4.7 (a) and (b). The segmentation result shows that many meaningful regions are not well detected, and border fluctuations occurs in some areas of the image. Figure 4.7 (c), (d) and (e) shows the multiresolution segmentation results. Although multiresolution segmentation has less border fluctuation, more under-segmentation means that more semantic regions are missed than with single resolution segmentation. Table 4.5 shows the number of regions for single and multiresolution segmentation algorithms at different resolutions.

To segment more/most meaningful regions, colour information is necessary in the segmentation algorithm. Figure 4.8(a) shows the original 256 × 256 Guitar colour image. The single resolution segmentation of the colour image in the YUV colour space is shown in Figure 4.8(b). 560 regions are detected with many non-meaningful regions, which indicates over-segmentation. Border roughness decreases for many regions creating a better visual quality. The segmentation results are given in Fig-
Further Development of the Scalable Segmentation: Smoothness and Colour

Table 4.6 Number of regions for segmentation of colour Guitar

<table>
<thead>
<tr>
<th>Segmentation</th>
<th>64 × 64</th>
<th>128 × 128</th>
<th>256 × 256</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSeg</td>
<td>139</td>
<td>308</td>
<td>447</td>
</tr>
<tr>
<td>MRSeg</td>
<td>139</td>
<td>143</td>
<td>172</td>
</tr>
<tr>
<td>SSeg</td>
<td>173</td>
<td>288</td>
<td>342</td>
</tr>
</tbody>
</table>

Figure 4.8(c), (d) and (e) which show the multiresolution segmentation of the colour image at different resolutions. While border smoothness is increased, many meaningful regions are not detected. This therefore over corrects the over-segmentation of the single resolution segmentation into under-segmentation. Many meaningful regions of the Guitar instrument and filing cabinet are not well detected and are mixed irreversibly with the background. The numbers of regions are shown in Table 4.6. Finally, the proposed scalable multiresolution segmentation with the smoothness constraint at three different resolutions is shown in Figure 4.8(f), (g) and (h). Most important and meaningful regions are extracted and the segmentation maps at different resolutions are similar. The borders are significantly smoother. Then numbers of regions for different colour image segmentation algorithms are shown in Table 4.6.

In the next visually pleasing segmentation example, the Office image is segmented. The original grey and colour images at 256 × 256 resolution are shown in Figure 4.9(a) and (c). It includes many objects such as books, computer, desk, a person’s head and shoulder. The colour contrast between the person’s jacket and the background is not very high. Therefore it is a relatively complex image for segmentation algorithms. The grey level single resolution image segmentation is shown in Figure 4.9(b). Some parts of the hair and right shoulder are not detected or not separated from the background. The single resolution colour image segmentation in YUV colour space is shown in Figure 4.9(d). It includes 952 regions, and over-segmentation has resulted in the detection of many non-meaningful regions. The resulting border roughness reduces the visual quality of the detected regions. The multiresolution colour image segmentation is shown in Figure 4.9(a), (b) and (c). Some important meaningful regions such as the ears are not detected, which shows the under-segmentation that can result from this algorithm. Finally, segmentation results obtained by the proposed scalable segmentation algorithm at different resolutions are
shown in Figure 4.10 (a), (b) and (c). The most important meaningful regions are detected. Segmentation maps at different resolutions are similar, and the comparison of region borders especially in the background area confirms that smoother borders and more visually pleasing regions are extracted by the proposed scalable colour image segmentation algorithm than the regular Bayesian based segmentation algorithm.

In the next three examples segmentation of colour images in YUV colour space with luminance components at full resolution and chrominance components U and V at half resolution is considered.

In the first colour space segmentation example, frame 34 of the Mother and Daughter sequence is segmented. The image is in QCIF format and is given in the YUV colour space. Regular colour-image segmentation needs the information in the same resolution. Therefore, in the first part of the solution, the image is segmented in grey-level space by a single resolution statistical image segmentation algorithm [4]. The result is shown in Figure 4.11(b). The left area of the daughter’s face has not been well separated from the background because there is not enough grey-level contrast between the face and the background. The same shortcoming is observed for the other grey level segmentation algorithms except when there is over-segmentation, which is not desired for segmentation applications. To successfully separate an object’s regions from the background, colour segmentation is performed as an alternative solution. The proposed scalable segmentation algorithm can perform colour segmentation using half resolution chrominance components. The result of segmentation by the scalable colour image segmentation is shown in Figure 4.11(d). The number of regions in grey-level segmentation is 273 while in colour segmentation it is 112, which shows a reasonable colour image segmentation algorithm.

In the next colour space segmentation example, the \(256 \times 256\) colour image of Lena in YUV space is segmented. The original image is shown in Figure 4.13 (a). In the first experiment, U and V are projected to \(256 \times 256\) resolution by a \(1:4\) pixel transform, and then single resolution segmentation is performed [19]. The result is shown in Figure 4.13 (b). It can be seen that the top part of the hat is not well separated from the background. Finally, the result from the proposed multiresolution scalable segmentation algorithm, which uses half resolution U and V, is shown in Figure 4.13(c).
Table 4.7 Misclassified pixels in Foreman image segmentation

<table>
<thead>
<tr>
<th>Resolution</th>
<th>72 × 88</th>
<th>144 × 176</th>
<th>288 × 352</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm A</td>
<td>50</td>
<td>186</td>
<td>786</td>
</tr>
<tr>
<td>Algorithm B</td>
<td>123</td>
<td>511</td>
<td>2118</td>
</tr>
<tr>
<td>improvement</td>
<td>59%</td>
<td>64%</td>
<td>63%</td>
</tr>
</tbody>
</table>

This algorithm can separate all the foreground (Lena) regions from the background successfully. It is interesting to note that the single resolution method divides the image into 578 regions, while the proposed scalable segmentation separates the image into 427 regions, which is a 26% reduction in the number of regions. This confirms that the proposed algorithm overcomes over-segmentation compared to single-level segmentations while still separating the objects’ regions from the background. Similarly, single resolution segmentation in RGB space cannot separate the hat from the background and divides the image into 779 regions with similar parameters. Considering the over and under-segmentation problems of single resolution and multiresolution segmentation algorithms respectively, failure to separate an object or a region in single resolution will surely lead to an even bigger chance of missing the object in multiresolution segmentation algorithm.

In the third colour space segmentation example, frame 30 of the QCIF sequence foreman is considered. The original image is in YUV format. In Figure 4.14(a) the original image can be seen. The image is segmented with the proposed scalable multiresolution segmentation. The result is compared with the single and multiresolution segmentation algorithms. To perform these algorithms, the U and V colour components are again projected to full resolution by a 1 : 4 pixel transform, and the regular single, multiresolution and proposed scalable segmentation algorithms are performed. The initial segmentation estimation comes from k-means clustering for different channels and the number of classes are chosen as \( k = 10, 4, 2 \) for the YUV or RGB colour channels used, respectively. In the first experiment the components U and V are projected to the next higher resolution, and then the proposed scalable and the regular multiresolution segmentation algorithms are performed. The results can be seen in Figure 4.14 (b) and (c). It is clear that in this example regular multiresolution segmentation cannot separate the foreground (foreman) from back-
Further Development of the Scalable Segmentation: Smoothness and Colour

ground regions. This is more pronounced in separating the left area of the hat from the background. Furthermore some other details such as the left eyebrow have not been detected. Similarly, the scalable segmentation with U and V projected to higher resolution could not detect the corner of the hat.

The image with the real full resolution size of U and V is segmented and will be considered as a ground truth for comparison in the following. The QCIF size U and V components are taken from the available YUV, CIF size image sequence. Segmentation by the scalable segmentation algorithm, which uses real full resolution QCIF size U and V is shown in Figure 4.14 (c). Figure 4.14 (d) shows the segmentation with the proposed algorithm which uses full resolution Y and half resolution U and V components. As can be seen, the proposed algorithm separates the foreground regions from the background successfully, as the scalable algorithm with the full resolution information does. As a statistical test, scalable segmentation using half resolution U and V (Algorithm A) and scalable segmentation using projected U and V (Algorithm B) are compared with the ground truth. The number of misclassified pixels in the proposed algorithm using half resolution U and V (Algorithm A) is about 30% of the ones achieved by the algorithm which uses the projected U and V components in high resolution (Algorithm B). The numbers of misclassified pixels at different resolutions are shown in Table 4.7. In Figure 4.14(e) the segmented image in RGB space using the full resolution information is shown. The right and top area of the hat are not separated well. To remedy the problem, the number of classes is increased from 10, 4, 2, to 10, 10, 10 classes to separate the hat, resulting in an increase in over-segmentation. Increasing the number of regions will increase the computational complexity of the segmentation algorithm. The number of regions with $k = 10, 4, 2$ is 279 for the proposed algorithm in YUV space and 337 for RGB space, which increases to 739 regions for $k = 10, 10, 10$ in RGB space.

In the last example three colour images are segmented. The original images and their segmented image are shown in Figure 4.15. First the colour image of the Car is segmented. The original $256 \times 256$ colour image in YUV format is shown in Figure 4.15 (a). The image is segmented by the proposed scalable multiresolution colour image segmentation algorithm and the result is shown in Figure 4.15 (b). It includes
276 regions, considering the textured areas of the image it is a good result. In the second example, the Lifting image is segmented. The original $288 \times 216$ image and its segmentation by the scalable segmentation are shown in Figure 4.15 (c) and (d). It includes 562 regions. The textured area of the wall increases the number of regions, although the textured loan and trees area are well segmented. In the third example, the colour image of House is segmented. The original $192 \times 256$ YUV colour image is shown in Figure 4.15 (e). The segmented image is seen in Figure 4.15 (f). The textured areas of building wall and grass are well segmented by the proposed scalable segmentation algorithm. The segmented image includes 329 regions.

4.5 Conclusions

In this chapter a new quantitative criterion for the segmentation algorithm was introduced. This criterion, which is a smoothness function based on the pixel segmentation labels, represents the visual quality of the objects/regions. Different smoothness coefficients, considered for different resolutions, extend the extraction of visually pleasing region to different resolutions. In addition to making the segmentation more visually pleasing, this criterion modifies the segmentation algorithm to better capture the structure of the objects/regions. The proper smoothness coefficients are entered to the algorithm and its automatic determination needs more research. Finally, considering this criterion increases the computational complexity of the segmentation algorithm.

The proposed scalable segmentation algorithm is developed to segment colour images. The proposed multi-scale analysis, incorporated in the objective function of Bayesian segmentation, improves the sensitivity to colour information variations while maintaining high performance in noisy environments. Different objective and subjective tests such as number of regions, discriminating between meaningful regions, smoothness and examination of visual attractiveness by measuring/estimating the smoothness function confirm the superiority of the proposed scalable algorithm over the regular single and multiresolution segmentation algorithms. The novel objective function gives flexibility to the proposed algorithm to segment YUV colour images where Y is in full resolution but U and V are in half resolution. The pro-
posed low level grey/colour scalable multiresolution segmentation is useful for the high level segmentation and “object-of-interest” extraction which is discussed in the next chapter.
Figure 4.1 Circles at different resolutions. Down sampling of pixels at higher resolution with even indexes, creates the shapes at lower resolution: (a) closer approximation of a digital circle at high resolution; (b) down sampling to low resolution; (c) worse approximation of a digital circle at high resolution; (d) down sampling of (c) to low resolution.
Further Development of the Scalable Segmentation: Smoothness and Colour

Figure 4.2 Curvature estimation: (a) corner point, $k=90$; (b) same direction $k=0$; (c) change direction point $k=45$. 
Figure 4.3 Scalable segmentation of a digital circle with an emphasis on low level smoothness: (a) original image; (b) noisy image; (c) segmentation at $20 \times 20$ resolution. (d) segmentation at $10 \times 10$ resolution;
Figure 4.4 Frame 5 of the Miss America CIF colour sequence segmentation (a) original image; (b) regular single resolution segmentation; (c) scalable segmentation; (d) scalable segmentation with smoothness constraint.
Further Development of the Scalable Segmentation: Smoothness and Colour

Figure 4.5 Frame 5 of the Miss America CIF grey sequence segmentation (a) original image; (b) regular single resolution segmentation; (c) single resolution segmentation in RGB colour space; (d) scalable segmentation with smoothness constraint in YUV colour space.
Figure 4.6 The edges extracted from the segmented image of Miss America by the Canny edge extraction algorithm: (a) edges of single resolution grey segmentation; (b) edges of scalable grey segmentation; (c) edges of single resolution colour segmentation in RGB space; (d) edges of scalable colour segmentation with smoothness constraint in YUV space.
Figure 4.7 Guitar segmentation: (a) The $256 \times 256$ grey image; (b) single resolution segmentation; (c) multiresolution $64 \times 64$ segmentation; (d) multiresolution $128 \times 128$ segmentation; (e) multiresolution $256 \times 256$ segmentation.
Figure 4.8 Guitar segmentation: (a) the $256 \times 256$ grey image of Guitar; (b) single resolution segmentation; (c) multiresolution segmentation at $64 \times 64$; (d) multiresolution segmentation at $128 \times 128$; (f) scalable segmentation at $64 \times 64$; (g) scalable segmentation at $128 \times 128$; (h) scalable segmentation at $256 \times 256$.
Figure 4.9 Scalable segmentation of colour Office image: (a) original office grey-level image at $256 \times 256$; (b) regular grey-level single resolution segmentation; (c) colour image of Office; (d) single resolution colour segmentation; (e) MRes segmentation at $64 \times 64$; (f) MRes segmentation at $128 \times 128$; (g) MRes segmentation at $256 \times 256$. 
Further Development of the Scalable Segmentation: Smoothness and Colour

Figure 4.10 Multiresolution Office segmentation by the proposed scalable segmentation: (a) 64 × 64; (b) 128 × 128; (c) 256 × 256.

Figure 4.11 Frame 34 of Mother and Daughter QCIF sequence segmentation with $k = 7, 2, 2$ clusters and $\beta = 40$: (a) original grey-level image; (b) regular grey-level single resolution segmentation; (c) colour image of Mother and Daughter where U and V are in half resolution; (d) proposed scalable segmentation.
Further Development of the Scalable Segmentation: Smoothness and Colour

Figure 4.12 Lena colour image at $256 \times 256$ resolution.

Figure 4.13 Lena image segmentation at $256 \times 256$ with $k = 6, 4, 4$ clusters and $\beta = 100$: (a) regular single resolution segmentation where $U$ and $V$ are projected to higher resolution; (b) proposed scalable segmentation where $U$ and $V$ are in half resolution.
Figure 4.14 Segmentation of frame 32 of Foreman QCIF sequence with $k=10,4,2$ clusters at different colour channels: (a) original image; (b) scalable segmentation where U and V are projected to higher resolution; (c) Regular multiresolution segmentation; (d) scalable segmentation where YUV are in full resolution; (e) scalable segmentation in YUV space where U and V are in half resolution;(f) scalable segmentation in RGB space where components are in full resolution.
Figure 4.15 Car segmentation: (a) original Car colour image at $256 \times 256$; (b) proposed scalable colour segmentation; (c) original lifting colour image at $288 \times 216$; (d) highest resolution segmentation by scalable algorithm; (e) original House colour image at $192 \times 256$; (f) highest resolution House image segmentation by scalable algorithm.
Chapter 5

Meaningful Image Object Extraction

5.1 Introduction

Recent advances in internet technology, digital imaging and storage devices have resulted in massive amounts of image and video acquisition, storage and transmission. In line with these advances, it is important to develop technologies toward effective content-based image/video processing tasks such as retrieval, interactive image editing and manipulation. The main challenge in implementing these processes is semantic segmentation and object extraction. Intelligent image manipulation requires object extraction and recognition and recognition in turn needs high level knowledge. Therefore semantic segmentation and “object-of-interest” extraction in a general scene are not a trivial task and have received great attention in recent years [15, 138, 142, 144, 152].

In this chapter a model-based semantic image segmentation is proposed which extracts a predefined “object-of-interest” from the image. The algorithm sorts, groups and compares the regions with the templates to find and extract possible object(s) of interest from the image. To reduce the computational complexity, the global precedence effect (GPE) of the human visual system (HVS) is considered [14, 223] and a hierarchy for object/region processing is presented. The proposed multiresolution scalable segmentation presents the segmented regions in a pyramid structure that allows us to process the low frequency/global information first, followed by
finer/higher frequency local information in a hierarchical structure. This chapter is organised as follows: Section 5.2 explains template search and matching in a single resolution segmented image. Furthermore, the computational complexity of the algorithm is analysed. The next section presents a proposal to consider the global precedence effect of the human visual system to reduce the computational complexity of the search algorithm. The experimental results and discussion including different examples are presented in Section 5.4. Static and dynamic or deformable templates are also explained in this section. The last section is the conclusion, which includes some proposals for future work.

5.2 Template Based Object of Interest Extraction at Single Resolution

2-D objects are naturally represented by their boundaries. Therefore shape or template analysis and matching are necessary for object recognition, which in turn, with a search algorithm, form some of the fundamental building blocks for the “object-of-interest” extraction algorithms. Object of interest extraction is often based on the minimisation of a suitable similarity measure between a reference such as a template and the group of regions in the test image. The comparison should be scale, rotational and translation invariant. There are many shape matching techniques in the literature, such as local features template matching, moment invariant, Fourier transform, generalised Hough transform, finite element analysis and modal matching. A survey of these methods can be found in [224, 225]. Some algorithms classify the object categories [127, 134, 224, 226] and others that have more computational complexity can recognise different objects in the same class [126, 141, 227].

Template matching is an approach to recognising the “object-of-interest” in digital images. In a real scenario, the “object-of-interest” is searched in a segmented image. Therefore, due to the huge number of possible region combinations, a simple shape matching algorithm is preferred. Also, error in segmentation of real and noisy images should be considered. Suitable matching algorithms should tolerate the segmentation errors in real images to some extent although local differences between shapes are
still important in shape classification and recognition.

A region-based shape matching is introduced which is a combination and modification of the two approaches proposed in [226] and [145]. The proposed algorithm measures the similarity between the template and segmented region combinations. Briefly, at first the regions with low similarity are rejected by an aspect ratio test. The affine transform of the region’s shape is then computed and the generalised Hausdorff distance [228] between the two shapes is considered as the their distance. The less the distance, the more similar the shapes. The algorithm is not very complex, and the Generalised Hausdorff distance makes it insensitive to small noise and local error, but it is interesting that the proposed algorithm is not a global decision algorithm, such as Fourier transform based shape matching algorithms, and to some extent it can see the local changes. The details of the algorithm are explained in the next subsections:

5.2.1 Aspect Ratio Test

The regions with a high degree of dissimilarity to the template of the "object-of-interest" are rejected by a simple algorithm. The covariance matrix of the border pixel coordinates is described by the following matrix:

\[
C = \begin{pmatrix}
\sigma_x^2 & \sigma_{xy} \\
\sigma_{yx} & \sigma_y^2
\end{pmatrix}
\]

The eigenvectors of \(C\) determine the major and minor axes of the shape [146]. The spread of the shape in the direction of the eigenvectors determines the major and minor axis lengths of the shape [146]. The ratio of the eigenvalues is computed for both the template and the candidate regions. If the two ratio values are not close, the candidate region will be rejected. To determine the values close to the aspect ratio value, a threshold is needed which can be set by the application’s user. Suppose that \(\lambda_{T_1}\) and \(\lambda_{T_2}\) are the template’s eigenvalues and \(\lambda_{R_1}\) and \(\lambda_{R_2}\) are the candidate region’s eigenvalues, also supposing that \(\lambda_{T_2} > \lambda_{T_1}\) and \(\lambda_{R_2} > \lambda_{R_1}\), then the following equation test the aspect ratios to reject the very dissimilar candidate regions:

\[
\frac{\lambda_{T_1}}{\lambda_{T_2}} \notin (K_1, \frac{\lambda_{R_1}}{\lambda_{R_2}}, K_2, \frac{\lambda_{R_1}}{\lambda_{R_2}}) \Rightarrow \text{Reject.} \quad (5.1)
\]

\(K_1\) and \(K_2\) are coefficients with values such as 0.8 and 1.2. They can be changed by the user, and their values are not very critical. This introductory test reduces the
computational complexity effectively. This is because only groups of region with a high degree of “aspect ratio” similarity to the template are passed to a more complex similarity test stage.

5.2.2 Affine Transform Normalisation

The proposed affine invariant matching includes rotational, scale and translational normalisation. This means that if one of the two similar/dissimilar shapes is translated, rotated or scaled the degree of similarity/dissimilarity is unaffected. Therefore the first stage of comparison is variation compensation. First the shape rotation is compensated. The idea is to find the major axes of the two shapes. The angle between the two axes determines the rotational angle factor. The major axis is a straight line which connects the two furthest pixels on the shape’s border. The template is rotated so that its major axis lie in the same direction as the candidate region’s major axis. In Figure 5.1 (a) and (b) two shapes and their major axes are shown.

The rotation normalisation can be done in the opposite direction which has a 180 degree difference. Also horizontally and vertically flipped shapes define the same shape. Therefore there are four possible results for the rotation normalisation and all of them should be examined. Figure 5.1 (a) and (b), shows two shapes, and four different rotationally compensated shapes corresponding to the shape in Figure 5.1 (a) for matching with the shape 5.1 (b), are shown in Figure 5.1 (c) to (f). All these four shapes should be compared with the template shape. There are other possible methods for rotation compensation such as modal matching [145] which finds only one compensated shape, but they have higher computational complexity.

The ratio between the two major axis lengths determines the scale normalisation factor. The shape’s size is normalised by the scale factor. Using a similar scaling approach, the shape is scaled in the major and minor directions so that both shapes have the same bounding box, which is the smallest rectangle containing the shape. Finally, the bounding box areas of the two shapes are translated to the origin. In Figure 5.1 (c) the bounding box of the shape is shown. Then after this affine compensation the shapes are ready for the comparison.
Figure 5.1 (a) a shape with its major axis is shown; (b) another shape which has its major axis in a horizontal direction; (c) a typical shape rotation normalisation. The shape is rotating $\alpha$ degrees to coincide with the direction of the major axis of the second shape. The shape bounding box is also determined; (d) rotation normalised and flipped horizontally; (e) shape c is rotated $180^\circ + \alpha$; (f) shape c is rotated $180^\circ + \alpha$ and then flipped horizontally.
5.2.3 Hausdorff Distance and Computational Complexity Reduction

The Hausdorff distance measures the distance between two sets of binary image pixels. Initially Huttenlocher et al. in [228] proposed to use the Hausdorff distance as a similarity/dissimilarity criterion for shape comparison. Let \( T = \{ t_1, t_2, \ldots, t_m \} \) denote the set of template border pixels and \( C = \{ c_1, c_2, \ldots, c_n \} \) be the border pixels of the candidate region of the segmented image. The Hausdorff distance is

\[
H(T, C) = \max \{ h(T, C), h(C, T) \} \tag{5.2}
\]

where

\[
h(T, C) = \max_{t \in T} \min_{c \in C} ||t - c|| \tag{5.3}
\]

and

\[
h(C, T) = \max_{c \in C} \min_{t \in T} ||c - t|| \tag{5.4}
\]

\( h(T, C) \) measures the maximum distance of the template’s border pixels to the nearest pixel of the candidate’s border pixels. Similarly \( h(C, T) \) measures the maximum distance between the candidate and the template’s border pixels. Finally \( H(T, C) \), the Hausdorff distance is the maximum of the two maxima \( h(T, C) \) and \( h(C, T) \). It is easy to see that if \( H(T, C) = d \), every template pixel must be within a distance less than \( d \) of pixels from the candidate regions in \( C \) and vice versa. In equations 5.3 and 5.4 different distance definitions can be used. The Euclidean distance between two pixels is used, where horizontally or vertically adjacent pixels have unit distance and diagonally adjacent pixels are at a distance equal to \( \sqrt{2} \).

To reduce the sensitivity of the Hausdorff distance to the outer pixels, which can come from wrong segmentation in complex images, the generalised Hausdorff distance is defined [228, 229]. Instead of using the maximum value in equation 5.3 and 5.4, the distances are sorted in ascending order, then the \( k_{th} \) and \( l_{th} \) value are chosen. Therefore equations 5.3 and 5.4 are modified according to the following:

\[
h(T, C) = \min_{t \in T} \min_{c \in C} ||t - c|| \tag{5.5}
\]
and
\[ h(C, T) = l_{th} \min_{c \in C, t \in T} ||c - t|| \] (5.6)

The \( k \) and \( l \) parameters determine how many of template’s pixels are expected to be near to the candidate shape’s border and vice versa. They are selected by the user and a reasonable choice could be \( 0.85m \) and \( 0.85n \) where \( m \) and \( n \) are the number of border pixels in the first and second shapes, respectively. This method reduces the sensitivity to noise while maintaining the response to local changes.

Based on the application, the Hausdorff distance between the shapes is measured and the minimum distance or distances less than a predefined threshold determine the similar shapes in the image database.

Because the proposed “object-of-interest” extraction algorithm tests all region combinations, which could be a large number of combined regions, it is useful to reduce the computational complexity by reducing the number of border pixels. The number of pixels is decreased by gridding technique without significant negative effect. The shape bounding box is divided into a set of grids of fixed or variable sizes. The grids, which are totally located on the inside or outside of the object area, do not include any border pixels. The boundary grids include both shape and background pixels. In every boundary grid the number of pixels is counted, and if it is more than a threshold, such as 50% of the grid area, the grid is replaced with a boundary pixel at its center. Therefore the number of border pixels is reduced significantly depending on the grid’s size while the precision is mostly not affected because of the insensitivity of this method to small distortions. In Figure 5.2 an example of shape gridding is shown.

5.2.4 Single Resolution Search and Computational Complexity Analysis

The search for the “object-of-interest” algorithm includes testing possible region combinations of the segmented image. Any combination of regions can be an acceptable answer. For example consider the segmentation in Figure 5.3 which includes four regions. The possible regions combinations are:
Meaningful Image Object Extraction

Figure 5.2 The shape bounding box is graded to reduce the number of borders pixels: (a) the grids; (b) borders grids are replaced with pixels.

Figure 5.3 The image is segmented into four regions. All regions are connected.

\[ \{R_1\}, \{R_2\}, \{R_3\}, \{R_4\} \]
\[ \{R_1, R_2\}, \{R_1, R_3\}, \{R_1, R_4\}, \{R_2, R_3\}, \{R_2, R_4\}, \{R_3, R_4\} \]
\[ \{R_1, R_2, R_3\}, \{R_1, R_2, R_4\}, \{R_1, R_3, R_4\}, \{R_2, R_3, R_4\} \]
\[ \{R_1, R_2, R_3, R_4\} \]

Each possible region combination, which passes the introductory aspect ratio test, is examined by the region matching algorithm. The similarity value is computed and compared with a user defined threshold then it is accepted or rejected. Based on the scenario, the search continues until the region combination with the most similarity and a distance less than the predefined threshold is found. Alternatively, regions
are analysed until a number of regions with a similarity greater than the threshold are found. The number of region combinations to be analysed in both of these two scenarios could be very high, resulting in high computational complexity.

In the following, the maximum number of candidate region combinations is computed. Generally, if it is supposed that that there are \( N \) fully connected regions, the maximum number of possible combinations is:

\[
\sum_{k=1}^{N} \binom{N}{k} = 2^N - 1
\]

This is a very big number for normal numbers of segmentation regions such as \( N = 50 \). Of course this is the worse case which assumes that all regions are connected together and all the combinations are examined. Practically, neighbouring is a local feature, and the number of possible combinations is much less than \( 2^N - 1 \). However, it is still a very large number. The experimental results in Section 5.4 show that computational complexity is so high that it practically renders the algorithm useless for real applications.

### 5.3 Hierarchical Search

In a regular search over regions of a single resolution segmented image, all image areas and region combinations are analysed with the same priority. Inspired by a well known feature in the human visual system called the “global precedence effect” (forest before trees) where the processing pathway for outline (low frequency) is faster than detail (high frequency), a hierarchical search is proposed. In a simple way, a low resolution image where the outline of the “object-of-interest” is defined is given higher priority in the search process, and if the search fails, higher resolutions are searched until the search process is exhausted. However, the search at low or very low resolutions such as \( 4 \times 4 \) is not accurate or useful. To simplify this problem an irregular pyramid in Section 5.3.2 is proposed. In this section, considering the “global precedence” feature of the human visual system, a pyramidal-based hierarchical template search is proposed which gives different priorities to different region groupings.
5.3.1 Global Precedence Effect

To reduce the computational complexity of the search over the entire image, a feature of the human visual system known as the global precedence effect (GPE) [14, 223, 230] is considered. The saying “forest before trees” summarises this effect. This feature was reported for the first time by Navon in 1979 [223]. He proposed a view of visual processing in which the global percept precedes local analysis. For example in Figure 5.4, the global and large size letter ‘E’ will be seen before the small ‘H’ characters. Based on his experiments, Navon found that there is a perceptual advantage for the larger and more global stimulus as compared with the smaller local stimulus. He proposed that global information is coded first, whereas local information is analysed at the next step of visual perception. This effect is still a topic of research. Different mechanisms to interpret this effect have been proposed [14]. For example, the global precedence may simply reflect the difference in discrimination between global and local shapes of a compound stimulus. Alternatively, it may results from the intrinsic properties of the transient and sustained visual systems that are most sensitive to low and high spatial frequencies and carry global or local information, respectively. Shulman et al. [231] first demonstrated a close relationship between global and local processing and low and high spatial frequency information. They showed that the low spatial frequencies play a key role in mediating information at the global level of a compound stimulus whereas high spatial frequency channels are important in carrying information at the local level.

Correspondingly, it is fair and useful that the main global objects existing in the image be detected first. To implement a similar object detection priority a pyramid-based search is proposed in the next section.

5.3.2 Multiresolution Segmentation and Multi-Level Search

The “global precedence effect can be interpreted ” as a hierarchy in object recognition during human visual perception. Considering similar effects in object extraction, the large size objects will be detected before the small objects, and low frequency information is processed before the high frequency components.
Priority in detecting the global, large sized objects and low spatial frequency information processing can be implemented by a multiresolution search through a wavelet pyramid image decomposition, which produces similar images at different resolutions [214, 232]. The wavelet filter separates the high pass band components from the low pass band signal, and due to down sampling, the small size objects are gradually deleted as resolution goes down through pyramid levels towards the lowest resolution. Therefore a hierarchical template search through this pyramid, starting from low resolution towards higher resolutions, simulates the “global precedence effect”. For the hierarchical template search, segmentation at different levels is needed. This is done by the proposed multiresolution scalable segmentation algorithm. The scalability of the proposed segmentation is a valuable feature at this stage because it maintains the shapes’ patterns at different resolutions. This increases the accuracy and reliability of the search at the lower resolutions. Furthermore, the perfect relationship of parent and children between regions at different resolutions will detect the extracted object at other resolutions.

Template searching through a multiresolution pyramid has different applications in image processing and pattern recognition algorithms [233–235]. Briefly, the idea in a pyramid search is to aim for coarse detection at lower resolution, then the result is tuned by a local search at the higher resolutions. For example, in motion estimation the low resolution search is projected to higher resolutions to be more re-
Figure 5.5 The hierarchical stack or irregular pyramid segmentation corresponding to the pyramid segmentation. The lowest resolution segmentation includes 2 regions and the similar segmentation map at the top level of the irregular pyramid can be seen. Similarly, the other corresponding segmentations at different levels of the regular and irregular segmentation pyramids have similar segmentation maps.

The advantage is taken of all different resolutions of the pyramid decomposition, and the “global precedence effect”, is fully implemented by defining a stack as a complementary data structure. The defined stack is a set of full-size image segmentation maps which correspond to the segmentation at different resolutions of the pyramid. We call this stack the irregular pyramid. The elements of the stack or irregular pyramid are built hierarchically from fine to lowest resolution. At each resolution, the hierarchical segmentation is obtained by considering three different segmentations: 1) the corresponding pyramid segmentation at the same resolution; 2) the pyramid segmentation of the neighbouring finer resolution; and 3) the hierarchical segmenta-
tion of the neighbouring finer resolution. Figure 5.5 shows this relationship.

At the bottom of the stack, the segmentation of the finest resolution is equal to the fine resolution scalable segmentation at the top of the regular pyramid segmentation. On going down through pyramid toward lower resolutions, small objects/regions are deleted, and the number of existing regions decreases. Similarly, these regions should be deleted from the corresponding hierarchical segmentation. The size reduction during the pyramid decomposition deletes the regions physically. However, in the hierarchical segmentation, the size is kept the same and the physical deletion of regions doesn’t occur. Therefore the regions are deleted logically: the deleted regions are merged with the neighbouring regions by a criterion such as most similarity and the existence of salient edges between regions. Actually, the regular pyramid guides the hierarchical segmentation to delete \( m \) regions hierarchically in \( n \) steps, where \( n \) is the number of levels in the pyramid-based decomposition and \( m \gg n \). The regions corresponding to the global shapes are also maintained, as much as possible, during the hierarchical segmentation process by the pyramidal regions deletion rule. Therefore corresponding to the lower resolutions of the regular pyramid, the smaller regions in the hierarchical scale segmentation are (logically) deleted and global regions corresponding to low spatial frequency with larger size objects remain. Finally, at the lowest level of the pyramid there is the hierarchical full size segmentation with only two regions at the top of the stack. Figure 5.6 shows the flow chart for creation of the irregular segmentation pyramid.

The search is started through the hierarchical segmentation patterns at the top of the stack. If the “object-of-interest” is not found at the current resolution, the hierarchical segmentation patterns corresponding to the next higher scale will be popped from the stack, and it will be searched for the “object-of-interest”. The search will continue through higher scale hierarchical segmentation images until the “object-of-interest” is found. The lower resolution region combinations have corresponding regions in the hierarchical segmentation of higher resolutions. Therefore they need not be tested at the higher resolutions. Only newly emerging region combinations at the higher resolutions theoretically need to be tested.

It is clear that the proposed hierarchical object search detects global and large size
Meaningful Image Object Extraction

Figure 5.6 Irregular Pyramid (IRPRMD) creation algorithm. From highest resolution to lowest resolution, different levels of the irregular pyramid are created from the regular pyramid (RPRMD) segmentation.

objects much faster than the regular single resolution search. However, the computational savings for the detection of the local and small size objects is minimal. Nevertheless this priority search for the detection of the “object-of-interest” is more efficient and is consistent with the human visual system. Figure 5.7 shows a flow chart of a multiresolution search for the “object-of-interest” through the pyramid.

5.4 Experimental Results and Discussion

To show the full benefit of the simulation of the “global precedence effect” and advantages of hierarchical object extraction, this section presents the simulation results for some real images including “head and shoulders”, car, etc. The shape matching algorithm described in Section 5.2 is utilised to measure the similarity between the
shape template and the candidate regions. Because each example has many images at different resolutions, they are shown by equal small sizes. The results are discussed and the advantages/disadvantages of the proposed multiresolution segmentation and hierarchical search are illustrated. Since the computational complexity of the algorithm is an exponential function of the number of regions, regions smaller than a predefined threshold are deleted from the pyramid segmentation.

5.4.1 “Head-and-Shoulders”

The human “head-and-shoulders” is one of the most important examples in image/video object extraction and processing. It is often the subject of many applications in image and video databases, coding, video conferencing, etc. In this section several examples of “head-and-shoulders” image object extraction are presented. The examples are ordered from simple to more complex scenes in terms of complexity.
for the object extraction algorithm.

As the first example, a relatively simple image of the first frame of the grey-level Clair CIF sequence is chosen. In many of the video object tracking algorithms, a semi-automatic process such as user intervention and fine tuning is used to detect the “object(s)-of-interest” in the first frame [184, 199, 220, 237]. The proposed object detection algorithm, however, can be used for automatic extraction of the “object(s)-of-interest” from the first frame of image sequences. The original Clair image is shown in Figure 5.8 (a). The initial hierarchical segmentation which is equal to the scalable segmentation at the highest resolution is shown in Figure 5.8 (b). The scalable pyramidal segmentation and its corresponding hierarchical segmentation at each resolution segmentation from $144 \times 176$ resolution to the lowest resolution of $2 \times 2$ are shown in Figure 5.8 (c) to (r).

Table 5.1 shows the number of regions and the region combinations at each resolution. The highest resolution includes 31 regions and 8731 region combinations. Searching over all these region combinations will be computationally very complex. However, through the pyramid decomposition and scalable segmentations algorithm, the “object-of-interest” is accurately separated by the hierarchical segmentation algorithm at the $3 \times 3$ resolution. The scalable segmentation and its corresponding hierarchical segmentation at the $3 \times 3$ resolution are shown in Figure 5.8 (o) and (p). The hierarchical segmentation includes only four regions. If the lower $2 \times 2$ resolution which includes 7 region combinations and the $3 \times 3$ resolution which includes 12 region combinations regions are considered, the maximum\(^1\) number of tested region combinations is $12 + 7 = 19$ regions. Therefore there is at least a $(1 - 19/8731) \times 100 = 99.78\%$ reduction in the number of region combinations and consequently, the computational complexity.

The template or model of the “object-of-interest” (“head-and-shoulders”) and the extracted object are shown in Figure 5.9 (a) and (b). The matching of the template and the extracted objects are shown in Figure 5.9 (d) and (e). The Hausdorff distance of the template matching is 5.6, which is less than the predefined threshold. As an

\(^1\)Because the search stops when the object is found and all the region combinations are not necessarily tested.
Table 5.1 Number of regions and region combinations at different resolutions of Clair image Segmentation.

<table>
<thead>
<tr>
<th>Resolution</th>
<th>288×352</th>
<th>144×176</th>
<th>72×88</th>
<th>36×44</th>
<th>18×22</th>
<th>9×11</th>
<th>5×6</th>
<th>3×3</th>
<th>2×2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Regions</td>
<td>31</td>
<td>25</td>
<td>23</td>
<td>21</td>
<td>17</td>
<td>13</td>
<td>9</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Number of Combinations</td>
<td>8731</td>
<td>4312</td>
<td>2680</td>
<td>1681</td>
<td>491</td>
<td>192</td>
<td>55</td>
<td>12</td>
<td>3</td>
</tr>
</tbody>
</table>

example of a rejected match, Figure 5.9 (f) to (k) shows another candidate region and its match against the template. Figure 5.9 (j) and (k) shows that the rotated template to some extent fits the candidate regions. The Hausdorff distance between this region’s shape and the template is 11.06. Therefore this candidate’s rejection needs a relatively finely tuned threshold which could be entered by the user.

As the second example, the “head-and-shoulders” is extracted from the first frame of the Foreman image sequence. The image is CIF size with a YUV colour format where Y is in full resolution and U and V are in half resolution. This image object extraction has more computational complexity than the Clair image, because the background is more cluttered and the object and background contrast is lower than in the Clair image. Therefore it is segmented into more regions than the Clair image segmentation, which increases the computational complexity. The original image is shown in Figure 5.10 (a). The decomposed pyramid images are segmented by the proposed scalable segmentation. The scalable segmentation and hierarchical segmentation are also shown in Figure 5.10 (b) to (r). 9 × 11 is the lowest resolution in which the “object-of-interest” is effectively separated from the image background area. Therefore the “object-of-interest” is searched from low to high resolutions at 2 × 2, 3 × 3, 5 × 6, and 9 × 11 resolutions respectively. The maximum number of candidate regions will be:

$$\sum_{k=1}^{21} \binom{21}{k} + \sum_{k=1}^{16} \binom{16}{k} + \sum_{k=1}^{7} \binom{7}{k} + \sum_{k=1}^{3} \binom{3}{k} = 2097151+65535+127+7 = 1114244$$

(5.8)

The number of regions and the region combinations are shown in Table 5.2. The real number of region combinations in the four lowest resolutions, including 2 × 2,
Figure 5.8 Clair original image, its scalable segmentation (SSeg) and hierarchical segmentation (HSeg) at different resolutions. The hierarchical segmentation images are just after the scalable segmentation at each resolution: (a) the original Clair image at 288 × 352 resolution; (b) 288 × 352 SSeg; (c) 144 × 176 SSeg; (d) HSeg corresponding to 144 × 176; (e) 72 × 88 SSeg; (f) HSeg corresponding to 72 × 88; (g) 36 × 44 SSeg; (h) HSeg corresponding to 36 × 44; (i) 18 × 22 SSeg; (j) HSeg corresponding to 18 × 22; (k) 9 × 11 SSeg; (l) HSeg corresponding to 9 × 11; (m) 5 × 6 SSeg; (n) HSeg corresponding to 5 × 6; (o) 3 × 3 SSeg; (P) HSeg corresponding to 3 × 3; (q) 2 × 2 SSeg; (r) HSeg corresponding to 2 × 2.
Figure 5.9 (a) Clair “head-and-shoulders” template; (b) texture of the extracted object; (c) the extracted object’s shape; (d) match between the template and the region, where the candidate region is drawn over the template; (e) template is over candidate region; (f) texture of the second candidate region; (g) candidate region’s shape; (h) match between the template and the candidate region, where the candidate region is drawn over the template; (i) template is over region; (j) match between the rotated template and the candidate region, where the candidate region is drawn over the template; (k) rotated template is over regions.
Table 5.2 Number of regions and region combinations at different resolutions of Foreman image segmentation.

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Number of Regions</th>
<th>Number of Combinations</th>
</tr>
</thead>
<tbody>
<tr>
<td>288 × 352</td>
<td>89</td>
<td>8.37 × 10^7</td>
</tr>
<tr>
<td>144 × 176</td>
<td>85</td>
<td>2.01 × 10^7</td>
</tr>
<tr>
<td>72 × 88</td>
<td>81</td>
<td>9.5 × 10^6</td>
</tr>
<tr>
<td>36 × 44</td>
<td>69</td>
<td>5.94 × 10^5</td>
</tr>
<tr>
<td>18 × 22</td>
<td>61</td>
<td>174749</td>
</tr>
<tr>
<td>9 × 11</td>
<td>61</td>
<td>3298</td>
</tr>
<tr>
<td>5 × 6</td>
<td>16</td>
<td>732</td>
</tr>
<tr>
<td>3 × 3</td>
<td>7</td>
<td>43</td>
</tr>
<tr>
<td>2 × 2</td>
<td>3</td>
<td>7</td>
</tr>
</tbody>
</table>

3 × 3, 5 × 6 and 9 × 11 is equal to 3298 + 732 + 43 + 7 = 4090. Therefore the real number of searches has to cover than 4090 regions and is greatly less than 1114244 regions. As Table 5.2 indicates, moving from the resolution 18 × 22 toward higher resolutions, the number of region combinations increases so much that practically it is impossible to search for the “object-of-interest” over these resolutions. In particular, at the highest resolution the number of region combinations is so high that the search is practically impossible. The efficiency of the pyramidal template search compared to the single resolution template search is \((1 - 4090/8.37 \times 10^7) \approx 99.99\%\) which is close to 100\%. Regular single resolution produces more regions than regular multiresolution segmentation and the proposed scalable pyramid segmentation algorithms [238]. This increase in the number of regions increases the computational complexity of the search algorithm.

The extracted “object-of-interest”, its template and the regions matching with the template are shown in Figures 5.11 (a) to (e). The Hausdorff distance between the object’s template and the extracted object is 7.4. As an example of a rejected region, a region and its match with the template model is also shown in Figures 5.11 (f) to (h). The Hausdorff distance of this tested object and template is 30.65.

As the last example of “head-and-shoulders”, an ordinary image of an office is considered. It again includes a “head-and-shoulders” and some other meaningful objects such as a computer monitor, case, keyboard, speaker, books. The size of “object-of-interest” is not big or dominant. Moreover its contrast in some parts of its area, compared to the background, is not enough to separate it easily by a regular segmentation. As a result, to separate the “head-and-shoulder” the number of regions
Figure 5.10 Foreman original image, its scalable segmentation (SSeg) and hierarchical segmentation (HSeg) at different resolutions. The hierarchical segmentation images are just after the scalable segmentation at each resolution: (a) the original image at $288 \times 352$ resolution; (b) $288 \times 352$ SSeg; (c) $144 \times 176$ SSeg; (d) HSeg corresponding to $144 \times 176$; (e) $72 \times 88$ SSeg; (f) HSeg corresponding to $72 \times 88$; (g) $36 \times 44$ SSeg; (h) HSeg corresponding to $36 \times 44$; (i) $18 \times 22$ SSeg; (j) HSeg corresponding to $18 \times 22$; (k) $9 \times 11$ SSeg; (l) HSeg corresponding to $9 \times 11$; (m) $5 \times 6$ SSeg; (n) HSeg corresponding to $5 \times 6$; (o) $3 \times 3$ SSeg; (p) HSeg corresponding to $3 \times 3$; (q) $2 \times 2$ SSeg; (r) HSeg corresponding to $2 \times 2$. 
Figure 5.11 (a) The extracted Foreman “head-and-shoulders” shape; (b) The extracted Foreman "head-and-shoulders" texture; (c) template; (d) match between the template and the region, where the candidate region is drawn over the template; (e) template is over candidate region; (f) a (rejected) candidate region; (g) match between the template and the (rejected) candidate region, where the candidate region is drawn over the template; (h) template is over the (rejected) region.
increases, and therefore the “object-of-interest” extraction for this example is computationally more complex than the last two examples.

The original image is shown in Figure 5.12(a). It is CIF size in YUV colour format. The image is segmented at different resolutions of the pyramid decomposition by the proposed scalable segmentation. In Figure 5.12 (b) to (r) the scalable segmentation and its corresponding hierarchical segmentation at different resolutions are shown. The number of region combinations at each resolution is shown in Table 5.3. There is a considerably higher number of possible region combinations for this image compared to the last two examples.

For this example, the “object-of-interest”, the ”head-and-shoulders”, is separated in the hierarchical segmentation corresponding to the $64 \times 64$ resolution. However, if a small distortion, which is a small part of the background added to the object area around the elbow, is acceptable then the “object-of-interest” can be extracted in the lower $32 \times 32$ resolution with less computational complexity. This object is called a low quality shape from now on.

The numbers of regions at different resolutions are shown in Table 5.3. The low quality object is detected at $32 \times 32$ resolution, while the best quality object is detected at $64 \times 64$ resolution. The number of tested candidate regions is about $\approx 5.14 \times 10^8$ for the search at $32 \times 32$ resolution, which detects the “object-of-interest” with a little distortion, and $\approx 6.35 \times 10^{10}$ for the search at the $64 \times 64$ resolution to detect the object without distortion.

Analysing this number of candidate regions is practically very time consuming. However, the computational complexity can be reduced by a heuristic algorithm. The face region is detected using a face detection algorithm. There are many such face detection algorithms in the literature [15, 148, 239]. The basic idea is to detect the skin regions and then delete the non-face skin regions using geometrical constraints such as considering an elliptical shape of the face. Finally the face region is refined. For example there are small non-skin regions inside the skin area corresponding to face components such eyebrows, eyes, lips. These regions are merged with the skin area to create the final face region. In this application, pixel-wise bor-
der accuracy of the detected face region is not necessary. The face regions and the face bounding box are seen in Figure 5.13 (a) and (b). Considering the face region’s size, the “head-and-shoulders” search bounding box can be estimated. The bounding box is a rectangular area that includes the “object-of-interest”. For example, the width of the bounding box is considered to be 3 times the face width. The length of the area over the face is less than half of the face length, and the area’s length under the face is 4 times the face length. These values are conservative, ensuring that that the “head-and-shoulders” will be inside the predetermined search bounding box. If the estimated search bounding box falls outside the image boundary, it will be adjusted to fall within the image boundary. In this example, the face bounding box is a $76 \times 50$ window and the “head-and-shoulders” search bounding box is a $190 \times 145$ window which is about 44% of the original $256 \times 256$ image at the finest resolution. The separated rectangular processed image area, can be seen in Figure 5.13 (c). The “head-and-shoulders” bounding box estimation is explained in Figure 5.13 (d). The regions which are partially or totally out of the bounding box are not processed, therefore the number of regions is significantly reduced. Table 5.4 shows the number of regions and region combinations for the simplified “office” image.

The Hausdorff distance of the extracted shape at the $64 \times 64$ resolution, shown in Figure 5.14(a), compared to the template is equal to 6.7 while the Hausdorff distance of the low quality extracted shape shown in Figure 5.14(g) is equal to 8.61. The matches of the extracted shapes with the template are shown in Figure 5.14 (d), (e), (g), and (f). Fine tuning the threshold to a value that passes the best match and rejects the other candidate matches needs a very well tuned threshold. This scenario can happen with real images. Fine tuning the threshold to a value that passes the best match and rejects the other candidates is critical for the success of the algorithm. This is particularly true in complex and cluttered images. The algorithm will be less sensitive to fine tuning if the following procedure is followed: at any resolution only the best match that passes the threshold is accepted. This separates the best object at a given resolution.

This example confirms that in complex images, the computational complexity remains too high. Furthermore, the object and the chair are not well separated at the
Table 5.3 Number of regions and region combinations at different resolutions of Fardin image segmentation.

<table>
<thead>
<tr>
<th>Resolution</th>
<th>256×256</th>
<th>128×128</th>
<th>64×64</th>
<th>32×32</th>
<th>16×16</th>
<th>8×8</th>
<th>4×4</th>
<th>2×2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Regions</td>
<td>128</td>
<td>116</td>
<td>98</td>
<td>82</td>
<td>61</td>
<td>33</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>Number of Combinations</td>
<td>4.96×10¹³</td>
<td>1.64×10¹²</td>
<td>6.35×10¹⁰</td>
<td>5.14×10⁸</td>
<td>8.13×10⁶</td>
<td>2.13×10⁵</td>
<td>43</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 5.4 Number of regions and region combinations at different resolutions of simplified segmentation of Fardin image.

<table>
<thead>
<tr>
<th>Resolution</th>
<th>256×256</th>
<th>128×128</th>
<th>64×64</th>
<th>32×32</th>
<th>16×16</th>
<th>8×8</th>
<th>4×4</th>
<th>2×2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Regions</td>
<td>69</td>
<td>58</td>
<td>46</td>
<td>37</td>
<td>24</td>
<td>15</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>Number of Combinations</td>
<td>6.84×10⁹</td>
<td>8.73×10⁷</td>
<td>4.15×10⁶</td>
<td>3.36×10⁵</td>
<td>84536</td>
<td>1659</td>
<td>15</td>
<td>3</td>
</tr>
</tbody>
</table>

segmentation stage (since small parts of the chair are added to the “object-of-interest” area). This segmentation problem is due to low colour contrast between background (chair) and the object (“head-and-shoulder”) in these areas. A feedback from the extraction/recognition to the segmentation stage can help to correct the segmentation output.

5.4.2 Template Design

In the last three examples the same class of objects, “head-and-shoulders”, was searched. However, due to different shapes for the different examples, different templates for the “object-of-interest” search have been used. Clearly, it is more desirable if a general template can be used in a search of all the “head and shoulders” extraction examples. This problem to some extent can be solved by a dynamic or deformable template mechanism for some specific “object-of-interest” such as “head-and-shoulders”. The deformable template is flexible and changes to adapt to the shape of the “object-of-interest”. A search algorithm over the template model space should be considered but it significantly increases the computational complexity. To reduce the computational complexity different heuristics can be used for each
Figure 5.12 Office original image with its scalable segmentation (SSeg) and hierarchical segmentation (HSeg) at different resolutions. The HSeg images are just after the SSeg at any resolution; (a) The original image at $256 \times 256$ resolution; (b) $256 \times 256$ SSeg; (c) $128 \times 128$ SSeg; (d) HSeg corresponding to $128 \times 128$; (e) $64 \times 64$ SSeg; (f) HSeg corresponding to $64 \times 64$; (g) $32 \times 32$ SSeg; (h) HSeg corresponding to $32 \times 32$; (i) $16 \times 16$ SSeg; (j) HSeg corresponding to $16 \times 16$; (k) $8 \times 8$ SSeg; (l) HSeg corresponding to $8 \times 8$; (m) $4 \times 4$ SSeg; (n) HSeg corresponding to $4 \times 4$; (o) $2 \times 2$ SSeg; (p) HSeg corresponding to $2 \times 2$. 
Figure 5.13 (a) Office original image where the face area is extracted; (b) the bounding box which includes the “head-and-shoulders”; (c) The processed image area which includes the "head-and-shoulders"; (d) The bounding box for the "head-and-shoulders" is drawn where the ratio between the face and the bounding box determines the width and length.

object class. For example, if the differences between the “head-and-shoulders” images are examined, the main difference often involves the length of the body that is inside the image area. This can be solved by defining a flexible template in which body part length is adaptable to fit with the “object-of-interest” regardless of the object’s body length. To reduce the computational complexity, the search space of the template length is divided into four intervals and each interval is shown by a static template. Therefore, there are four templates modes corresponding to a very short, short, medium and long body in the image. This is equal to quantizing the template search space. Therefore, for each candidate region four template modes are tested, which is equal to a four times increase in the computational complexity. The aspect ratio test will decrease the computational complexity significantly because the aspect ratio values of the four different templates modes are computed only once. For each
region combination, the shape’s aspect ratio is compared, and many of the shapes are rejected due to the simple aspect ratio comparison test. In Figure 5.15 (a), (b), (c) and (d) four defined general templates for the “head-and-shoulders” applications are shown. These four templates can also be interpreted as a library of “head-and-shoulders” templates. Actually a library of templates is created which shows different views of the “object-of-interest”. Region combinations are compared with the templates in this library.

The Clair, Foreman and Office examples are again examined. Deformable templates
Table 5.5 Hausdorff distance of objects from "head-and-shoulders" templates.

<table>
<thead>
<tr>
<th>Templates</th>
<th>Special</th>
<th>Very Short</th>
<th>Short</th>
<th>Medium</th>
<th>Long</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clair</td>
<td>3.8</td>
<td>12.5</td>
<td>7.4</td>
<td>7.2</td>
<td>9.8</td>
</tr>
<tr>
<td>Foreman</td>
<td>9.8</td>
<td>11.3</td>
<td>18.2</td>
<td>18.1</td>
<td>17.7</td>
</tr>
<tr>
<td>Fardin</td>
<td>7.8</td>
<td>15</td>
<td>14.9</td>
<td>9.2</td>
<td>6.1</td>
</tr>
</tbody>
</table>

for the “head-and-shoulders” object which include the four modes are used. The templates and results are shown in Figures 5.15 and 5.16. The match between the extracted object and the library of templates are shown in Figure 5.15 (e) to (t) and 5.16 (a) to (h). For the “object-of-interest”, one of the four different templates is a better fit to the object. The Hausdorff distances of the matches are shown in Table 5.5.

In respect to the general templates, the Hausdorff distances are increased compared to the case when a special template model is used. Therefore bigger thresholds values are used. The Foreman shape is detected with a bigger Hausdorff distance, which is a result of the Foreman’s hat. In the proposed template shape a hat is not considered, which increases the distance between the extracted Foreman shape and the template.

The template design depends on the application. For rigid object searches, a library of templates is useful, but in the case of a non-rigid object search such as for a human that can have very different shapes, a dynamic and deformable model that analyses and supports changes of shape is needed. In the next section another example of a dynamic template is explored. However, effective dynamic template design needs further research which is beyond the scope of this thesis.

### 5.4.3 Natural Objects Extraction Examples

In this section object extraction for different objects such as a car, ball and tools is explored. In the first example, a car as the “object-of-interest” is searched in the scene shown in Figure 5.17(a). The size of the image is $256 \times 256$ in YUV colour format. Its scalable and hierarchical segmentation are shown in Figure 5.17 (b) to (p). The template is shown in Figure 5.18 (a). The numbers of regions at different resolutions are shown in Table 5.6. For the three highest resolutions of $256 \times 256$, $128 \times 128$ and $64 \times 64$ the numbers of region combinations are so high that practically...
Figure 5.15 (a) The very short “head-and-shoulders” template; (b) short template; (c) medium template; (d) long template; (e) Clair object match, shown over very Short template; (f) Clair object match shown under very Short template; (g) over short template; (h) under short template; (i) over medium template; (j) under medium template; (k) over long template; (l) under long template; (m) Foreman object match shown over very Short template; (n) under very Short template; (o) over short template; (p) under short template; (q) over medium template; (r) under medium template; (s) over long template; (t) under long template.
it is equal to infinity\(^2\). Therefore in Table 5.6 their number of region combinations are shown with the infinity symbol, \(\infty\). The “object-of-interest” is recognisable at the 8th level of hierarchical segmentation which corresponds to \(2 \times 2\) resolution. The maximum number of candidate tests to find the “object-of-interest” is 12. The extracted objects with their match images are shown in Figure 5.18 (b) to (e). The Hausdorff distance of the extracted object compared to the template is 7.3.

Detection of the main object in the image is influenced by the “global precedence effect”. However, if there are some small objects attached to the main object, depending on the contrast of the object and its background, their detection can be done at the next levels of the hierarchy. For example, if detecting the car with bumper and wheels is important, they are detected at the \(16 \times 16\) resolution with much more complexity. The number of region combinations at the \(16 \times 16\) resolution and its lower resolutions is equal to \(12 + 43 + 732 + 212980 = 213767\). The regions, including the bumper (not wheels), its texture and its matched figures, are shown in Figure 5.18 (f) to (i). Its Hausdorff distance to the template is 6.2. The region including bumper and

\(^2\)At these resolutions, testing the existence of the region groups, to count their number, needs very powerful computers with huge memories.
Table 5.6 Number of regions and region combinations at different resolutions of Car image Segmentation.

<table>
<thead>
<tr>
<th>Resolution</th>
<th>256×256</th>
<th>128×128</th>
<th>64×64</th>
<th>32×32</th>
<th>16×16</th>
<th>8×8</th>
<th>4×4</th>
<th>2×2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Regions</td>
<td>276</td>
<td>229</td>
<td>165</td>
<td>107</td>
<td>53</td>
<td>20</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>Number of Combinations</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>8.61×10^{12}</td>
<td>432746</td>
<td>732</td>
<td>43</td>
<td>12</td>
</tr>
</tbody>
</table>

wheels is detected at 16 × 16 resolution, and its match with the template is shown in Figure 5.18 (j) to (m). The Hausdorff distance from the template is 10.1 which is relatively high. Detection of the car with its wheels needs a template with more similarity to car shapes, such as the one shown in Figure 5.19 (n), which includes wheels. The best region detected by this template and its corresponding match are shown in Figure 5.19 (o), (p) and (q). The Hausdorff distance is reduced to 6.4, which is an acceptable distance for recognising a candidate region.

In this example the car is the main object in the image. Therefore, its simple body is detected very easily at low resolution. The wheels, bumper and car lights are the details attached to the object. Due to their colour/contrast situation, they are mixed with the background at low resolutions. Therefore their detection will be done at higher resolutions with increased complexity. This example shows that in real images, emphasising the detection of details or small objects attached to the main object with low contrast with the background will increase the computational complexity significantly. However, in many applications their detection is not necessary. Therefore, depending on the application, a decision on the level of details should be made. The decision has implications for template design and threshold values.

In the next example the detection of a small size object is considered. The original image is seen in Figure 5.20 (a). The grey-level image is in SIF size and the “object-of-interest” is the ball, which is a small size object. The image is decomposed to 10 different scales by the wavelet decomposition. The image pyramid is then segmented by the scalable segmentation. The scalable and its corresponding hierarchical image segmentation at the different resolutions can be seen in Figure 5.20 (b) to (p). Due
Figure 5.17 Car original image with its scalable segmentation (SSeg) and hierarchical segmentation (HSeg) at different resolutions. The HSeg images are just after the SSeg at each resolution: (a) The original image at $256 \times 256$ resolution; (b) $256 \times 256$ SSeg; (c) $128 \times 128$ SSeg; (d) HSeg corresponding to $128 \times 128$; (e) $64 \times 64$ SSeg; (f) HSeg corresponding to $64 \times 64$; (g) $32 \times 32$ SSeg; (h) HSeg corresponding to $32 \times 32$; (i) $16 \times 16$ SSeg; (j) HSeg corresponding to $16 \times 16$; (k) $8 \times 8$ SSeg; (l) HSeg corresponding to $8 \times 8$; (m) $4 \times 4$ SSeg; (n) HSeg corresponding to $4 \times 4$; (o) $2 \times 2$ SSeg; (p) HSeg corresponding to $2 \times 2$. 
Figure 5.18 (a) The Car template (b) The extracted car shape at $2 \times 2$ resolution; (c) the extracted car shown with its texture; (d) match between the template and the object, where the candidate region is drawn over the template; (e) template is over candidate region; (f) the extracted car shape at $16 \times 16$ resolution; (g) the extracted car shown with its texture; (h) match between the candidate rough region and the template, where the candidate region is drawn over the template; (i) template is over candidate region; (j) a candidate region at $16 \times 16$ resolution; (k) the candidate region shown with its texture; (l) match between the template and the object, where the candidate region is drawn over the template; (m) template is over candidate region;

to the small size of the “object-of-interest”, it is not detected before the $5th$ level of pyramid decomposition. Therefore the resolutions $1 \times 2, 2 \times 3, 4 \times 6, 8 \times 11$
Figure 5.19 (n) a different car template includes the car’s wheels; (o) the object extracted by the new template at $16 \times 16$ resolution; (p) match between the template and object, where the candidate region is drawn over template; (q) template is over candidate region.

Table 5.7 Number of regions and region combinations at different resolution of the Table_Tennis image segmentation.

<table>
<thead>
<tr>
<th>Resolution</th>
<th>240×352</th>
<th>120×176</th>
<th>60×88</th>
<th>30×44</th>
<th>15×22</th>
<th>8×11</th>
<th>4×6</th>
<th>2×3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Regions</td>
<td>79</td>
<td>62</td>
<td>45</td>
<td>27</td>
<td>18</td>
<td>11</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Number of Combinations</td>
<td>$8.93\times10^8$</td>
<td>$1.83\times10^6$</td>
<td>$2.12\times10^5$</td>
<td>7019</td>
<td>1058</td>
<td>77</td>
<td>16</td>
<td>3</td>
</tr>
</tbody>
</table>

are searched, and finally the “object-of-interest” is found at the $15 \times 22$ resolution. This hierarchal search, from global to local information, is quite consistent with the “global precedence effect”. The template, the found region and their match are shown in Figure 5.21 (a) to (d). The Hausdorff distance of the match is 4.62. Table 5.7 shows the number of regions and their combinations. $3 + 15 + 78 = 96$ region combinations are searched at the three resolutions lower than $15 \times 22$ and the 1058 combinations at this resolution which the object is found. Therefore in total $96 + 1058 = 1152$ region candidates are searched to find the “object-of-interest”. From this number $96/1152 \times 100 = 8.3\%$ of regions are searched at lower resolutions.

This example shows that the detection of small size objects is done at higher resolutions of the pyramid with more complexity than the large size objects at lower resolutions. But this is an acceptable property consistent with the “global precedence effect” of the human visual system.
Figure 5.20 Table Tennis original image with its scalable segmentation (SSeg) and hierarchical segmentation (HSeg) at different resolutions. The HSeg images are just after the SSeg at each resolution: (a) the original image at $240 \times 352$ resolution; (b) $240 \times 352$ SSeg; (c) $120 \times 176$ SSeg; (d) HSeg corresponding to $120 \times 176$; (e) $60 \times 88$ SSeg; (f) HSeg corresponding to $60 \times 88$ SSeg; (g) $30 \times 44$ SSeg; (h) HSeg corresponding to $30 \times 44$; (i) $15 \times 22$ SSeg; (j) HSeg corresponding to $15 \times 22$; (k) $8 \times 11$ SSeg; (l) HSeg corresponding to $8 \times 11$; (m) $4 \times 6$ SSeg; (n) HSeg corresponding to $4 \times 6$; (o) $2 \times 3$ SSeg; (p) HSeg corresponding to $2 \times 3$.

In the last example, three different objects of two classes in an image are searched. The original image is shown in Figure 5.22 (a). It is a $256 \times 256$ grey image. The “objects-of-interest” are a spanner and a couple of short and normal length screwdrivers in the image. These objects are found at different resolutions. Each class needs its specific template. The screwdriver’s shape has two parts, including the
The ball template; (b) The extracted ball shape at $15 \times 22$ resolution; (c) match between the template and the extracted ball, where the candidate region is drawn over the template; (e) template is over candidate region.

The handle and the metal body, which has a narrow and long length shape and is distorted/absent at the lower resolutions of the pyramids. The scalable segmentation of the image pyramid and its corresponding hierarchical segmentation are shown in Figure 5.22 (b) to (p). The spanner and short screwdriver are detected very easily at the hierarchical segmentation corresponding to the $4 \times 4$ resolution, but the normal screwdriver detection is only possible at the $8 \times 8$ resolution. The numbers of regions and region combinations at different resolutions are shown at Table 5.8.

The spanner template is shown in Figure 5.24(a), but the screwdriver template is more complex. The length of the metal body part is different for different screwdrivers. Therefore a deformable template model is necessary. The metal body part of the template should be flexible to fit to the original objects as shown in Figure 5.23. Therefore a variable range space for the tip length such as $[L/2, +2L]$, where $L$ is the handle length, is considered. The length of the flexible body part is matched to the region by a trial and error algorithm. The steps of the template change do not need to be very fine, and the step size is selected practically as $1/5th$ of the search interval length. This is similar to 5 different constant templates that should be tested. Most of these templates are rejected due to the introductory aspect ratio test without much increase in computational complexity. The flexible template can be seen in Figure 5.23. The three short, normal and long templates are shown in Figures 5.25(b), (c) and (d). The extracted spanner and the two normal and short screwdriver objects and their matches with the templates are shown in Figures 5.24 (a) to (d), 5.25 (a) to (o) and 5.26 (p) to (s). Table 5.9 shows the Hausdorff distance of
Table 5.8 Number of regions and region combinations at different resolutions of the Tools image segmentation.

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Number of Regions</th>
<th>Number of Combinations</th>
</tr>
</thead>
<tbody>
<tr>
<td>256×256</td>
<td>115</td>
<td>5.43×10^{11}</td>
</tr>
<tr>
<td>128×128</td>
<td>98</td>
<td>6.29×10^{9}</td>
</tr>
<tr>
<td>64×64</td>
<td>72</td>
<td>2.40×10^{8}</td>
</tr>
<tr>
<td>32×32</td>
<td>44</td>
<td>8.03×10^{5}</td>
</tr>
<tr>
<td>16×16</td>
<td>24</td>
<td>7.676</td>
</tr>
<tr>
<td>8×8</td>
<td>14</td>
<td>7.06</td>
</tr>
<tr>
<td>4×4</td>
<td>7</td>
<td>5.3</td>
</tr>
<tr>
<td>2×2</td>
<td>4</td>
<td>1.2</td>
</tr>
</tbody>
</table>

Table 5.9 Hausdorff Distances of Tools from templates.

<table>
<thead>
<tr>
<th>Object</th>
<th>Template</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Very Short</td>
</tr>
<tr>
<td>Screwdriver normal</td>
<td>10.81</td>
</tr>
<tr>
<td>Screwdriver short</td>
<td>7.14</td>
</tr>
<tr>
<td>Spanner</td>
<td>–</td>
</tr>
</tbody>
</table>

the matches between different templates and the found objects. The detected normal screwdriver has a little distortion in the metal body part. This comes from the downsampling shape distortion effect. The undistorted object can be detected at 64 × 64 resolution with much more computational complexity. Due to the small distortion it can be passed over, but the other problem here is that rejecting the distorted object and passing the undistorted object needs a finely tuned threshold, which could be the subject of further work. In this example, the “objects-of-interest” are found at different resolutions and depend on the high level knowledge that there are just two screwdrivers that should be detected. Also, already the same assumption that just one spanner is present has been used. This high level knowledge can be used to select the templates and also stop the search when the “objects-of-interest” are found.
Figure 5.22 Tools original image with its scalable segmentation (SSeg) and hierarchical segmentation (HSeg) at different resolutions. The HSeg images are just after the SSeg at each resolution: (a) The original image at 256 × 256 resolution; (b) 256 × 256 SSeg; (c) 128 × 128 SSeg; (d) HSeg corresponding to 128 × 128; (e) 64 × 64 SSeg; (f) HSeg corresponding to 64 × 64 SSeg; (g) 32 × 32 SSeg; (h) HSeg corresponding to 32 × 32; (i) 16 × 16 SSeg; (j) HSeg corresponding to 16 × 16; (k) 8 × 8 SSeg; (l) HSeg corresponding to 8 × 8; (m) 4 × 4 SSeg; (n) HSeg corresponding to 4 × 4; (o) 2 × 2 SSeg; (p) HSeg corresponding to 2 × 2.

5.5 Conclusion

In this chapter a novel hierarchical image object extraction and recognition algorithm was proposed. Simulating the “global precedence effect” of the human visual system
Figure 5.23 The flexible screwdriver template.

Figure 5.24 (a) The spanner template; (b) extracted spanner shape; (c) match between template and the extracted spanner, where the candidate region is drawn over the template; (d) template is over candidate spanner region.

results in a hierarchy of objects and significantly decreases the number of tested candidate regions and the computational complexity. The proposed hierarchical segmentation patterns organised in a irregular pyramid allow us to detect the global objects
Meaningful Image Object Extraction

first and local or small size objects later. This reduces the computational complexity for detecting the main objects of the image. There are many suitable shape matching algorithms. However because the number of regions tested is incredibly high, a matching algorithm with lower computational complexity is preferred. A region-based shape matching algorithm is used, which after affine variation compensation, measures the Hausdorff distance. The proposed algorithm classifies the extracted object into a known class of objects. The proposed recognition needs a template of the “object-of-interest”. The template, which is classed as high level knowledge, can be selected from a library of templates by the application user. Rigid or complex objects can have many similar shapes, and for their detection a flexible or deformable template model is necessary. Some parts of the deformable template are flexible and can be adapted, as much as possible, to the shape of the “object-of-interest”, existing in the image. Deformable templates require searches in the template space and significantly increase the computational complexity. Therefore, more effective and low computational complexity shape matching and recognition that can recognise rigid and non-rigid objects needs further research. Of course many of the candidate regions can be rejected by an introductory aspect ratio test.

The main global object of the image is well detected at low resolution, and small objects are detected at higher resolutions with more computational complexity. Therefore, if the “object-of-interest” includes different parts, some small parts may not be well detected at low resolution. This is because, depending on the contrast and grey/colour similarity between the object and background, part of the deleted small object can be undesirably mixed with the background or desirably with the foreground. Therefore, their detection and processing might be put off to the higher resolutions, which increases the computational complexity. Small distortion in the detection of the “object-of-interest” might be removed at higher resolutions which increases the computational complexity. Here, depending on the thresholds, the detected objects at low resolution can be accepted or rejected, and the search continues at the higher resolutions. The suitable threshold for decisions about accepting or rejecting a region as the “object-of-interest” is tuned by the user, and its automatic setting needs further research. While most of the “object-of-interest” algorithms uniformly search through the image or use an application dependent heuristic such as
first finding the human face, but the proposed algorithm has defined a natural pri-
ority for the information examined. The proposed algorithm can be useful in many
different applications.

Establishing a feedback mechanism from the recognition stage to the low level seg-
mentation is also a challenge which needs further research. The strength of the al-
gorithm is its ability to extract normal objects from real and natural images, which
makes it useful for real scenario applications. Even so, in real images where the
“object-of-interest” is not the dominant object in the image, the detection can still
requires high computational complexity. However, it is consistent with the HVS. In
these cases an application dependent heuristic can decrease the computational com-
plexity. Finally, the proposed algorithm is a big step towards object extraction from
real images.
Figure 5.25 (a) The screwdriver long template; (b) normal template; (c) short template; (d) extracted normal screwdriver shape; (e) extracted normal screwdriver texture; (f) match between the long template and the extracted normal screwdriver, where the candidate region is drawn over the template; (g) normal template is over candidate region; (h) match between the normal template and the extracted normal screwdriver, where the candidate region is drawn over the template; (i) normal template is over candidate region; (j) match between the short template and the extracted normal screwdriver, where the candidate region is drawn over the template; (k) short template is over candidate region; (l) extracted short screwdriver shape; (m) extracted short screwdriver texture; (n) match between the short template and the extracted short screwdriver, where the candidate region is drawn over the template; (o) short template is over candidate region.
Figure 5.26 (p) match between the normal template and the extracted short screwdriver, here the candidate region is drawn over the template; (q) normal template is over candidate region; (r) match between the short template and the extracted short screwdriver, where the candidate region is drawn over the template; (s) short template is over candidate.
Chapter 6

Multiresolution Scalable Video Object Extraction

6.1 Introduction

In this chapter, a method for object-based video segmentation is proposed. The ideas of scalable and visually pleasing segmentation and object extraction are extended to video signals. The video object planes (VOPs) are extracted at different resolutions with scalability and smoothness as constraints. The extracted objects are useful for generic object-based applications and especially for scalable object-based coding applications.

Video signals are treated as sequences of still images. Therefore the computational complexity is a critical problem for video segmentation. A way to deal with this challenge is to promote the video processing from pixel to region-based. Therefore in this chapter a novel region-based video segmentation algorithm is proposed which partitions video frames into foreground and background regions. The proposed multiresolution video segmentation algorithm tracks the objects detected in previous frames, while newly appearing moving objects/regions are also extracted. For clarity, the algorithm is explained for single resolution first, and then it is developed for scalable multiresolution segmentation. First, the frame is partitioned into different regions by a spatial segmentation algorithm followed by global motion estimation and compensation. Region labelling is modelled as a MRF process, where
the optimisation of the objective function generates the final object/region labels. The proposed objective function includes four terms: the temporal continuity term, the motion constraint for detecting newly appearing objects/regions, the spatial continuity term, and the smoothness term. To expand the algorithm to multi-dimensional mode, the analysis and processing are computed in multi-dimensional space over the pyramid of the decomposed frame.

Considering the probable shortcomings of spatial segmentation in discriminating between the foreground and background regions in images with low contrast areas, two versions of the algorithm are proposed. In the first version, the algorithm uses the proposed scalable image segmentation to partition the image into different regions. In the second version, the regions are divided into several watershed basin regions by the watershed algorithm to obtain an over-segmentation algorithm which ensures the separation of foreground/background regions. The classification of watershed basins extracts foreground/background areas. The chapter is organised as follows. Section 6.2 describes the global motion estimation algorithm. In Section 6.3, single resolution MRF modelling and its objective function for classification is described. Different terms of the objective function, including the region-wise smoothness are explained in this section. Section 6.4 extends the single resolution segmentation to the scalable pyramid video frame segmentation. Initial estimation and optimisation of the objective function are discussed in this Section. Some experimental results are presented in Section 6.6, and finally conclusions are drawn in Section 6.7.

6.2 Global Motion Estimation

In image sequences, the camera motion as well as the object motion create differences between frames. Since for tracking the already detected objects and extracting the newly appearing objects, the object motion is examined, the camera motion should be estimated and compensated. To estimate the camera motion, it is often assumed that the background and stationary regions of the objects cover more than 50% of the image area. In other words, the camera motion is equal to the global motion in the frame. The global motion is often simple and consists of only translation
and possibly pan/zoom. Therefore an affine motion model described in Section 2.7 of the literature review chapter is often enough to model the global motion model.

First, the image is divided into different blocks and each block is assumed to have constant motion. Then the previous frame is searched to find the best matches for the current block [166]. The matching criterion is the mean square error, but for computational simplicity, often it is replaced by the mean absolute difference (MAD) which is expressed as follows:

\[
MAD(v_x, v_y) = \frac{1}{N \times M} \sum_{(i,j) \in B} |f_k(i, j) - f_{k-1}(i + v_x, j + v_y)|,
\]

where \( N \) and \( M \) are the width and length of the rectangular block area \( B \). In the forward motion estimation, the previous frame is replaced with the next frame. The lower resolution motion field is projected to the next level as the initial motion estimation and is refined through searching neighbouring vectors over that resolution.

After the dense motion field estimation, the parameters of the global motion model are estimated. Here the least squares method result proposed by Wang et al. [240] is used. The parametric affine motion model gives the following equations:

\[
\hat{v}_x = a_1 v_x + b_1 v_y + c_1 \\
\hat{v}_y = a_2 v_x + b_2 v_y + c_2,
\]

where \( \hat{v}_x, \hat{v}_y \) represent the parametric motion model. The difference between the dense and parametric motion models should be minimised by the least squares criterion.

\[
E = \sum_{\text{Background}} (\hat{v}_x - a_1 v_x - b_1 v_y - c_1)^2 + (\hat{v}_y - a_2 v_x - b_2 v_y - c_2)^2
\]

Minimising the expression over the background area gives the parameters \( a_1, \ldots, a_6 \) according to the least squares method. The global motion estimation is performed on the area, corresponding to the detected background in the previous frame. According
to the least squares method of Wang et al. [240], the six parameters of the affine model are denoted by the following equation. If \( P \) denotes the parameter matrix

\[
P = \begin{bmatrix}
a_1 & b_1 & c_1 \\
a_2 & b_2 & c_2
\end{bmatrix},
\]

(6.4)

and \( B \) shows the background area, the least squares error estimate of \( P \) is given by the following equation:

\[
P^T = \left[ \frac{1}{\sum_B v_x} \sum_B v_x \quad \frac{1}{\sum_B v_x^2} \sum_B v_x v_y \quad \frac{1}{\sum_B v_y} \sum_B v_y \right]^{-1} \times
\begin{bmatrix}
\sum_B v_x(x_i, y_i) \\
\sum_B v_y(x_i, y_i) \\
\sum_B v_x v_y(x_i, y_i)
\end{bmatrix}
\]

(6.5)

### 6.3 Single Resolution Markov Random Field Modelling

MRF-based processing is the most frequently used stochastic model in image processing and computer vision. It has the ability to capture the spatial continuity of natural images, and similarly it can capture the spatial and temporal continuity of video signals. Pixel-based processing increases the computational complexity of the algorithm; therefore, in this work, MRF modelling is used for region labelling. Regions are obtained from the spatial segmentation, therefore, region-based processing increases the spatial accuracy of the video segmentation processing. Since the number of regions is much less than the number of pixels, the presented algorithm is very effective.

In this section, the single resolution version of video segmentation is presented. The algorithm starts by partitioning the current frame into different regions using a suitable spatial segmentation algorithm. The proposed scalable segmentation algorithm is used, but in a single resolution mode with a smoothness constraint. If the MAP
estimation criterion is followed, the conditional probability of the segmentation labelling $X$, given the observations, should be maximised. The observations include the last frame segmentation $X^-$ and $\theta$ the motion information in the $P(X|X^-, \theta, I)$.

Using the Bayes theorem,

$$
P(X|X^-, \theta, I) \propto P(X^-|X, \theta, I)P(\theta|X, I)P(X|I) ,
$$

where $X$ is the current frame classification, $X^-$ is the previous frame classification and $\theta$ is the region motion vector.

The first term on the right hand side of equation 6.6 explains the temporal continuity of the segmentation field. The conditional probability of the estimated label field at the previous frame $X^-$ is modelled as a Gibbs distribution:

$$
P(X^-|X, \theta, I) = \frac{1}{z_1} \exp\{-E_T(X, X^-, \theta, I)\} ,
$$

where $z_1$ is a normalisation constant, which does not affect the optimisation process. The energy term $E_T(X, X^-, \theta)$ is modelled by the Gibbs distribution potentials $V_{R_i}^T$ over single cliques combined of just one region as follows:

$$
E_T(X^-, \theta, X, I) = \sum_{i=1}^{k} V_{R_i}^T(X^-, \theta, X, I)
$$

$$
V_{R_i}^T(X^-, \theta, X, I) = z_t Q_{R_i} 
$$

$k$ is the number of regions, and the index $i$ points to different regions. $z_t$ is a normalisation constant. $Q_{R_i}$ is the number of pixels in $R_i$ which after the back projection process have different label compare to the current frame. Therefore a smaller $Q$ indicates a higher probability that the region has the same label as the corresponding projection at the previous frame determined by $\theta_{R_i}$. The coefficient $z_t$ determines the trend to track the same label field for corresponding regions in consecutive frames. This term also allows tracking of stationary objects/regions.

The second term on the right hand side of equation 6.6 is motion constraint which explains the relationship of the motion vectors to the labelling process. It is modeled as a Gibbs distribution:

$$
P(\theta|X, I) = \frac{1}{z_2} \exp\{-E_M(\theta, X, I)\} 
$$

(6.9)
\( z_2 \) is a normalisation constant which does not affect the optimisation process. The region label fields along the motion trajectory should be conserved. Considering the compensated, global motion and the labels set as \( F, B \), the above-mentioned requirement for labels along the motion trajectory means: any non zero motion vectors indicate foreground areas. Therefore the energy term is formed by the Gibbs potential function as:

\[
E_{M}(X, \theta, I) = \sum_{i=1}^{K} V_{R_i}^{M}(X, \theta, I),
\]

(6.10)

where the energy term \( V_{R_i}^{M} \), corresponding to the region \( R_i \), is described as the following:

\[
V_{R_i}^{M}(X, \theta, I) = \begin{cases} 
-\alpha A_i & (X_{R_i} = F \text{ and } \theta_{R_i} \neq 0) \text{ or } (X_{R_i} = B \text{ and } \theta_{R_i} = 0) \\
+\alpha A_i & (X_{R_i} = F \text{ and } \theta_{R_i} = 0) \text{ or } (X_{R_i} = F \text{ and } \theta_{R_i} \neq 0)
\end{cases}
\]

(6.11)

where \( A_{R_i} \) is the size of the region \( R_i \), and \( \alpha \) is a coefficient. This term encourages moving regions to be classified as foreground. The magnitude of the motion vector is not considered, but only whether it is zero or not. Therefore a simple translation model for the motion vector can be considered, which significantly reduces the computational complexity.

The third term on the right hand side of equation 6.6 models the spatial continuity of the segmentation field. It is modelled as a Gibbs distribution whose energy term \( E_{Sp} \) is formed by the Gibbs potentials \( V_{Sp}^{R} \) as a clique function of two neighbouring regions \( R_i \) and \( R_j \) as follows [5]:

\[
V_{R_i,R_j}^{Sp}(X, \theta) = \begin{cases}
-z_f \cdot f(M_{R_i} - M_{R_j})N_{R_i,R_j}, & X_{R_i} = X_{R_j} = F \\
-z_b \cdot f(M_{R_i} - M_{R_j})N_{R_i,R_j}, & X_{R_i} = X_{R_j} = B \\
z_{diff} \cdot f(M_{R_i} - M_{R_j})N_{R_i,R_j}, & X_{R_i} \neq X_{R_j}
\end{cases}
\]

(6.12)

where \( N_{R_i,R_j} \) is the length of the common border between the regions \( R_i \) and \( R_j \). \( M_{R_i} \) and \( M_{R_j} \) are the means of regions \( R_i \) and \( R_j \) respectively. \( f \) is a function of the averages, which gives a small value for dissimilar regions and a large value for
similar regions. A good definition for $f$ is given by Tsaig et al. [5], which is shown in Figure 6.1, and its formula can be expressed as the following:

$$f(d) = \begin{cases} 
T_h & d < d_l \\
T_l - \frac{T_h - T_l}{d_h - d_l} (d - d_l) & d_l < d < d_h \\
T_l & d > d_h ,
\end{cases}$$

where $T_l$, $T_h$, $d_l$ and $d_h$ are the entered thresholds. Therefore, two regions with similar spatial properties are more likely to have the same label.

### 6.3.1 Smoothness Factor

The energy function of the MRF-based model labelling is equal to:

$$E_X(X^-, \theta, X, I) = E_T(X^-, \theta, X, I) + E_M(\theta, X, I) + E_{S_p}(X, I),$$

where each one of the above three energy functions modelled by MRF and their corresponding potential functions were obtained in the previous frame. The other factor that can be added to the energy function is a smoothness term. As explained in Chapter 4, natural objects have smooth shapes, therefore smoothness can contribute to the classification process. If a process on a region increases the foreground/background smoothness, then it is an indication of the validity of the process. This term is especially effective for the regions where the other terms in the objective function cannot
strictly determine the classification. For example, consider the values of the objective function for a region where different classification labels are very close. This can happen in the regions around an object’s border where the contrast between neighbouring regions is not large enough. In this situation, the smoothness factor leads to classification towards smoother object extraction. The defined smoothness function in Chapter 4 is pixel-based, which is useful for pixel classification. However for the region-based classification, the smoothness should be extended to the region-based definitions. The proposed smoothness corresponding to region \( R \) is the average of the smoothness function along its common border with regions having different classification labels. If the foreground smoothness values before and after merging region \( R \) are equal to \( SMT_1 \) and \( SMT_2 \), respectively, then the value of:

\[
\Delta SMT = SMT_2 - SMT_1
\]

shows the increase/decrease of foreground smoothness due to merging region \( R \). The value of \( \Delta SMT \) affects the classification of region \( R \). Figure 6.2 shows the region smoothness effect for object classification.

This value multiplied by a coefficient \((l)\), is added to the objective function as the fourth term. Therefore the objective function is a composition of four terms as the

Figure 6.2 The processed regions are shaded. (a) Adding the shaded region to the object area increases the smoothness; (b) Adding the shaded region to the object area decreases the smoothness.
follows:
\[
U(X|I, X^-, \theta) = \sum_{i=1}^{k} \left\{ z_t.Q_{R_i} + \alpha.A_{R_i} + \sum_{R_j \in \partial R_i} z_X.f(M_{R_i} - M_{R_j}).N_{R_iR_j} + l.\Delta(SMT) \right\}, \quad (6.15)
\]

where \( \partial R_i \) the set of neighbouring regions of \( R_i \) and \( z_X \) is equal to:

\[
z_X = \begin{cases} 
-z_f & X_{R_i} = X_{R_j} = F \\
-z_b & X_{R_i} = X_{R_j} = B \\
z_{\text{diff}} & X_{R_i} \neq X_{R_j}
\end{cases}
\] (6.16)

### 6.4 Multiresolution Scalable Video Segmentation

In this section, the proposed video region labelling algorithm is developed to scalable multiresolution mode. First the wavelet transform decomposes the proposed frame into different resolutions. Three levels of decomposition is used. The proposed scalable image segmentation partitions different levels of the pyramid into homogenous regions. Scalability and smoothness are segmentation constraints. Therefore every region has corresponding regions at lower and higher resolutions where the down-sampling relation between these regions is maintained. The corresponding regions are classified using the same label. Therefore they are processed together. This proposes a multi-dimensional processing similar to pixel processing where the symbol \( \{S\} \) points to the region \( S \) and its corresponding regions at other resolutions. Instead of multi-dimension, the term vector is used for convenience. With this introductory preparation, the objective function of single resolution video region labelling explained in equation 6.15 is extended to the multiresolution mode. The computations of the processed features of the regions such as intensity/colour mean, motion and percentage of the projection are tracked in multi-dimensional space. The symbol \( \{\} \) points to the multi-dimensional computation of the features. The first term, \( Q_{\{R_i\}} \) is considered as the average of the single resolution definition of \( Q_R \) at different resolutions for corresponding regions of \( \{R_i\} \). The potential function \( V^M_{\{R_i\}} \) in equation 6.11 is simply extendable to vector mode, where it is identified by \( V^M_{\{R_i\}} \). The corresponding regions similarly are moving or stationary. Since the exact value of the motion
vector is not needed, the region’s motion vector at a specific resolution such as lowest resolution is used for the corresponding regions at different resolutions\(^1\). The average size of the region’s area at different resolutions is used for \(A_{R_i}\). The spatial continuity is the average of the computed spatial continuity term \(V^{C}_{R_i R_j}\) at different resolutions. Similarly the smoothness term is the average of the computed smoothness term at different resolutions. In a similar way to the spatial segmentation, the smoothness of the different resolutions can be emphasised by considering different coefficients for the smoothness term at different resolutions. Therefore the objective function for the scalable multiresolution video segmentation is equal to:

\[
U(X|I, X^{-}, \theta) = \sum_{\{S\}} \left\{ z_t \cdot Q_{(s)} \pm \alpha \cdot A_{(s)} + \sum_{p \in \partial\{s\}} z_x \cdot f(M_{(s)} - M_{(p)}) \cdot N_{(s)}(p) + \sum_{q \in \{S\}} l_{\text{res}(q)} \cdot \Delta SMT(q) \right\}, \quad (6.17)
\]

where \(q\) is a region of the set \(\{S\}\) of regions which includes region \(S\) and its corresponding regions at different resolutions. \(\partial\{s\}\) is the set of neighbouring regions of \(\{s\}\), and \(z_x\) is equal to:

\[
z_x = \begin{cases} 
-z_f & X_{(s)} = X_{(p)} = F \\
-z_b & X_{(s)} = X_{(p)} = B \\
z_{\text{diff}} & X_{(s)} \neq X_{(p)}
\end{cases}
\]

### 6.4.1 Objective Function Optimisation

The objective function should be optimised by one of the MRF optimisation methods. However, at first, an initial estimation is necessary. The initial estimation is obtained by considering the temporal continuity term. The regions are simply back projected to the previous frame, and the number of object pixels is counted. If the ratio of the counted object pixels over the area of a region is more than a threshold, the processed region is considered as a foreground area. In multiresolution mode, the average of the computed ratio at different resolutions is compared with the threshold. Then an ICM-like optimisation is performed. However a raster scan of regions, unlike the

\(^{1}\)Therefore the global motion estimation and motion compensation are needed only at this resolution.
raster scan of an image’s pixels, does not have a physical interpretation. Since large size regions are more likely to be classified correctly, regions are put in a queue in the order of their size from large to small size regions. The correct classification of large regions can help with the right classification of their neighbouring small regions. Regions are visited according to the priority queue. For any vector region such as \( \{S\} \), the terms of the objective function in this vector region are optimised given the classification of all the other regions. The objective function related to this vector region is in the following equation:

\[
U(\{S\}) = z_t Q_s(\{S\}) \pm \alpha A_s(\{S\}) \sum_{p \in \partial(\{S\})} z_x f(M_s - M_p) . N_s(p) + \sum_{q \in \{S\}} l_{res(q)} \Delta SMT(q)
\]  

(6.18)

One cycle of optimisation process continues until the queue is empty. The convergence criterion updates more than a threshold value such as 5% of regions, in one cycle of region visits. To reduce the computational complexity, regions which when back projected to the previous frame are covered by foreground (background) pixels by more than a threshold such as 90% do not need reclassification, and they take part in the objective function only for classification of their neighbouring regions. The different coefficients are determined empirically.

However, more reduction in the computational complexity is achieved by classifying each region in a proper resolution and extending the result to the corresponding regions at the other resolutions. Depending on the size of the region and the defined thresholds, a resolution is selected, the region at that single resolution is classified, and the result is extended to the other lower and higher resolutions. For example, the largest regions are classified at the lowest resolution, and very small regions are classified at the highest resolution. This significantly reduces the computational complexity because motion estimation and back projecting at the lower resolution has much less computational complexity than at the higher resolutions. Experimental results confirm that, if the proposed thresholds for selecting the resolution according to the region size are considered, this procedure’s result is the same as that of the classification of multiresolution vector regions.

The proposed objective function does not need the exact motion vector. Therefore a
simple translational motion model in the following equation is used, which reduces
the computational complexity:

\[
\begin{align*}
\hat{v}_x &= b \\
\hat{v}_y &= c
\end{align*}
\] (6.19)

The motion is obtained by shifting the region over the last frame and finding the best
match. The second energy term \(E_M\) in the objective function encourages regions with
non-zero motions to be classified as foreground. The problem behind this classifi-
cation is the occlusion problem related to covered and uncovered regions described
in Section 2.7 in the literature review chapter. Backward apparent motion classifies
these regions as moving regions, and in the classification they might be incorrectly
detected as foreground regions. To remove this problem, only valid motion vectors
in the energy term \((E_M)\) in the objective function are processed. The backward mo-
tion vector such as \((v_{x1}, v_{y1})\) computed for region \(A\) is valid, if the corresponding
forward motion vector from the projected regions in the previous frame toward cur-
rent frame is in the opposite direction. However, in practice some variations could
be tolerated and a threshold for the differences can be determined. These will project
the corresponding region in the previous frame to region \(A\). Figure 6.3 explains this
relationship. Otherwise this motion vector is called invalid and is replaced with the
zero vector. This replacement prevents the detection of uncovered regions.

## 6.5 Object’s Border Fine Tuning

For most of the object-based applications such as video editing and manipulation,
the object of interest should be extracted with pixel-wise accuracy. However, the
proposed scalable grey-level segmentation can result in under-segmentation and may
fail in discriminating between foreground and background objects in areas with low
contrast. One way to increase the discriminating power of the segmentation is by us-
ing colour segmentation, which partitions the image into more regions than the grey-
level segmentation, which decreases the under-segmentation and increases the com-
putational complexity of spatial segmentation. Furthermore, increasing the number
of regions, increases the computational complexity of the classification. However, in
some image sequences with low colour contrast, under-segmentation can still happen. In this case, the suggestion is to divide the image into watershed basins, which results in an over-segmentation including many small regions [59, 181]. The region growing algorithms can also produce over-segmentation, but the watershed is more faithful to the natural borders.

To retain the smoothness feature of the extracted regions and ensure visually pleasing segmentation, the scalable multi-resolution grey/colour image segmentation is used. The regions which are smaller than a threshold are left, and the other regions are divided into smaller basin regions by the watershed algorithm [59]. The watershed basins are also down-sampled to lower resolutions to create the corresponding regions at the lower resolutions. Subsequently, the vector basin regions are classified. This leads to avoiding the unnecessary partitioning of small regions and retaining most of the aesthetically pleasing borders resulting from the scalable segmentation. Figure 6.4 shows the idea. If the spatial segmentations of the frame be displayed in Figure 6.4 (a), the partitioning of the regions to the basins is shown in Figure 6.4 (b).
Partitioning into basins removes the under-segmentation problem, but it significantly increases the number of regions and the computational complexity of the labelling optimisation process. In addition, due to the process of more information in the large size regions, their classification is also more confident than for small size basin regions. However, the challenge is how to automatically determine the use of grey-level or colour segmentation and whether the partitioning of the image into watershed basin regions is necessary or not. It is clear that it depends on the contrast between foreground and background. However, except through human intervention, we are not aware of any effective solution for an automatic decision to choose regular or over-segmentation for generic application. This is somewhat similar to the problem of threshold and parameter tuning that requires many thresholds and parameters to be set by the users in different algorithms for image/video processing and generally in signal processing algorithms.

6.6 Experimental Results and Discussion

To evaluate the performance of the proposed algorithm, five different MPEG-4 sequences, Clair, Hall Monitor and Foreman CIF sequences, Table Tennis SIF se-
In each sequence, as a first step, in the first frame, a user determines the rough boundary of the object of interest through a graphical user interface (GUI). Subsequently, all regions for which the majority of their area, more than a predetermined percentage such as 50%, is located inside this closed contour are selected to belong to the extracted object. This is fully explained in the second example of the experimental results section in Chapter 3. The user intervention can be reduced to minimum where the user only determines the type of the object of interest, such as “head and shoulders”, and the object of interest extraction algorithm presented in Chapter 5 provides the shape in the first frame.

In the first example, the proposed video segmentation and tracking algorithm is run over the 75 frames of the Clair image sequence. The extracted objects in frame numbers 20, 45 and 65 in multiresolution mode are shown in Figure 6.5(a), (b) and (c).

To compare the proposed algorithm with other region-based object tracking and extraction methods, an alternative tracking algorithm is used. It is an ordinary backward tracking algorithm [241, 242] which includes only the temporal continuity term at the
highest resolution. First the current frame is partitioned into different regions by the MRF-based single resolution image segmentation proposed by Pappas [4]. Each region is then back projected to the previous frame. If the number of projected pixels inside the foreground area at the previous frame is more than a threshold, such as 50% of the region’s area, the region is classified as a foreground region. The alternative algorithm will be called the “regular (backward) tracking algorithm”.

Both the proposed scalable video segmentation and regular tracking algorithms are performed, and the extracted objects are compared subjectively and objectively. Our qualitative criterion for objective comparison is border smoothness of the extracted objects. Object smoothness is averaged over the curvature of the foreground’s border. Although it is not an ideal criterion, it has confirmed the results of our subjective tests. The smoothness comparison for the 75 frames of the Clair sequence for the 3 resolution levels are shown in Table 6.1\(^2\). The smoothness term modifies the segmentation in areas of the image that have lower grey-level contrast. In the Clair sequences the regions around the head have lower contrast compared to the shoulder and body areas. If only the head area is considered, the smoothness improves by 13.17%, 11.5% and 10.5% at different resolutions. As a subjective test example, Figure 6.6 shows the extracted objects of the 23rd frame of the Clair sequence when using the scalable algorithm and regular tracking algorithm, respectively. In this figure, images of different resolutions are shown at the same size to highlight the details. The analysis of both images shows that our algorithm has extracted a smoother and more visually pleasing object.

In the second example, the standard MPEG-4 Table_Tennis sequence which has textured background with fast moving objects is processed. In Figure 6.8, the frame

---

\(^2\)The proposed scalable tracking algorithm directly produces the object at different resolutions, however, the object produced by regular tracking algorithm is down-sampled to lower resolutions.
numbers 10, 20 and 32 and the extracted objects are shown. As an example, observe the objects in frame number 10 of the Table_Tennis sequence that were extracted by the proposed scalable tracking algorithm and by the single level version of the proposed tracking algorithm without any smoothness criterion. The objects extracted at 3 different resolutions are shown in Figure 6.8. For a quantitative comparison the object smoothness is measured for the first 35 frames of the sequence as presented in Table 6.2. Again, if only the hand and fingers with the racket are considered, the smoothness is nearly doubled. The computational complexity of the multiresolution tracking algorithm is reduced, typically to less than 30% of tracking at the finest resolution, because smaller regions and less motion decrease the complexity of the matching procedure at lower resolutions.

The proven high noise tolerance of the multiresolution image segmentation [238] is extended to video segmentation by the proposed algorithm. In video object extrac-
Figure 6.7 Table Tennis object extraction: (a) frame 10; (b) frame 23; (c) frame 32.

Figure 6.8 Table Tennis object 10th frame: (a₁) scalable 240 × 352; (b₁) scalable 120 × 176; (c₁) scalable 60 × 88; (a₂) regular 240 × 352; (b₂) regular 120 × 176; (c₂) regular 60 × 88.

<table>
<thead>
<tr>
<th></th>
<th>60 × 88</th>
<th>120 × 176</th>
<th>240 × 352</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scalable Tracking</td>
<td>55.6</td>
<td>53.87</td>
<td>53.10</td>
</tr>
<tr>
<td>Regular Tracking</td>
<td>58.82</td>
<td>57.63</td>
<td>56.22</td>
</tr>
<tr>
<td>Improvement</td>
<td>6.84%</td>
<td>6.97%</td>
<td>5.88%</td>
</tr>
</tbody>
</table>

Table 6.2 Table Tennis sequence smoothness.

...tion, especially at low contrast areas, noise can adversely affect the region matching, resulting in wrong classifications. For example, some small background regions...
close to object areas are merged with the object, and some regions belonging to the object areas are merged with the background. To overcome these matching errors, the proposed algorithm effectively uses the noise-reduced, lower resolution information to classify the regions. This is possible due to the proposed multiresolution video segmentation algorithm.

To test the algorithm in noisy environments, a uniform noise in the range $(-25, +25)$ is added to the Table_Tennis sequence. The noisy sequence is segmented with the proposed algorithm, and the results are compared with the single level tracking algorithm. Table 6.3 presents the smoothness of both algorithms. The misclassified numbers of pixels for different resolutions are counted in Table 6.4. The number of misclassified object pixels in the scalable multiresolution video segmentation algorithm decreases to about 50% of the pixels misclassified by the regular single level segmentation algorithm. This confirms the superiority of the multiresolution algorithm. Figure 6.9 shows the extracted objects in frame 18 for both multiresolution and single level object extraction.

In the third example, the Hall_Monitor CIF sequence is segmented. In this example
the object of interest appears gradually. Consequently, the change detector embedded in the second term of the MRF objective function identifies newly appearing objects/regions, while the tracking algorithm inherited in the first term of the MRF objective function detects already present objects/regions. In this algorithm, due to low contrast of the foreground and background, the spatial segmentation cannot discriminate between the foreground and background in some areas of the image. Therefore, the algorithm partitions the regions bigger than 20 pixels by the watershed algorithm, and the basin regions are classified.

The object of frame number 40 extracted by the scalable algorithm at different resolutions is shown in Figure 6.10. The extracted objects of frame numbers 34, 44 and 60 using the scalable and the regular algorithms can be seen in Figure 6.11. Some regions related to shaded areas are also detected as objects, because the shading between two consecutive frames is also changed. This requires more sophisticated processing of motion information than the motion constraint considered in the second term of the objective function. Increasing the change detector thresholds can reduce the size of detected areas of shading but increases the risk of missing some parts of the object during the detection process. Figure 6.11 confirms the superiority of the proposed algorithm over the regular object detection algorithm in creating a visually more pleasing segmentation. Table 6.5 confirms the improved smoothness of the proposed algorithm.
Figure 6.10 Hall Monitor sequence object extraction at frame 40; (a) resolution 288 × 352; (b) resolution 144 × 176; (c) resolution 72 × 86.

Figure 6.11 Hall Monitor sequence object extraction: (a₁) scalable extraction at frame 34; (b₁) scalable extraction at frame 44; (c₁) scalable extraction at frame 60; (a₂) regular extraction at frame 34; (b₂) regular extraction at frame 44; (c₂) regular extraction at frame 60.

In the fourth example, the 75 frames of the QCIF size Mother & Daughter colour image sequence are processed. The frames are in YUV format, where Y is in full resolution and U and V are in half resolution. The images are segmented by the proposed scalable colour image segmentation at 3 different resolutions. The object of interest is selected by user intervention at the first frame, and it is tracked in the next frames by the proposed video segmentation algorithm. In Figure 6.12, the frame numbers 32, 50 and 68 are shown with the extracted objects at the highest resolutions.
In Figure 6.13 the objects of frames 48, 60 and 72 extracted by the proposed scalable algorithm and regular tracking algorithm are compared. The objects extracted by the proposed object extraction algorithm are shown in the top row of the Figure. The objects extracted by the regular tracking algorithm are shown in the second row. Subjective comparison shows the better visual quality of the proposed object extraction algorithm.

In the last example, the 50 frames of the CIF size colour image sequence Foreman are segmented. The images are in YUV format, where U and V are in half resolution. Each frame is segmented by the proposed scalable colour image segmentation at three resolutions. The object of interest is determined by user intervention at the first frame. It can also be automatically extracted as proposed in the previous chapter. The proposed video segmentation algorithm tracks the object of interest in the consecutive frames. The extracted objects in frames 5, 20 and 30 are shown in Figure 6.14. In Figure 6.15, the extracted object from frame 18 is shown at three different resolutions. Finally, in Figure 6.16, the objects extracted by the proposed scalable video segmentation algorithm and the algorithm proposed by Zhou et al. [243] can be seen. The algorithm proposed by Zhou [243] is a forward tracking algorithm. It segments the extracted object at the current frame, and then each region is projected to the next frame by the estimated affine motion model for the region. Then the projected regions’ borders are refined until convergence. In the refinement phase, each border pixel is examined, and its label is updated to one of the neighbouring regions for which the reverse of its motion model at the current pixel has the least motion.
Figure 6.13 Mother & Daughter sequence object extraction by the proposed scalable algorithm and regular tracking algorithm at different frames: (a) scalable at frame 48; (b) scalable at frame 58; (c) scalable at frame 72; (d) regular tracking algorithm at frame 48; regular tracking algorithm at frame 58; regular tracking algorithm at frame 72.

Figure 6.14 Foreman image sequence object extraction at frame (a) 5; (b) 20; (c) 30.

compensation error. Subjective comparisons confirm the superiority of the proposed algorithm. The object border smoothness is shown as an objective test in Table 6.6. Comparison of the smoothness for the two segmentation algorithms confirms the superiority of the proposed algorithm.
Figure 6.15 Foreman image sequence object extraction at frame 18 by the proposed scalable segmentation algorithm: (a) extracted object at $288 \times 352$ resolution; (b) extracted object at $144 \times 176$ resolution; (c) extracted object at $72 \times 88$ resolution.

Figure 6.16 Foreman image sequence object extraction at frames (a) 8; (b) 28; (c) 42.

Table 6.6 Foreman sequence smoothness.

<table>
<thead>
<tr>
<th></th>
<th>$72 \times 88$</th>
<th>$144 \times 176$</th>
<th>$288 \times 352$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scalable Tracking</td>
<td>48.4</td>
<td>49.2</td>
<td>48.4</td>
</tr>
<tr>
<td>Regular Tracking</td>
<td>58.7</td>
<td>59.3</td>
<td>56.2</td>
</tr>
<tr>
<td>Improvement</td>
<td>17.5%</td>
<td>17%</td>
<td>13.8%</td>
</tr>
</tbody>
</table>
The simulation details include the number of frames, size of frames, grey-level or colour images, with/without global motion estimation and compensation, divided to basins or not, average the time of frame processing and the number of processed frames per minute for the proposed scalable algorithm. Details of the proposed and the alternative algorithms for different sequences are shown in Tables 6.7 and 6.8. The strings “++” and “−” declare that the sub-process determined at that column’s title is performed for that sequence or not. The following comparisons were made:

- The Clair and Mother & Daughter sequence: compared with regular backward tracking without global motion compensation.

- The Table_Tennis and Hall_Monitor sequences: compared with an algorithm similar to the proposed algorithm, but in the single resolution mode without the smoothness constraint.

- The Foreman sequence: compared with the forward tracking algorithm of Zhou [243], which does not include global motion estimation.

In the Clair and Mother & Daughter sequences, the alternative regular backward tracking is faster than the proposed algorithm, because the proposed scalable segmentation includes loop of optimisation procedure which continues until convergence.

In the Table_Tennis and Hall_Monitor sequences, the running time for the proposed multiresolution scalable segmentation is longer than the alternative single resolution algorithm. The reason is the computation of the smoothness term. If the smoothness term is deleted from the optimisation process, the computational complexity of the proposed scalable algorithm decreases to 50% to 70% of the corresponding single resolution classification algorithm.

Although inherently the algorithm can be performed in real time, practically, as the Tables 6.7 and 6.8 show, due to too much computational complexity the algorithms are not real time. In the sequences such as Table_Tennis and Foreman which need global motion compensation, the computational complexity is much higher. Also, switching from the grey-level to colour segmentation nearly doubles the complexity.
Table 6.7 Details of the proposed scalable video segmentation algorithm.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>No. of frames</th>
<th>Size</th>
<th>Grey/Color</th>
<th>Global Motion</th>
<th>basins</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clair</td>
<td>75</td>
<td>CIF</td>
<td>grey</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>Table_Tennis</td>
<td>35</td>
<td>SIF</td>
<td>grey</td>
<td>++</td>
<td>--</td>
</tr>
<tr>
<td>Hall_Monitor</td>
<td>65</td>
<td>CIF</td>
<td>grey</td>
<td>--</td>
<td>++</td>
</tr>
<tr>
<td>Mother &amp; Daughter</td>
<td>75</td>
<td>QCIF</td>
<td>colour</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>Foreman</td>
<td>50</td>
<td>CIF</td>
<td>colour</td>
<td>++</td>
<td>--</td>
</tr>
</tbody>
</table>

Table 6.8 Running times for the proposed and the alternative algorithms, performed on a Pentium 4 PC with 512 MBytes Ram.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Proposed Scalable</th>
<th>Alternative Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sec/Frame</td>
<td>Frame/Min</td>
</tr>
<tr>
<td>Clair</td>
<td>6.9</td>
<td>9</td>
</tr>
<tr>
<td>Table_Tennis</td>
<td>76</td>
<td>0.8</td>
</tr>
<tr>
<td>Hall_Monitor</td>
<td>19.3</td>
<td>3</td>
</tr>
<tr>
<td>Mother &amp; Daughter</td>
<td>12.8</td>
<td>4.7</td>
</tr>
<tr>
<td>Foreman</td>
<td>148</td>
<td>0.4</td>
</tr>
</tbody>
</table>

Similarly, decomposition of the segmented grey regions to basins increases the computational complexity by about 3 times. In some tracking algorithm such as [243], the global motion estimation is deleted, which decreases the computational complexity. However, this algorithm tracks the already detected objects, and detecting newly appearing objects is not considered.

6.7 Conclusion

In this Chapter a new semi-automatic MRF-based multiresolution video segmentation algorithm for VOP extraction is proposed. The objective function of the algorithm includes spatial and temporal continuity. Temporal continuity tracks the objects already extracted in the previous frames even when they stop. The motion constraint term detects newly appearing objects/regions. The motion validity examination removes the occlusion problem. Region continuity considers the spatial consistency of the labelling algorithm. Region smoothness is introduced as a new criterion for region classification and is added to the objective function. The algorithm is extended to multiresolution by considering the corresponding regions at different resolutions.
and processing them in multi-dimensional or vector space. The final solution is obtained by the MAP criterion and an ICM-like optimisation method. The objects are extracted at different resolutions of the pyramid. The algorithm includes a version for object extraction from scenes with low grey-level or colour contrast. This version divides the region into watershed basin regions and classifies the basins. The proposed method provides fine localization of the borders of regions. Multiresolution processing allows larger motion, better noise tolerance and less computational complexity. The algorithm also deals with the occlusion problem and corrects motion estimation. Comparison with different algorithms confirms the superiority of the proposed algorithm.

For further improvement of the algorithm, a more sophisticated solution for the occlusion problem can be considered. Better processing of the motion information to prevent shade detection is also necessary. Discrimination between different objects in the scene can be considered. More research is needed to determine the necessity of partitioning the segmentation into basins. Most of the computational complexity of the algorithm lies within the global motion estimation. Therefore more effective global motion estimation or deleting its role from the algorithm can be considered. Finally more research into fully automatic object extraction including the identification of the object of interest in the first frame and the automatic determination of the parameters and thresholds are necessary.
Chapter 7

Summary, Conclusions and Future Research

7.1 Introduction

This final chapter presents a summary of the thesis and concluding remarks followed by some suggestions for new directions and improvements. The thesis has considered multiresolution image and video segmentation and object extraction algorithms with visual quality and scalability as constraints. Segmentation and object extraction have a wide range of applications, such as pattern recognition, machine vision, content-based image/video retrieval. Although the results are useful for generic segmentation applications, the focus is placed on scalable wavelet-based object coding, which can efficiently distribute the visual information over networks. This application requires multiresolution scalable segmentation and effective object extraction. Moreover, multiresolution processing is often useful for reducing the computational complexity which is one of the significant issues in image/video segmentation algorithms. Towards ensuring an effective and useful segmentation, a visual quality criterion was added to the segmentation algorithm. Furthermore, semantic segmentation for limited number of applications was explored. Having proved the efficacy of semantic segmentation for selected “objects-of-interest”, it is possible to extend the algorithm to much larger categories of “objects-of-interest”.

The algorithms were presented in two main categories: image “object-of-interest”
extraction and moving video object extraction. For the first category, the aim is to perform a fast search through the image to find the “object-of-interest”. For the second category, the aim is to classify the regions in a frame as foreground or background. For both categories objects are extracted at different resolutions with scalability and visual quality as constraints.

7.2 Summary and Conclusions

The thesis has addressed two important categories of meaningful image and video segmentation algorithms which are (1) image “object-of-interest” extraction and (2) moving video object extraction. The input can be a grey or colour image/video, and user intervention could be limited to only determining the kind of object, such as human head and shoulders, car, etc. The work has proposed several algorithms for low level and high level image and video segmentation, which are summarised as follows:

- **Chapter 1** provided a brief introduction to object-based processing and motivation. A deeper insight into the problem statement and goals of this thesis were presented in this chapter. The organisation, description and major contributions of the thesis were outlined as well as a list of publications resulting from the research.

- **Chapter 2** presented background information on issues related to the image/video segmentation and also reviewed the outstanding works in the literature on multiresolution image segmentation, semantic segmentation and video segmentation. The literature survey included the development of a classification scheme for the segmentation algorithms. The chapter was concluded with research directions and explains the selected approaches for achieving the goals.

- **In Chapter 3** the concept of scalability was explained and two novel multiresolution image segmentation algorithms were introduced. First the down-sampling relationships between objects at different resolutions were intro-
duced. Then a morphology-based multiresolution image segmentation algorithm was proposed. At the lowest resolution, the image is segmented by the watershed basin method, and region merging is used to decrease the oversegmentation. Examination of the edge validity of borders between two adjacent regions allows merging of more neighbouring regions. Low resolution segmentation is projected to the next higher resolution, and the projected segmentation at the higher resolution is refined until the highest resolution is segmented. Region borders are matched with the watershed basins which results in smooth and well localized borders. The detection of the new objects/regions at the current resolution removes under-segmentation, which is a common problem with ordinary multiresolution segmentation algorithms. However, the downside of the process is the increase in the computational complexity. The proposed morphology-based algorithm, and similarly the other progressive multiresolution segmentation algorithms, cannot provide the required scalability for the multiresolution scalable object extraction algorithm.

To provide the required scalability feature for the segmentation algorithm, a MRF-based segmentation algorithm was proposed. The algorithm ties the corresponding pixels at different resolutions together as a vector and extends the objective function of the regular MRF-based single resolution segmentation to vector space. A novel idea for extending the clique function to multidimensional space was introduced. In the proposed algorithm the pixel labels are processed/updated using a multi-dimensional vector covering all resolutions, hence ensuring scalability. A modified ICM optimisation approach provides the low to high resolution segmentation and high to low feedback. The proposed algorithm provides a good balance between over- and under-segmentation compared to the single and multiresolution segmentation algorithms. While it detects more regions than ordinary multiresolution segmentation algorithms, it is still noise tolerant.

- **In Chapter 4** the proposed scalable grey level segmentation to enhance two aspects of the process was developed. The first aspect was related to the visual quality of the segmentation where a smoothness criterion was introduced and incorporated into the objective function of the segmentation algorithm. Al-
though the proposed smoothness criterion is not directly related to the semantic concept, the experimental results confirm its efficiency in extracting visually pleasing objects. By using different coefficients for different resolutions, the desired visual quality at all resolutions was maintained.

The second aspect was related to enhancing the segmentation process by using colour information. The proposed objective function is extended to colour space, while maintaining the scalability and smoothness as constraints. To reduce the computational complexity, the region borders are refined until convergence. Segmentation of different spaces is considered and discussed. The proposed algorithm can segment the colour image sequences in MPEG-4 databases where chrominance components are in half resolution. The advantages of the scalable grey image segmentation algorithms, such as better noise tolerant, better compromise between over and under segmentation, etc., also exist in the scalable colour image segmentation algorithm.

- Chapter 5 presented a hierarchical “object-of-interest” extraction algorithm. First a template matching algorithm was introduced, then a single resolution search through the image which examines different regions combinations was explained, and its computational complexity was discussed. An irregular pyramid including different segmentation maps corresponding to the regular pyramid, segmented by the scalable segmentation was introduced. The hierarchical template search was implemented using the newly introduced irregular pyramid, which is organised in a stack. The proposed search implements the GPE of the HVS and finds the global and large size objects first and the small and local objects later. It is assumed that the image is often searched for global objects. The proposed multiresolution search is consistent with the HVS where finding small size objects needs more attention, corresponding to more computational complexity in machine vision. This algorithm can be used for many different applications where finding the “object-of-interest” in the scene is required. In particular, it is useful for video tracking algorithms where the “object-of-interest” is determined at the first frame through user intervention and is tracked through the other frames. The requirement for user intervention can be eliminated by using the proposed algorithm to detect the “object-of-
Summary, Conclusions and Future Research

interest” in the first frame.

- **In Chapter 6** the object extraction algorithm was extended to video signals. Moving objects were extracted while maintaining the scalability and visual quality at different resolutions as constraints. The concept of smoothness was extended from the pixel-based definition to region-based. A MRF-based region classification algorithm extracted the foreground and background regions. The smoothness constraint as well as the spatial and temporal continuity and motion constraints were incorporated into the objective function of the classification algorithm. The proposed algorithm resulted in the extraction of more visually pleasing video objects while allowing for larger motion, better noise tolerance and less computational complexity. Incorporating the semantic object extraction algorithm proposed in the previous chapter with the tracking algorithm facilitates automatic semantic video object tracking and segmentation.

For each proposed approach, all the necessary mathematical basis, justifications and experimental results were provided. A smoothness criterion for objective performance evaluation of the visual quality criterion was used. Subjective testing has confirmed a sufficient correlation between this criterion and visual quality, which is a semantic concept. In conclusion, this thesis has presented several novel techniques for low level and high level image and video segmentation algorithms. They constitute a significant contribution towards semantic segmentation.

### 7.3 Future Research

A number of significant issues related to the scalable segmentation and “object-of-interest” extraction has been addressed in this work. However, there are still a number of challenges and possible improvements which require further research. Some suggestions for future research in this framework are addressed below:

- Although the Bayesian framework has the flexibility to implement the scalable segmentation, its dependence on an initial estimation of the segmentation is a
significant shortcoming. Performance of the Bayesian segmentation algorithm relies on this initial estimation especially with optimisation methods which trap in the local optimum. For initial segmentation, often the k-means clustering is used. However, the number of labels, which is a critical parameter and significantly affects the results, should be entered manually. It is, therefore, essential to find an effective method for initial parameter estimation for Bayesian based segmentation algorithms. The suggestion is a hybrid model such as a combination of the Bayesian based optimisation algorithm and a method such as region growing. The proposed method should also be applicable in the multiresolution framework. This method will be an effort towards an unsupervised Bayesian based segmentation algorithm.

- The proposed scalable image segmentation algorithm is used in the image and video object extraction algorithms. In this application, the “object-of-interest” such as head and shoulder, cars, etc., are the main subject of the image, and grey or colour features are enough for the segmentation. Although to some extent the proposed algorithm can segment the textured regions in normal images, for a comprehensive solution texture segmentation should also be considered. A hybrid of grey/colour segmentation with texture segmentation could be more efficient in segmenting different images. However, texture is resolution dependent, and a scalable texture segmentation algorithm and its integration into the grey/colour segmentation requires more research.

- The proposed smoothness function is not a perfect visual quality criterion. Although it improves the visual quality, it cannot prevent semantic distortion. Finding a function which effectively represents the visual quality is a challenging task which needs more research. This could improve the segmentation performance significantly. Towards this end, topology constraints could be considered as an idea for further research.

- Although some deformable templates were used in the proposed “object-of-interest” extraction algorithm, for an effective and generic algorithm, more effort in employing deformable templates for extracting dynamic objects such as a walking human is necessary. Partial template matching for the “object-
of-interest” extraction should be considered. Finally, the algorithm could be further evolved to use more complex models/templates to recognise objects with different perspectives such as 3-D objects. Admittedly, considering all these issues increases the computational complexity significantly and requires more research.

• Using a library of the templates will simulate the human knowledge system. In this scenario, the objects in the image are compared with the templates in the library. The library is a database of already extracted high level knowledge about the objects, such as their templates. The database could be extended to include each newly detected object. However, examining the region combinations for all the existing templates in the library increases the computational complexity significantly, rendering a large library impossible. Therefore it requires more research on exploring the scene for objects through a database of high level knowledge defined in a library.

• Developing the semantic region analysis, such as extracting the sky, sea, land, etc., in conjunction with the developed “object-of-interest” extraction algorithm and the database of high level knowledge can result in full semantic segmentation which divides the scene into semantic objects and regions. This will be a significant step towards full scene segmentation, analysis and understanding.

• The video tracking algorithm can be further developed to track multiple objects and extract different low and high level knowledge about the objects, such as their size, colour, motion direction, collision and appearance and disappearance in a video shot. Shadow analysis and removal will allow for correct extraction of the objects and should be added to the extraction algorithm. A large amount of computational complexity of the video segmentation is related to global motion estimation. Therefore, more research on estimating the global motion or deleting/replacing its requirement from the algorithm is necessary.

• Many other parameters, including the number of classes in the spatial segmentation, are determined by the user. For example, continuity $\beta$, the parameters
and convergence threshold in the optimisation method, the smoothness coefficient at different resolutions in spatial segmentation and the coefficients in the objective function of the video region classifications such as $\alpha$, $z_t$ and $z_x$ should be entered manually. In the “object-of-interest” extraction some parameters such as the acceptable distance between the combination of regions and the template of the “object-of-interest” should be entered. Values of some of these parameters are less critical, and they can be determined automatically by data examination procedures. However, some others such as the number of classes or number of objects are very critical, and their automatic settings are very difficult. Determining the parameters and thresholds are generic problems in many image and video processing algorithms, and it is a serious obstacle to implementing a fully automatic and unsupervised algorithm. Therefore more research about automatic parameter estimation is necessary.

- Multi-view scene presentation which records the scene with multiple cameras is going to be more popular. Therefore finding a way to use information from different cameras for segmentation of the scene is necessary, and this will improve the segmentation accuracy.

As expected, image/video semantic segmentation and “object-of-interest” extraction is a very challenging task, and it needs low level processing as well as high level knowledge. Finally, although segmentation is a challenge and still requires further research, we are convinced that this thesis has contributed some novel ideas towards the fully automated semantic segmentation goal.
Bibliography


[49] A. Caro, M. Duran, P. G. Rodriguez, T. Antequera, and R. Palacios, “Mathematical morphology on mri for the determination of iberian ham fat content,” in proc. 8th Iberoamerican Congress on Progress in Pattern Recog-


