Multiple-feature Object-based Segmentation of Video Sequences

R. Piroddi

Submitted for the Degree of Doctor of Philosophy from the University of Surrey

Centre for Vision, Speech and Signal Processing
School of Electronics and Physical Sciences
University of Surrey
Guildford, Surrey GU2 7XH, U.K.

February 2004

© R. Piroddi 2004
Ai miei genitori
Considerate la vostra semenza:
fatti non foste a viver come bruti,
ma per seguire virtù e canoscenza".

Dante Alighieri, *Divina Commedia*
Inferno, Canto XXVI
Versi 118-120
Summary

Emerging multimedia applications and services require efficient and flexible coding (MPEG-4) and description (MPEG-7) of visual information. Object-based representations of visual information obtained by scene segmentation are particularly well-suited to this purpose. In this work, the segmentation of video sequences is addressed using a combination of features, such as motion, texture and colour. First, the Recursive Shortest Spanning Tree (RSST) is considered as a baseline segmentation tool and is adapted to perform single-feature segmentation using different visual cues. A novel motion-based RSST segmentation algorithm that incorporates multiple motion features into a single cost function is presented. Effective texture segmentation is achieved by a novel scheme relying on mathematical morphology operators. This approach is further extended to become applicable to colour texture segmentation. Second, multiple-feature segmentation of video sequences emerges as a major focus of this work. The RSST has been employed in order to perform simultaneous multiple-feature segmentation of video sequences in a hierarchical fashion. The presented work demonstrates that the performance of this approach rapidly degrades as the dimensionality of the feature space increases. To overcome this problem, a novel two-stage architecture for object-based segmentation is presented. The first stage locates perceptually meaningful objects using a hierarchy of single-feature segmentation processes. The second stage refines the boundaries of located objects using a suitable combination of features and a set of appropriate rules. This model is further simplified by minimizing the number of required sequence-dependent parameters and also by minimizing the number of inputs to the rule-based part of the algorithm. A comparative evaluation with state-of-the-art competing algorithms is favourable, demonstrating that the proposed architecture is capable of achieving accurate, meaningful and consistent segmentations which are intuitively correct and have good correspondence with a human viewer's notion of the decomposition of a natural scene to its constituent objects.

Key words: Image Segmentation, Video Segmentation, Spatio-temporal Segmentation, Object-based Segmentation, Multiple-feature segmentation, Dynamic Scene Analysis, Graph-based Representations, Motion Estimation, Texture Analysis, Multiple-feature Fusion, Psychovisual Perception, Video Coding
Acknowledgements

I would like to acknowledge my supervisors Dr T. Vlachos and Professor J. Kittler for their guidance during my PhD.
I would like to acknowledge EPSRC for funding this project.
I would like to thank Damitha, Evan and Rob for their friendship.
I would like to thank my family and Kai for their never-ending support.
# Contents

1 Introduction ................................. 1
   1.1 Digital Convergence and Emerging Multimedia Applications .................... 1
   1.2 Object-based Segmentation of Visual Information ................................. 6
   1.3 Aims of This Project ......................................................... 9
   1.4 Overview of This Thesis ..................................................... 10
   1.5 Achievements ................................................................. 11

2 Object-based Segmentation of Dynamic Sequences .................................. 13
   2.1 Introduction ................................................................. 13
   2.2 Definition of Segmentation ............................................... 14
   2.3 Baseline Segmentation Techniques .......................................... 16
      2.3.1 Discontinuity-based Techniques ..................................... 17
      2.3.2 Continuity-based Techniques ........................................ 18
      2.3.3 Classification-based Techniques ..................................... 20
   2.4 Object-based Segmentation Techniques .......................................... 25
      2.4.1 Motion-based Approaches ............................................. 28
      2.4.2 Spatio-temporal Segmentation ........................................ 33
      2.4.3 Multiple-feature Segmentation ....................................... 41
   2.5 Conclusions ................................................................. 43

3 Methodology .................................... 45
   3.1 Multiple-feature Combination ............................................. 45
   3.2 Baseline Segmentation Technique ........................................... 46
      3.2.1 Graph-based Segmentation Techniques ............................. 47
      3.2.2 Recursive Shortest Spanning Tree ................................... 48
3.2.3 Application of Shortest Spanning Tree to Image Analysis ........................................ 49
3.2.4 Shortest Spanning Tree Segmentation ........................................................................ 50
3.3 Characteristics of SST ................................................................................................... 51
3.4 General Formulation of RSST and Application to Multiple-feature Segmentation ........ 52
3.4.1 RSST Performance Issues ......................................................................................... 56
3.5 Relationship to Human Visual Perception ................................................................. 61
3.5.1 Psycho-visual Perspective .......................................................................................... 61
3.6 Conclusions .................................................................................................................. 69

4 Feature Extraction ............................................................................................................ 71
4.1 Introduction ..................................................................................................................... 71
4.2 Motion Segmentation .................................................................................................... 72
4.2.1 Role of Motion .......................................................................................................... 72
4.2.2 Robust Optic Flow Estimation .................................................................................... 76
4.2.3 Motion Segmentation using the RSST ....................................................................... 83
4.2.4 Derivation of Motion Features from Translational Optic Flow Components ............ 89
4.2.5 Estimation of Number of Moving Objects ................................................................. 99
4.2.6 Background/foreground Labelling and Object Characterisation .............................. 101
4.2.7 Consistency Check .................................................................................................... 105
4.2.8 Role of Outliers ........................................................................................................ 111
4.2.9 Conclusions on Motion .............................................................................................. 117
4.3 Texture Segmentation .................................................................................................... 118
4.3.1 Role of Texture .......................................................................................................... 118
4.3.2 Statistical Texture Description .................................................................................... 120
4.3.3 Conclusions on Statistical Texture Segmentation .................................................... 127
4.3.4 Morphological Texture Description .......................................................................... 128
4.3.5 Conclusions on Morphological Texture Segmentation ........................................... 148
4.4 Colour Segmentation .................................................................................................... 148
4.4.1 Colour Segmentation with the RSST ....................................................................... 150
4.4.2 Colour Spaces ............................................................................................................ 150
4.4.3 Experiments ............................................................................................................... 152
4.4.4 Conclusions on Colour .............................................................................................. 154
4.5 Conclusions .................................................................................................................. 157
## 5 Feature Integration

5.1 Introduction .................................................. 159

5.2 Hierarchical Multiple-feature Segmentation ................. 160

5.2.1 Link Weight Options .................................... 161

5.2.2 Multiple-feature Combination .......................... 161

5.2.3 Linear Weighting of Intensity and Motion .......... 167

5.2.4 Using Texture as a Reliability Measure ........... 176

5.2.5 Comments on Hierarchical Multiple-feature Segmentation ... 183

5.3 Hybrid Hierarchical Architecture for Moving Object Segmentation .... 187

5.3.1 Proposed Architecture .................................. 191

5.3.2 Building a Feature Hierarchy .......................... 191

5.3.3 Boundary Refinement .................................... 193

5.3.4 Experiments with Hybrid Hierarchical Architecture .... 196

5.3.5 Conclusions on the Hybrid Hierarchical Architecture .... 197

5.4 Incorporating Further Aspects of Psycho-visual Perception ....... 200

5.4.1 Perceptual Parallel Paradigm .......................... 202

5.4.2 Experimental Results and Comparison .............. 206

5.5 Conclusions on Multiple-feature Segmentation ............... 212

## 6 Performance Evaluation

6.1 Introduction .................................................. 213

6.2 Choice of Benchmark Algorithms ................................ 215

6.3 Evaluation using Ground Truth ................................ 216

6.3.1 The Misclassification Penalty (DP) ....................... 216

6.3.2 Shape Penalty (DS) ....................................... 218

6.3.3 Motion Penalty (DM) ...................................... 220

6.4 Evaluation without Ground Truth ............................ 221

6.4.1 Intra-frame Colour Metrics ............................... 221

6.4.2 Inter-frame Colour Metrics ............................... 224

6.4.3 Motion Metric .............................................. 227

6.5 Quantitative Evaluation ...................................... 228

6.5.1 Evaluation with Ground Truth ........................... 230

6.5.2 Evaluation without Ground Truth ....................... 231

6.6 Qualitative Evaluation ....................................... 234

6.7 Conclusions .................................................. 239
Contents

7 Conclusions 245
  7.1 Summary 245
  7.2 Contributions 249
  7.3 Future Work 252

A Test Sequences 257

B Performance Evaluation Metrics 263

C Ground Truth Sequences 265
List of Figures

2.1 Simplified representation of steps required to achieve segmentation (adapted from [1]) ................................................................. 14
2.2 Classification of still image segmentation methods. .................. 16
2.3 State-of-the-art object-based segmentation techniques (simplified from [2]) ........................................................................ 27
2.4 Difference in the structure of the architecture between (a) hierarchical and (b) parallel spatio-temporal methods. .................. 34

3.1 Schematic representation of the elements of a graph $G = (V, E)$ ................................................................. 49
3.2 Example of one-to-one mapping of an 0 image onto a planar weighted graph. Each pixel in an image is mapped onto a vertex in the graph and links are established between 4-neighbour pixels. The vertex weight is the 0 of each pixel. The link weight is a 0 measure between linked vertices (in this example the absolute difference of the intensity of each pixel/vertex) ................................................................................ 50
3.3 Building a SST from a graph. In an image of $N$ pixels, its SST can be built in $(N-1)$ iterations. At each iteration, the two most similar vertices (that do not form a circle) are joined. The links belonging to the SST are shown as full lines. The links that do not qualify for the SST are shown as dotted lines ............................................................................................... 51
3.4 Flowchart of RSST algorithm ................................................................. 53
3.5 Hierarchical representation of the information. The RSST algorithm allows the user to choose the detail of the segmentation. In this case, the sequence is represented with the respective edge maps at 10000 regions in (a) and (b), 5000 regions in (c) and (d) and 1500 regions in (e) and (f). 58
3.6 Assessment of algorithmic performance: response of the algorithm (noise produced as output, computed as PSNR in dB) when different levels of (a) Gaussian additive noise, (b) blurring or (c) the two sources combined are corrupting the image. It is possible to notice a critical level of noise beyond which the algorithm breaks down .......................................................... 60
3.7 LGN acts as a bridge between the eyes and the visual cortex. Simplified diagram from [3] ................................. 62
List of Figures

3.8 Location of functional areas of the visual cortex. Adapted from http://www.biol.susx.ac.uk/home/GeorgeMather/. ........................................ 62

3.9 Schematic structure of functional areas of the visual cortex and their connections. ................................................................. 63

3.10 *Dorsal* and *ventral pathways* stem from areas V1 and V2. They have complementary functionalities. Simplified diagram from [4]. .................. 65

3.11 Temporal (T), parietal (P) and occipital (O) lobes locations. The lobes are interconnected. Simplified diagram, originally found in [5]. .............. 66

4.1 Differences in the influence of non-robust (first row) and robust (second row) estimators. .................................................. 79

4.2 Example of horizontal optic flow component. The translational motion flow is obtained in (a) using LS method and in (b) using RE method. Segmentation using the RSST framework for 500 regions in (c) in the case of LS method and in (d) in the case of RE method. ............... 83

4.3 Segmentation of dense motion field in (a) using a LS method and in (b) using a RE method. Both segmentations contain 100 segments. ............. 85

4.4 Results of the segmentation using RSST on the basis of translational components of optic flow for test sequences *Renata* (first column), *Mobile and Calendar* (second column) and *Garden* (third column). .................. 87

4.5 MSE of segmented motion fields for test sequences: (a) *Renata*, (b) *Mobile and Calendar* and (c) *Garden*. ........................................ 88

4.6 Optic flow of a rotating object, taken from [6]. ........................................ 90

4.7 Schematic representation of rigid 2-D rotation of points $P_1$ and $P_2$ aligned horizontally. ............................................................. 90

4.8 Optic flow generated by uniform rotation. ........................................ 92

4.9 Optic flow generated by uniform divergence. ..................................... 92

4.10 MSE of translational optic flow components: comparison of performance of the cost function of Equation A (translational features) and of the cost function of Equation B (multiple motion features). Test sequences: in (a) *Renata*, in (b) *Mobile and Calendar* and in (c) *Garden*. .................. 96

4.11 Comparison of performance of the cost function of Equation A (translational features) and of the cost function of Equation B (multiple motion features). $E_A$: in (a) *Renata*, in (c) *Mobile and Calendar* and in (e) *Garden*. $E_A$: in (b) *Renata*, in (d) *Mobile and Calendar* and in (f) *Garden*. ................ 97

4.12 Moving object segmentation obtained with the use of RSST: using the cost function of Equation A (translational features only) and using the cost function of Equation B (multiple motion features). .......... 98
4.13 2-D histograms calculated on the basis of translational optic flow components (horizontal and vertical displacements). Test sequences: in (a) Renata, in (b) Mobile and Calendar and in (c) Garden. 100

4.14 Typical histogram of (a) curl and (b) divergence calculated from translational components of optic flow for the test sequence Mobile and Calendar. 102

4.15 Layered representation of test sequence Garden with labels for different layers (first row) and borders of layers superimposed on original frames (second row). 104

4.16 Background/foreground labelling using: (a) translational optic flow components, (b) acceleration and (c) rotational features of the optic flow. 106

4.17 Characterisation of moving objects using multiple motion features. 107

4.18 Size of the segmented moving object against the average size for frames 18 to 48 (a) and zoom of size of segmented and projected moving object against the average object size (b). 111

4.19 Moving object segmentation obtained with conventional labelling (first column) and using the projections (second column). 112

4.20 Example of the influence of outliers on the segmentation. In (a) map of outliers (in black) and in (b) their position in relation to the original frame (in white). In (c) moving object segmentation without taking into consideration the motion outliers (c) and in (d) with the use of a rule for the inclusion and/or discarding of outliers. 114

4.21 Technique for labelling uncovered and occluded areas, according to [7]. 115

4.22 Definition of areas of search (occlusion and uncovering areas) (a) and rule for inclusion and exclusion of connected components of outliers (b). 116

4.23 Segmentation with the RSST of test sequence Renata. In (a) using variance as the only textural feature and in (b) combining variance and grey-level intensity. 123

4.24 Use of co-occurrence matrices: in (a) contrast feature map and in (b) segmentation obtained with the RSST and using contrast as textural feature. 124

4.25 Use of edge density. In (a) example edge gradient feature map and in (b) segmentation obtained with the RSST and using edge gradient as textural feature. In (c) edge density feature map and in (d) segmentation obtained with RSST using edge density as textural feature. 125

4.26 Mask for detection of edges between different homogeneous regions and/or between textured and homogeneous regions. 126

4.27 Map of edges calculated using the Sobel filter: in (a) with the edges representing high contrast between homogeneous regions and between homogeneous and textured regions (in lighter grey) and in (b) the same mask where the edges representing higher contrast have been omitted. 127
4.28 Texture segmentation obtained in (a) using an edge density measure inside a given window. In (b) boundaries of textured regions superimposed onto the original frame. .......................... 128

4.29 Results of Canny edge detection for sequences: (a) Renata, (b) Mobile and Calendar and (c) Garden. ................................................................. 132

4.30 Results of morphological closing before and after the filling of the holes (the holes are indicated in light grey) respectively for (a) and (b) Renata, (c) and (d) Mobile and Calendar and (e) and (f) Garden. ........................................... 134

4.31 Outlines and spatial support for textured areas of the image frames respectively for: (a) and (b) Renata, (c) and (d) Mobile and Calendar and (e) and (f) Garden. Outlines and spatial support are indicated either in black or white for visualisation purposes. .......................................................... 135

4.32 Examples of the use of morphological filters in order to segment structuring textural elements of same shape but different size. ...................... 138

4.33 Examples of the use of morphological filters to segment directional structuring elements. .............................................................. 139

4.34 Examples of the use of morphological filters to segment structuring textural elements in regular textures. ................................. 140

4.35 Examples of the use of morphological filters to segment structuring textural elements in irregular textures. ............................................. 141

4.36 Examples of the use of morphological filters to separate homogeneous and textured regions in an image. ........................................... 141

4.37 Examples of the use of morphological filters to segment different texture samples. .............................................................. 142

4.38 Comparison of performance between the conventional RSST segmentation and the modified segmentation using strategies 1 and 2. Test sequences: (a) Renata, (b) Mobile and Calendar and (c) Garden. ........ 146

4.39 Comparison of performance between (a) the conventional RSST segmentation and the modified segmentation using (b) strategy 1 and (c) strategy 2. All the segmentations contain 4000 segments. ........ 147

4.40 Relationship between RGB and \( Y'CB'CR \) colour spaces (adapted from [8]). 151

4.41 Colour texture maps for test sequences Renata, Mobile and Calendar and Garden using (a) \( RG'B' \) and (b) \( Y'CB'CR \) colour spaces. Refer to Table 4.7 for interpretation of the colour codes corresponding to colour texture mixtures. ................................. 155

4.42 Results of colour segmentation using colour texture description for test sequences, Mobile and Calendar and Garden using (a) \( RG'B' \) and (b) \( Y'CB'CR \) colour spaces. The segmentations contain 4000 segments. .... 156

5.1 Distance metrics used as cost function (associated to links in the graph) for preliminary experiments with grey-level intensity. ................................. 162
5.2 Results obtained for different distance metrics used as the cost function (associated to links in the graph) for experiments with grey-level intensity. 162

5.3 Qualitative comparison of performance using distance metrics in (a) \( |x| \), in (b) \( x^2 \), in (c) \( |x|^{1/2} \) and in (d) \( \exp|x| \). The variable \( x \) is a measure of error. 163

5.4 Combination of intensity and motion features in (a) linear and (b) non-linear (b) fashion. 165

5.5 Qualitative comparison of performance using (a) linear and and (b) non-linear combination of intensity and motion. 165

5.6 The introduction of multiple features does not improve the PSNR of the segmentation. Here, the comparison is between using intensity only and using intensity and motion. 166

5.7 Comparison between normalisation strategies. Example of test sequence Renata, with 2000 regions segmentation. 167

5.8 Comparison of performance between the use of cost function 5.6 and of the set of parameters \((0.5,0.5,0.5)\) and set of parameters \((0.5,0.25,0.25)\). Test sequences: (a) Renata, (b) Mobile and Calendar and (c) Garden. 168

5.9 Comparison of performance with different weights for intensity and motion features. Test sequences: Renata (a), Mobile and Calendar (b) and Garden (c). 169

5.10 Qualitative comparison of performance with different weights for intensity and motion features. Settings: \((1,1,1)\) (in (a), (b) and (c)), \((0.8,0.1,0.1)\) (in (d), (e) and (f)), \((0.6,0.2,0.2)\) (in (g), (h) and (i)), \((0.4,0.3,0.3)\) (in (j), (k) and (l)) and \((0.2,0.4,0.4)\) (m,n and o). The number of regions is 1000. 170

5.11 Qualitative comparison of performance with different weights for intensity and motion features. Settings: \((0.5,0.25,0.25)\) (in (a), (b) and (c)), \((0.8,0.2,0.2)\) (in (d), (e) and (f)), \((0.6,0.4,0.4)\) (in (g), (h) and (i)), \((0.4,0.6,0.6)\) (in (j), (k) and (l)) and \((0.2,0.8,0.8)\) (in (m), (n) and (o)). The number of regions is 1000. 171

5.12 Comparison of performance for different values of the parameter of influence of outliers \( w \) (it varies from 2 to 8) and different settings of parameters \( (\alpha,\beta,\gamma) \). The test sequence is Renata. 173

5.13 Qualitative comparison of performance for \( (\alpha,\beta,\gamma) = (1,1,1) \) (in all four examples) and \( w = 2 \) (in (b) and (d)). In (b), an effective improvement in the accuracy of the motion object boundaries is achieved. The opposite is true for case (d). The segmentations consist of 4000 segments. Images have been equalised for better visibility of region boundaries. 174

5.14 Performance evaluation of weighting when taking acceleration into account, compared with the worst case settings for the parameters \((\alpha,\beta,\gamma)\) in Equation 5.6. Test sequence: Garden. 176
5.15 Qualitative comparison of performance for \((\alpha, \beta, \gamma) = (0.8, 0.2, 0.2)\), using outlier information (left column) and using acceleration (right column). In (b), an effective improvement in the accuracy of the motion object boundaries is achieved. The opposite is true for case (d). The number of regions is 500 in (a) and (b) and 16000 in (c) and (d). ... 177

5.16 Qualitative representation of the kind of profile used by [9] ......... 178

5.17 Schematic of textured area, transition area and homogeneous area as defined by masks \(M_1\) and \(M_2\) .................. 179

5.18 Map of distances \(d_1\) (in (a), (b) and (c)) and \(d_2\) (in (d), (e) and (f)). The mask \(M_1\) is obtained using a disk structuring element of radius \(r_1 = 7\) pixels and the mask \(M_2\) is obtained using a disk structuring element of radius \(r_2 = 3\) pixels. Test sequences: in (a) and (d) Renata, in (b) and (e) Mobile and Calendar, in (c) and (f) Garden. ....... 180

5.19 Diagram of the weighting profile \(D = \frac{d_1}{d_1 + d_2}\). \(D = 1\) where the mask \(M_2\) shows the textured areas of support and \(D = 0\) where the mask \(M_1\) shows the homogeneous areas of support. This function is used as a weight to modulate the influence of motion and intensity features. ... 181

5.20 Definition of distances \(d_1\) and \(d_2\) .................. 181

5.21 Comparison between the cost function of Equation 5.6 \((\alpha = 1, \beta = 1\) and \(\gamma = 1\)) and different sets of parameters for the profile of motion \(D\) and intensity \(C\). Test sequences: in (a) Renata, in (b) Mobile and Calendar and in (c) Garden. ....... 184

5.22 Comparison between the use of the cost function of Equation 5.11 \((\alpha = 0.8, \beta = 1\) and \(\gamma = 1\)) and that of Equation 5.6. Test sequences: in (a) Renata, in (b) Mobile and Calendar and in (c) Garden. ....... 185

5.23 Effect of introducing a textural profile function with \((\alpha, \beta, \gamma) = (1, 1, 1)\). Comparison between cost functions expressed by Equations 5.6 (left column) and 5.11 (right column). The use of texture yields an improvement in the case of (a) and (b), but the results are not always acceptable, as shown in (c) and (d). The number of regions is 350 for (a) and (b) and 500 for (c) and (d). ... 186

5.24 Diagram of proposed architecture and outputs of layers of processing of a dynamic scene. ....... 192

5.25 Schematic description of the notation for the definition of opposite segment. 194

5.26 Outputs of each layer of the hybrid hierarchical architecture, using test sequences Garden and Mobile and Calendar. The segments are indicated by the corresponding highlighted boundaries. ....... 198

5.27 Results obtained with the hybrid hierarchical architecture are consistent through time. Here are examples of Garden and Mobile and Calendar at frames 10, 20, 30, 40 and 49. ....... 199
5.28 Flowchart of parallel architecture for object-based segmentation which mimics the human perceptual visual pathways ................................................................. 203

5.29 Simplified schematic of homogeneous segment overlapping multiple motion object boundaries ................................................................................................. 206

5.30 Simplified schematic of textured segment overlapping multiple motion object boundaries. Average motion features for each part of the segment belonging to different moving objects are calculated without taking outliers into consideration ......................................................... 207

5.31 Comparison of the foreground-background object-based segmentation obtained with the use of the hybrid hierarchical architecture and the parallel architecture. For multiple moving object segmentation, the interested reader is referred to Figures 6.7, 6.8 and 6.9 ........................................................ 210

5.32 Labels corresponding to the foreground objects shown in Figure 5.31. Boundaries between different labelled regions are superimposed to facilitate visual inspection. For the corresponding multiple moving object segmentation, the interested reader is referred to Figures 6.7, 6.8 and 6.9. 211

6.1 Definition of turning angle function (TAF), adapted from [10] .............................. 218

6.2 Definition of boundary coordinates \((r, \theta)\) and regular sub-sampling of ground truth \(g_i\) object boundary and segmented \(s_i\) object boundary. ........................................ 219

6.3 Definition of rotation angle between two successive pointing vectors on the newly sub-sampled object boundary ........................................................................ 219

6.4 Definition of just inside and just outside areas as appears in [11] (a) and definition of the areas to evaluate the contrast at the object boundary in the work here presented (b) .................................................................................. 222

6.5 Diagram of the weighting function \(f(r)\) used to compute contrast on the object boundary which depends on the distance \(r\) from the boundary. 223

6.6 Original frame of test sequence (a), possible segmented object (b) and evaluation of contrast at the boundaries of the segmented object (c). 225

6.7 Comparison of the segmentation of frames 10, 20, 30 and 40 in test sequence Renata for the segmentations obtained with the use of [12] in (a), (b), (c) and (d); with the use of [13] in (e), (f), (g) and (h); with the use of [14] in (i), (j), (k) and (l) and with the use of the method here proposed in (m), (n), (o) and (p). ...................................................... 240

6.8 Comparison of the segmentation of frames 10, 20, 30 and 40 in test sequence Mobile and Calendar for the segmentations obtained with the use of [12] in (a), (b), (c) and (d); with the use of [13] in (e), (f), (g) and (h); with the use of [14] in (i), (j), (k) and (l) and with the use of the method here proposed in (m), (n), (o) and (p). ...................................................... 241
6.9 Comparison of the segmentation of frames 10, 20, 30 and 40 in test sequence Garden for the segmentations obtained with the use of [12] in (a), (b), (c) and (d); with the use of [13] in (e), (f), (g) and (h); with the use of [14] in (i), (j), (k) and (l) and with the use of the method here proposed in (m), (n), (o) and (p) ........................................ 242

A.1 4:2:2 sampling pattern .................................................. 259
A.2 Interlaced scanning method [8] ........................................ 259
A.3 Example of original fields for test sequence Renata .......... 261
A.4 Example of original fields for test sequence Mobile and Calendar .......... 261
A.5 Example of original fields for test sequence Garden .......... 261

C.1 Example of ground truth for test sequences Renata, Mobile and Calendar and Garden .......... 266
List of Tables

4.1 Parameters for the LS experiment. ........................................... 81
4.2 Parameters for the RE experiment. ........................................... 82
4.3 Motion segmentation with multiple features ................................ 95
4.4 Error in segmented and projected objects ................................. 110
4.5 PSNR in experiments with colour segmentation ..................... 152
4.6 PSNR in experiments with 1 texture segmentation ................. 153
4.7 Colour codes for texture description in Figure 4.41 ............... 154

5.1 Parameter setting for experimentation with Renata, Mobile and Calendar and Garden. ............................................................... 197

6.1 Results of evaluation with the use of ground truth for test sequence Renata. ................................................................. 231
6.2 Results of evaluation with the use of ground truth for test sequence Mobile and Calendar. ......................................................... 231
6.3 Results of evaluation with the use of ground truth for test sequence Garden. ................................................................. 232
6.4 Summarised results of performance evaluation gathered from intra-frame colour dissimilarity metrics. ................................. 233
6.5 Results of evaluation without the use of ground truth for test sequence Renata. ................................................................. 234
6.6 Results of evaluation without the use of ground truth for test sequence Mobile and Calendar. ......................................................... 235
6.7 Results of evaluation without the use of ground truth for test sequence Garden. ................................................................. 236
6.8 Number of homogeneous and textured regions automatically selected by the parallel method. ......................................................... 238
6.9 Number of segments composing spatial part of segmentation in the compared methods. ......................................................... 239

A.1 Technical information about test sequence Renata. .................... 258
A.2 Technical information about test sequence Mobile and Calendar. . . . 258
A.3 Technical information about test sequence Garden. . . . . . . . . . . . . 258
**List of Abbreviations**

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-D</td>
<td>one dimension(al)</td>
</tr>
<tr>
<td>2-D</td>
<td>two dimension(al)</td>
</tr>
<tr>
<td>3-D</td>
<td>three dimension(al)</td>
</tr>
<tr>
<td>AIT</td>
<td>Anterior InferoTemporal</td>
</tr>
<tr>
<td>DS</td>
<td>Description Scheme</td>
</tr>
<tr>
<td>DVD</td>
<td>Digital Versatile Disk</td>
</tr>
<tr>
<td>HDTV</td>
<td>High Definition TeleVision</td>
</tr>
<tr>
<td>HVS</td>
<td>Human Visual System</td>
</tr>
<tr>
<td>IT</td>
<td>InferoTemporal</td>
</tr>
<tr>
<td>ISDN</td>
<td>Integrated Services Digital Network</td>
</tr>
<tr>
<td>LGN</td>
<td>Lateral Geniculate Nucleus</td>
</tr>
<tr>
<td>LS</td>
<td>Least Squares</td>
</tr>
<tr>
<td>MM</td>
<td>Mathematical Morphology</td>
</tr>
<tr>
<td>MPEG</td>
<td>Moving Picture Experts Group</td>
</tr>
<tr>
<td>MRF</td>
<td>Markov Random Field</td>
</tr>
<tr>
<td>MSE</td>
<td>Mean Squared Error</td>
</tr>
<tr>
<td>MST</td>
<td>Medial Superior Temporal</td>
</tr>
<tr>
<td>MT</td>
<td>Medial Temporal</td>
</tr>
<tr>
<td>PIT</td>
<td>Posterior InferoTemporal</td>
</tr>
<tr>
<td>PSNR</td>
<td>Peak Signal to Noise Ratio</td>
</tr>
<tr>
<td>RAG</td>
<td>Region Adjacency Graph</td>
</tr>
<tr>
<td>RE</td>
<td>Robust Estimation</td>
</tr>
<tr>
<td>RSST</td>
<td>Recursive Shortest Spanning Tree</td>
</tr>
<tr>
<td>SST</td>
<td>Shortest Spanning Tree</td>
</tr>
<tr>
<td>TAF</td>
<td>Turning Angle Function</td>
</tr>
<tr>
<td>VIP</td>
<td>Ventral IntraParietal</td>
</tr>
</tbody>
</table>
List of Abbreviations
List of Publications


List of Publications
Chapter 1

Introduction

"The matrix is everywhere; it is all around us."

This is a key point in the analysis of the content of reality which surrounds the characters of the influential film The Matrix, by the Wackowski brothers (1999). Lawrence Fishburne's persona, Morpheus is indicating a veil of virtual reality created by a digital architect that permeates every aspect of the human activities.

The Matrix has been defined the first film to truly represent the twenty-first century. Perhaps most of its influence is due to the force with which the film plot instigates in the viewer's mind the suspicion that the pervasiveness and ubiquity of the digital reality is something that we are already witnessing in our days. It is a phenomenon called Digital convergence and its revolution is changing the way we learn, work, play, communicate and entertain ourselves.

1.1 Digital Convergence and Emerging Multimedia Applications

The phenomenon of the convergence of digital telecommunications, digital computing and digital media is called Digital Convergence [15]. During the last ten years, one has witnessed the development and mutual influence of these digital technologies and the growth of shared contents.
Chapter 1. Introduction

Historically, the first manifestation of digital convergence has been the World Wide Web, at the beginning of the nineties. The Web brought together telecommunications (the physical network), digital computing (servers and browsing software) and digital media (simple text and images).

We are now already at the second phase of the digital convergence: thanks to the improvements in the telecommunication technologies and infrastructure, it is possible to deliver rich media (like audio, video and animation) on the Web, on the PC and on the TV set. This is also the age of the mobile Internet service and the growth of wireless technology.

From a more limited point of view, digital convergence is indicated as the merging of entertainment sources like television, sound systems and photography to a digital format or the transmission of data, audio and video over the same telecommunication channels [15]. Even from this more limited definition, it is clear that digital convergence is about interaction, common interface and sharing of information.

Video is one of the rich media that characterises the second phase of the digital convergence and it is possibly the most challenging medium to integrate, because of the sheer amount of information to communicate and process.

Digital video comes in four big categories: broadcast video, digital video in sharing format (e.g. DVD), streaming video and video-conferencing. All these manifestations of video have their particular content, quality associated and target users.

Cable and satellite broadcasters already are trying to offer a digital service which overcomes the shortcomings of conventional television. In general a conventional television service presents a fixed schedule that cannot accommodate individual schedules and interests, the content of the offer is fixed for all the audience. There is little interactivity with the audience, no integration with other services like e-mailing or web-browsing. Now it is possible to acquire digital set-top boxes where it is possible to browse a menu of different service via remote control, order video-on demand, and switch to alternative services. High Definition Television (HDTV) offers a high quality digital video, although uptake has been slow. There are also ways to provide more flexibility in the fruition of the service with Personal Video Recorders, which can be accessed through
a user-friendly menu (e.g. TiVo).

Film production is very quickly going from the analogue into the digital realm, with digital recording and editing. Also of great interest is the creation of digital archives to preserve the cultural film heritage. Not to mention the possibility of having digital video libraries which in turn need facilities for easy and effective browsing, indexing and retrieval.

Digital Versatile Disk (DVD) is another very popular format of digital video media, which can be used on a PC: in contrast to broadcast video, it provides a more individual experience. It has some in-built interactivity in the presence of a menu and different options of browsing the content and added features by the producers. The most interesting and creative frontiers for the DVD users are related to software that enables DVD to be treated like other files on the PC: software for ripping, creating and editing DVDs are examples.

Streaming video offers a totally different fruition experience from the broadcast television. The quality is lower, but it is fully integrated into the Internet and can be used on demand to suit individual or niche exigencies.

Video-conferencing is another technology that has attracted interest. It is mainly of interest for business and performed with dedicated ISDN phone lines. One of the major complaints regarding this technology is the relative complexity of operating the software needed.

From the examples of emerging services and applications it is possible to see that the proliferation of digital format of media and new multimedia applications requires a flexible, user-friendly representation, description and coding of visual information. As it is possible to see from the examples of application technologies and services already available or in the process of development, the emphasis is focused onto getting more individual access and control over the content.

If one sees digital convergence merely as a technical phenomenon of integration of different instruments and media, it is easy to see that a generic platform is needed for the use and sharing and communication of information through telecommunication infrastructures, computers and media instruments (like digital cameras or digital video
cameras for example). This is the realm for the development of common formats, interfaces and ultimately a common representation of the information.

If, on the other hand, one sees the digital convergence from a more humanistic point of view, as a phenomenon of empowerment of the user/viewer, who assumes more control over the instrument, then it is possible to generalise all the requirements highlighted above in the need of interactivity, simplicity of use of the media, format and interface, the development of a new language to express video in terms in which a human can relate to.

This is a fundamental turning point: not just new integrated technologies change our world and our perception of it. This is also a two-way process: the way in which the human user/practitioner/technologist relates to digital video material also is changing. The way the digital media are expected to be experienced is changing. It is not acceptable to represent videos as a set of binary numbers coded according to a certain format. If the content of the video has to be addressed, summarised and modified, then the video must contain a description of its meaningful components, from a human point of view. In [16] this process is described in comparison with text editing: one would like to use video material with the same ease one uses text. With text it is possible to write words from primitives (the alphanumeric symbols), from the words one extracts concepts. It is possible to change the position of words, edit them, erase and replace them to create a totally new composition and communicate new ideas.

The challenge to video technology nowadays is to generate appropriate words to describe the content.

An example of the evolution in the needs and the treatment of the multimedia signal is given by the evolution of the 0 activities of the Moving Picture Experts Group (MPEG).

The MPEG standardisation work dealt, at least at the beginning, with issues related to coding of audio, video and joint audio-video compression [17].

Digital broadcasters experience problems with video compression, since some content of video (like sports) can be particularly difficult to code. Therefore, it is difficult to assure the same quality of experience throughout the range of content offer. This is not acceptable for paying viewers.
1.1. Digital Convergence and Emerging Multimedia Applications

assure the same quality of experience throughout the range of content offer. This is not acceptable for paying viewers.

The first two products of the MPEG group, MPEG-1 standardisation [18], which is the standard that supports Video CD and MP3 (audio), and MPEG-2 [19], which supports Digital television set-top boxes and DVD, were hugely successful and are de-facto standards.

Following the activities of the MPEG standardisation effort, it is possible to acquire an idea of the evolution in terms of complexity of integration of technologies and the evolution of features and coordination required by new multimedia services. Basically, it gives an idea of how the expectations of digital technology delivery have changed for practitioners, producers and users.

MPEG-1 [18] and MPEG-2 [19] systems dealt with issues of overall architecture, multiplexing and synchronisation. In addition to this, MPEG-4 [20] encompassed interactive scene description, content description and programmability. Before MPEG-4, audiovisual data was mainly a series of bits. Only the decoding of these bits would give some information about what the data was and what the user could do with it. MPEG-4 introduces the idea of multimedia objects and defines how to represent, compress, transmit and manipulate them. The emphasis on the access and meaningful description of multimedia content is accentuated on MPEG-7.

In MPEG-7 [21] the emphasis on the content of the material is even stronger, bringing new challenges of definition of languages for the description of the content, the binary representation of the content and the means of delivery of the content in visual format or joint audio-visual format [22, 23]. With MPEG-7 descriptors and description schemes, it is possible to extract the information about the data without the need of actually performing the decoding of the data [22]. A Descriptor defines the syntax and the semantics of an elementary feature. It can deal with low-level features such as colour, texture, shape, motion, audio energy and spectrum as well as high-level features such as the title or the author. The Descriptors are structured and related within a common framework based on Description Schemes (DSs). The Description Schemes are organised into different functional areas: basic elements, content management, content...
description, navigation and access, content organisation and user interaction [23].

However, the handling of video content would not be possible without the definition and extraction of the component objects.

1.2 Object-based Segmentation of Visual Information

Video objects have been introduced by MPEG-4 [20]. In the general description of the principal features of MPEG-4 standard, the general characteristics desired for satisfying the exigencies posed by new rich media and their integration into a varied technological network are also presented. The standard aims at addressing the needs of evolving digital television, interactive graphics applications (synthetic content) and interactive multimedia (e.g. World Wide Web) [20]. MPEG-4 aims at providing answers and tools for authors of multimedia content, multimedia service providers and users alike. For authors, it enables the production of content that, thanks to standardised description, can have more re-usability and flexibility in management, sharing and editing. For the service providers it grants unified representation of the information, definition of interface, ways of transmitting and synchronising data and a descriptor of quality of service. For the users it focuses on providing more interactivity and with effective compression delivers multimedia contents at low bit-rates and on mobile networks.

Object-based representation and description of visual information is particularly suited for all these purposes, since it endeavours to describe a dynamic scene in terms of its semantic components, the so-called objects, instead of low-level visual primitives, such as intensity, colour, texture, motion, depth, shape. Multimedia objects have been defined by MPEG-4 as atomic units of audio-visual content, each carrying individual and autonomous meaning (e.g. a still image of a background, a talking head without background, a voice associated to a person, a music background). The objects can be organised in a hierarchical way described by a scene graph in such a way so they form compound objects and the whole scene as the root of the graph. MPEG-4 describes the typology of objects, the way to represent their features (spatial, temporal and spatio-temporal), to modify them (by moving them, scaling them rotating them), to compress them and to transmit them.
1.2. Object-based Segmentation of Visual Information

The potential advantages of object-based video descriptions have attracted the attention of a wider audience, especially from the IT and multimedia industries and this explains the interest in the development of MPEG-4 as major international standardisation effort for multimedia coding. This put the basis for MPEG-7, a standardisation proposal for multimedia object description.

It is not difficult to imagine a future in which multimedia material will be accessed on the basis of its content and described in terms of semantic objects composing the scene. Object-based descriptions of natural scenes are becoming increasingly important in multimedia applications [16], in fact they allow the authoring, manipulation, editing and coding of digital video material in a far more creative, intuitive, efficient and user-friendly manner.

Content-based representation of visual information is therefore increasingly attracting plenty of attention in numerous other applications. Citing just a few: tracking, robot vision and database browsing.

Although MPEG-4 places a lot of emphasis on audio-visual objects, it does not provide a way to generate them from generic audio-visual material.

To date object-based description of multimedia material is obtained in a controlled environment, such as a studio, or is generated with the aid of a computer. For a more general and useful use, object-based description needs to be extended to frame-based videos of generic origin and content. Such descriptions are achieved primarily by means of video scene segmentation.

In the language of image and video processing, segmentation is the process of translation from low-level visual primitives to semantically meaningful objects for the human viewer. To date it remains the most fundamental step of the analysis of a dynamic scene and also the most elusive and difficult to achieve, since for general applications it is an ill-posed problem that cannot rely on prior knowledge or high-level modelling.

Object based segmentation is the first and most important step towards object-based representation of the information. It is a challenging task because objects are not made of homogeneous regions; therefore different sources of information are to be fused together.
Chapter 1. Introduction

Segmentation for object-based applications has to be performed in a spatial and temporal domain taking into account motion information on a diversity of spatial features like grey-level intensity, colour, texture and edge information.

Segmentation can be performed in a supervised and unsupervised way. In the context of new multimedia applications it is desirable that a certain level of automation is guaranteed as default and the flexibility of user interaction has also to be allowed and it is indeed a desirable feature in the interactive applications.

This constitutes an elusive problem, to which no current system or algorithm can provide a satisfactory solution.

The importance of segmentation is reinforced in recent coding scheme attempts. Segmentation is fundamental to the content description in MPEG-7. Content description refers to information that can be perceived in the content. Two different aspects are analysed in MPEG-7 DS: the structural and the conceptual aspects, which deal with the semantics. From the point of view of the structure of the content, the organisation of the data relies on the concept of segment. The segment DS describes the result of a spatial, temporal or spatio-temporal partitioning of the data. It can describe a hierarchical decomposition resulting in a segment tree. The segment relation DS describes additional relationships among segments and allows the creation of graphs [23].

The segment DS forms the base type of different specialised segment types such as audio segments, video segments, audio-visual segments, moving regions and still regions. The segment DS can be used to describe segments that are not necessarily connected, but composed of several non connected components. Connectivity refers to both spatial and temporal domains [23].

The segment DS may be subdivided into sub-segments and thus may form a hierarchy or a tree. The hierarchy decomposition is useful to design efficient search strategies. It also allows the description to be scalable [23].

In MPEG-7, certain features like the definition of a feature, are fairly standardised and therefore they will not offer any scope for further research. There are other aspects that are only specified at the interface and in their normative aspect. These will be
1.3. Aims of This Project

This project deals with object-based representation of visual information from generic content video sequences. It aims at developing an effective and accurate method to extract meaningful video objects by means of segmentation of video sequences. Since segmentation is a low-level operation, the sources of information useful for the identification and used for segmentation are to be low level features like luminance, chrominance, texture, shape, edges and motion.

Single-feature segmentation methods, that is to say segmentation methods based on only one of the features (e.g. colour segmentation and motion segmentation) are well-known operations of low-level image processing and still remain an open problem for the research community, since a generic method able to perform well in a variety of situations has not been developed yet. Moreover, the use of only one source of information does not help towards the generation of a meaningful video object, which will be a complex entity made up of different attributes coherently related to each other. This is where multiple feature segmentation is required.

The main focus of the project is therefore to explore the dynamic relationship between different features, cues, or sources of information, to establish a hierarchy between them, to seek how they influence each other. Perceptually plausible ways of combining cues are explored and incorporated in an overall system of hierarchical description and automatic segmentation of arbitrary content sequences.

The goal of the research is to exploit a combination of significant perceptual features.
in order to extract objects from segmented frames or groups of frames. Therefore the intention is not to provide any fundamental improvement to individual feature identification techniques, but to exploit the best current practice. The emphasis is therefore placed on determining useful combinations of different features.

However, since digital video content is a challenging medium to compress and integrate due to the sheer volume of information contained in a generic scene, it is of main importance to devise methods to achieve extraction of features and meaningful single and multiple feature segmentation in a simple and effective way.

Moreover, this project focuses on video sequences of generic content. Most of the available methods are developed taking into consideration a specific application. However, the amount of digital material available and the interest of practitioners and users in addressing their content and manipulation, means that a generic procedure to extract objects with reasonable accuracy and easiness for a general-purpose objective is needed.

In this project, the automatic segmentation of video objects is addressed, as well as a method of rendering the flexibility and interactivity of the use of the segmentation tool.

1.4 Overview of This Thesis

This thesis is structured as follows.

In Chapter 2, Object-based segmentation is presented as a particular case of segmentation as well as the need for multiple-feature segmentation. A literature review of available object-based segmentation methods is presented.

In Chapter 3, the baseline segmentation technique of choice for this project, the Recursive Shortest Spanning Tree (RSST) is presented. Moreover some elements of psychological perception are presented as they are of interest in the design of a procedure for fusion of different features.

In Chapter 4, multiple features, such as motion, texture, and colour information are taken into consideration. For each of them, a criterion to extract useful information for segmentation and highlight its relation with the other cues is specified. Moreover
1.5 Achievements

In order to achieve a segmentation (or decomposition) of a moving scene several aspects have to be addressed simultaneously.

- **Mathematical modelling:** A mathematical model to guide the process of the segmentation has to be defined. In this work an approach to dynamic scene segmentation based on the graph-theoretical concept of Recursive Shortest Spanning Tree (RSST) has been presented. In the past, RSST has been used as a tool for effective video coding for its capability to describe the image in a hierarchical framework, which is able to capture image details at various semantic levels and is well tuned to the characteristics of human vision. Exploiting the flexibility and ductility of this tool, I extended its area of application from grey-level intensity static image segmentation to motion segmentation, colour segmentation and texture segmentation, as well as edge-detection and employed it for multiple-feature segmentation.

- **Feature combination:** The automatic definition of visual objects relies mainly on motion features. However the introduction of numerous other visual cues
(such as colour, texture, shape, depth) is required for an accurate and meaningful extraction of such objects. The combination of such diverse and heterogeneous sources of information is an open issue for research, giving the intrinsic difficulty in the appraisal of relative merit of different features and their dynamic relationship. This issue has been investigated in this thesis, leading to the formulation of a novel architecture for feature combination and segmentation.

- **Integrated approach:** The need to take into consideration different visual features and their diversity in dynamic range, statistical distribution and overall appearance in the image or sequence of images suggests that a more integrated approach to segmentation is needed. Extensive review of segmentation techniques is needed in order to appreciate which segmentation tool is more suitable for segmentation of each visual feature. However, it is clear that only one segmentation model will not suffice in the quest for a definition of a meaningful object, which is composed by a set of inhomogeneous single feature components (or segments). The challenge in this work has been to implement efficient ways of performing segmentation for different features and developing an integrated set of segmentation techniques, their hierarchy of interaction and their location in a modular scheme of behaviour.

- **Psycho-visual perception modelling:** The design of a global architecture can take enormous advantage from the latest psycho-visual perception findings and from inspiration provided by the psychology and brain research community. Although the level of complexity and sophistication expressed by the human visual system cannot be emulated, the implementation of general biological ideas can be an effective guideline in the design of algorithms for the decomposition of a scene in semantically meaningful component for the human viewer.

Finally, one note about the citation from *The Matrix* that opens this introduction. It can be seen as a negative slant towards the excessive power of technology permeating our lives. However, the message from the film is the necessity of acquiring control over the digital content provided by modern technology and this is definitely one of the ultimate reasons that support the development of object-based representation of
1.5. Achievements
digital information.
Chapter 1. Introduction
Chapter 2

Object-based Segmentation of Dynamic Sequences

2.1 Introduction

In this chapter, a review of state-of-the-art object-based segmentation techniques is given. Object-based segmentation requires an integrated architecture for segmentation in order to deal with different sources of information and integrate them.

In order to present the fundamental issues related to this problem, a gradual approach is followed. First, a review of still-image segmentation techniques is given. These techniques are the building blocks to integrated object-based segmentation architectures. However, the straightforward extension of these techniques is unsatisfactory.

Second, a review of object-based segmentation techniques (also frequently indicated as moving-object segmentation) is presented, differentiating them on the basis of two different characteristics: the structure of the architecture, which depends on the segmentation criteria used to determine the segmentation, and the number of sources of information, which are also called features or cues, used to define the segments.
Chapter 2. Object-based Segmentation of Dynamic Sequences

2.2 Definition of Segmentation

Image segmentation is the basic low level process that poses the basis for any image analysis and scene understanding. It is defined as the operation of decomposition of an image into a set of disjoint (non-overlapping) and comprehensive (complete) regions (or partitions or segments). Pixels belonging to the same region or segment are similar according to a given criterion of homogeneity (or similarity). The simplest example of image segmentation is given by grey-level image segmentation, but homogeneity criteria can be defined for a variety of features, like colour, texture, motion, shape, histogram, and so on. Given a specific feature, several algorithms (baseline segmentation techniques) can be used in order to decide how to partition the image. Therefore, segmentation algorithms can be classified taking into consideration the feature on which they are operating. According to the feature space in which the segmentation is performed, a segmentation method is chosen to perform the segmentation (which is equivalent to a decision taken according to given criteria). This is summarised in Figure 2.1.

In the case of digital video the decision about the way of partitioning the scene can be taken in 1-D temporal space, 2-D spatial space or 3-D spatio-temporal space.
2.2. Definition of Segmentation

1-D temporal video segmentation is often referred to as video segmentation or shot segmentation. The aim of this segmentation is to divide a long sequence into its composing scenes, detecting the different shots. Each video can be seen as the union of different scenes that are homogeneous in content and represented objects. The 2-D spatial segmentation is the segmentation of each frame as an image on its own. This is analogous to still image segmentation. In the case of videos, still image segmentation is not useful, if not used in conjunction with some temporal information. Therefore 3-D spatio-temporal segmentation of videos combines spatial information related to a single frame in the video (intra-frame information) with temporal information related to the relative motion between the current frame and its neighbours (inter-frame information). 3-D spatio-temporal segmentation is performed on homogeneous video scenes, that is to say that the temporal video segmentation into homogeneous scenes has been already performed or it is taken as a prerequisite. The 3-D spatio-temporal decision domain is the domain of object-based segmentation. Object-based segmentation aims to decompose a scene into its semantically meaningful components. For example in a scene from a football match, object-based segmentation would decompose into a (stationary) background of the pitch and into moving objects like the footballers and ball. Since object-based segmentation aims to decompose the scene into a set of non-overlapping and complete objects, it is related to canonical image or scene segmentation. However, the latter segmentation methods divide the image into regions that are homogeneous according to a selected feature (grey-level intensity, colour, texture, motion and so on). In object-based segmentation, objects are composed of a set of regions or segments that are inhomogeneous. An object may be composed of regions of different colours or textures, therefore the challenge is in the definition of a meaningful semantic object in terms of low-level primitives (motion, colour, texture) and their dynamic interrelationship. When talking about relationships, it is necessary to note that not all the low-level features have the same importance in the definition of an object, but some combination of them reinforces the consistency of others.

There is no baseline segmentation technique that is able to deal with all kinds of image and feature space. On the contrary, the baseline segmentation technique has to be chosen taking into account the specificity of the feature space that is being dealt with.
Chapter 2. Object-based Segmentation of Dynamic Sequences

Figure 2.2: Classification of still image segmentation methods.

and the kind of application that is targeted.

For multimedia services, specific characteristics have to be taken into account. Digital video data are very large. Therefore, the algorithms should be efficient in terms of computational complexity and storage requirements. The algorithms need to be able to cope with generic scenes. This means that the amount of information available is limited to low-level features, like intensity, colour, texture, shape, size or motion. There is no a priori knowledge that can be exploited in order to build higher level models of behaviour because the content of the scene and the composition of the objects in the scene is not known in advance.

2.3 Baseline Segmentation Techniques

Segmentation is the process by which an image is subdivided into homogeneous and disjoint regions representing the objects composing a scene. A number of algorithms have been developed especially for the simplest case, that of a monochrome image, where the only available information is grey level intensity and spatial relationships. All these baseline techniques can be exploited to perform segmentation on different features and at the decision level in object-based segmentation techniques. Available techniques either rely on detection of discontinuity or on continuity of grey level
2.3. Baseline Segmentation Techniques

Baseline segmentation techniques are normally grouped into three categories [24, 25, 26, 27, 28, 29]: Discontinuity-based techniques or Edge-based techniques, Continuity-based techniques or Region-based techniques and Classification-based techniques.

A schematic classification of still image segmentation methods is presented in Figure 2.2.

2.3.1 Discontinuity-based Techniques

Edges characterise object boundaries and they are points on the image plane, where the gradient is maximum. Therefore these methods rely on the computation of the gradient of the intensity or the second order derivative [24, 26]. Methods based on gradients work best when the grey-level transition is quite abrupt. If the transition regions becomes wider, it is more advantageous to apply a second order derivative, implemented as a Laplacian operator. This operator is very sensitive to noise therefore it is better to use its zero-crossings to locate the edges. Boundaries can be found by tracing connected edges. Contour following algorithms trace boundaries by ordering successive edge points. In edge relaxation techniques, edge properties are considered in the context of their neighbourhood: if sufficient evidence of a border is provided, local edge strength increases and vice-versa [6]. Whenever any additional information is available, it can be used to define an optimality criterion to use for border detection by means of graph-searching or dynamic programming. In this way, the border detection problem is transformed into the search for the optimal path in a weighted graph. Global processing in order to find directions of boundaries can be obtained by the Hough Transform [25]. The Hough Transform can be used if the image consists of objects of known shape and size. It can detect lines or curves. If the analytic expression of objects is not known, than the generalised Hough transform can be used using a parametric definition of the objects. Another approach to border detection is represented by the use of active contour models (snakes) [30].

Edge-based techniques, depending on edge detectors, are usually sensitive to noise, spurious intensities or non uniform illumination. Edge-linking operations do not ensure
complete boundaries for a given region and region determination from partial boundaries is very complex. Moreover, they are not suitable for multidimensional analysis.

2.3.2 Continuity-based Techniques

Region-based segmentation techniques realise a complete segmentation of the image into a set of non-overlapping segments [24]. The segments are obtained using a specific homogeneity criterion in the selected feature space. The homogeneity criterion is evaluated on a local basis. The segments are built by means of region merging, splitting or a combination of both. In the region growing technique the image is divided into atomic regions of constant grey level. The growth of a region $R_i$ starts from a seed $k_i$. Neighbouring pixels $p$ are then sequentially considered to be merged into $R_i$ on the basis of a homogeneity criterion for the growing region $R_i$. Region growing [31] techniques are further divided into three groups. Single linkage region-growing is very sensitive to noise as one pixel leakage causes unwanted merging of regions. In hybrid-linkage region-growing, the homogeneity function is calculated using the properties of a small neighbourhood. In centroid-linkage region-growing, each margin is decided upon by comparing each pixel with its neighbouring regions [28].

The main disadvantage of region growing is the dependence on the selection of seeds. Single linkage image segmentation schemes are attractive for their simplicity. They do, however, have a problem with leakage, because it takes one erroneous element leaking from one region to a neighbouring one to cause the regions to merge. Hybrid linkage techniques seek to assign a property vector to each pixel where the property vector depends on the $K \times K$ neighbourhood of the pixel. In the centroid linkage techniques, the image is scanned in some predetermined way. The value of the pixel is compared with the mean of an already existing but not necessarily completed neighbouring segment. If its value and the mean value of the segment are close enough, then the pixel is added and the mean of the segment is updated. The centroid and the hybrid linkage schemes can be used together to take advantage of their strengths. The strength of the hybrid linkage is that boundaries are placed in a spatially accurate way. The strength of the centroid linkage is its ability to place boundaries in weak gradient areas.
2.3. Baseline Segmentation Techniques

Another popular technique is *split and merge*. The split method for the segmentation begins with the entire image as the initial segment. They typically use pyramid representations of the image and a popular approach is the quad-tree segmentation. In the pyramid each segment successively splits each current segment into quarters, if the segment is not homogeneous enough. When adjacent segments are similar enough, they are merged. Because segments are successively divided into quarters, the boundaries produced by this technique tend to be blocky and/or artificial.

The *watershed* technique finds inspiration from the concept of a catchment basin [32]. Catchment basins represent the regions of segmented image. The first watershed segmentation approach starts with finding the downstream path from each pixel of the image to local minima of the image surface altitude. A basin is defined as the set of pixels from which the respective downstream paths end up in the same altitude minimum. Second, each grey level minimum represents one catchments basin and the strategy is to start filling the basin from the bottom. The *watershed* technique takes advantage of the gradient magnitude information. In fact the edges of the regions should be the points of local maximal gradient magnitude. These points can be seen as the boundaries of watershed regions: everything on one side of the boundary belongs to one region and everything on the other side to a different one, forming different pools contained between watersheds. One way to find the regions is filling the pools from the bottom. On a grey-level image composed of 255 levels, one starts with the pixels of intensity 0. These form the basis for the segmentation. Then one adds the pixels labelled with 1. If they are next to an existing region they are labelled with the same label, otherwise they start a new region. The process repeats for every level of intensity [32]. In practical terms, the gradient magnitude information is used. The watershed technique is very computationally expensive and sensitive to noise as it employs gradient information. Moreover it is difficult to generalise for multiple cue segmentation.

Structural models using homogeneity criteria rely on the idea of subdividing the image into pieces that can be described by parameters. These parameters can be part of a regression plane or a higher order approximating surface, parameters of a grey-level function or representing the description of a texture by means of computer graphics methods. The portion of the image can be regular and then there are methods...
like piecewise approximation, pyramids and quad-trees. Structural models can also be content-adapted and then there are methods like facet-model, region-growing, split and merge. Piecewise image approximation subdivides the image domain into zones, then tries to approximate the gray value function in that zone with the analytical function of a spatial surface. Parameters of these functions are determined with least square error. As models, planes or higher order surfaces are often used. Another class of structural techniques is represented by surface approximating techniques. These are techniques close to region growing which try to approximate a region of given or arbitrary shape with a plane or another higher order plane. These techniques rely on graph-based representations like region adjacency graphs (RAGs) and build a hierarchical description of the information by means of regular or irregular pyramids.

*Pyramids* are data structures which can be used as tools for image analysis and therefore segmentation. A pyramidal representation of an image is a sequence of replicas of the original image [33]. In an adaptive pyramid [33], which can be useful for segmentation, the surviving cell from one level to another is not chosen regularly, but on the intensity of interesting features. Moreover, while in a regular pyramid the number of children for each parent node are the same, in an adaptive one this number changes. One representative technique for segmentation included in this group is Recursive Shortest Spanning Tree algorithm [34]. In this algorithm the image is mapped into a RAG and a homogeneity criterion is defined. At each iteration the two most similar regions are merged together and the subsequent region is approximated by a plane or plateau of average values. Then the RAG is updated to take into account this change and the procedure is iterated until the shortest spanning tree of the image is built. This technique is close to single linkage region growing with the advantage of not requiring any seed pixels to start. Moreover it allows the formation of regions of arbitrary shape as opposed to split and merge techniques.

### 2.3.3 Classification-based Techniques

Classification-based segmentation techniques are derived from statistical pattern recognition. Grey level values of pixels or other attributes like texture features or motion
can be regarded as features in a feature vector. Each vector representing a pixel or a region has to be assigned to a specific class. Thresholding can be seen as a special case of classification-based technique in which the pixels are classified into two classes: objects and background. Subsequently, classification-based methods can be divided into supervised and unsupervised methods [35].

**Thresholding**

*Thresholding* exploits global information about an image or its parts [25]. This information is represented by a histogram of image features. In the case of grey-level thresholding, a brightness constant or threshold is determined in order to segment objects and background. Only under rare circumstances thresholding can be successfully employed using one threshold for the whole image (global thresholding). Alternatively, segmentation using variable thresholds (adaptive thresholding), chosen in order to reflect some local property of the images, can be achieved. The effectiveness of this technique is related to an appropriate choice of threshold. If some property of the image is known a priori, the task of threshold selection is simplified. Automatic threshold selection methods rely on the analysis of the shape of the histogram in order to detect the valleys of the histogram. Histogram mode seeking and optimal thresholding are popular methods. A survey of other threshold detection techniques can be found in [25]. Thresholding can also be extended to colour or multi-band images (multi-spectral thresholding) and on hierarchical data structures. Thresholding has the advantage of being simple and fast to implement. However, it ignores spatial relationships between features, its effectiveness is limited to simple shapes and well-separated objects.

**Supervised Classification**

If supervised methods are used for classification, a priori knowledge is necessary in order to build a training set from which the classifier can learn how to assign labels to pixels, i.e., to decide which classes a pixel belongs to. Thus, the segmentation process divides the image into regions corresponding to known classes. Human interaction is needed in supervised classification in the form of a training set. It is possible to use trained
classification methods if the number of classes (regions) $R_i$ for ($i = 1, \ldots, n$) and the statistical features are known a priori. Each pixel is classified as an element of one of the known $R_i$, depending on where it is mapped in a suitably chosen $m$ - dimensional feature space. This technique for image segmentation uses the measurement space clustering process to define a partition in the measurement space. Then each pixel is assigned the label of the cell in the measurement space partition to which it belongs. The image segments are defined as the connected components of the pixels having the same label. The decision about the assignment of pixels or objects to a class is taken by a machine called a classifier. The classifier can be a deterministic or a probabilistic machine. It is deterministic if a discrimination function exists to define the discrimination hypersurface. However most classifiers are probabilistic machines. If it is possible to model the probability of one pixel or object to belong to a certain class depending on characteristic features, then the decision is taken on a minimum error criterion.

Cluster Analysis

On the contrary, in unsupervised classification there is no need for a training set but the pixels are clustered with no reference to labels [36]. Unsupervised classification is also termed as cluster analysis. When the location in the feature space and the number $n$ of the classes (regions) are not known a-priori cluster analysis should be used. A popular algorithm is the K-means algorithm [37]. Usually, a sequential cluster analysis is performed: $k$ initial seeds are chosen, each representing a region. The distances are computed between the elements of the feature vector associated with each pixel and the $k$ seeds. If the minimum of these distances is bigger than a given threshold, a new region is initialised. Otherwise the pixel is classified as belonging to one of the current regions. Critical points are the choice of the number of seeds and the choice of features. Statistical analysis is usually developed alongside the algorithm to determine the intrinsic number of clusters in the data. Additionally, a reliable and unambiguous stopping criterion is required.

The accuracy of the image segmentation process depends directly on how well the
objects of interest in the image separate into distinct measurement space clusters. More information about unsupervised as well as supervised learning can be found in [35].

Relaxation

Relaxation segmentation techniques derive from the relaxation labelling procedures. Relaxation labelling techniques incorporate semantic information in the form of models of interrelationships between objects composing a scene. Object properties are described by unary relations and object interrelationships are described by n-ary relations. The final scene must be consistent and favour the more probable interpretation of the scene. Relaxation techniques can be classified into probabilistic or fuzzy relaxation. Here, a description of probabilistic relaxation is given. The fuzzy relaxation is similar. In order to use relaxation for segmentation, the scene must be specified by a set of $n$ objects or pixels, $f_1, f_2, \ldots, f_n$, to be classified into a set of $m$ classes $C_1, C_2, \ldots, C_m$. For each pair of class assignments $f_i \in C_j$ and $f_h \in C_k$, a quantitative measure of compatibility $C(i, j, h, k)$ of these assignments is specified, i.e. the objects have relationships between each other. For example, a positive value of $C(i, j, h, k)$ means that the assignment is compatible (it therefore leads to a coherent interpretation of the scene), a negative value means incompatibility and a null value indicates indifference. Indicating with $p_{i,j}$ the probability of $f_i \in C_j$, if $p_{h,k}$ is high and $C(i, j, h, k)$ is positive, then $p_{i,j}$ will increase, but, if $C(i, j, h, k)$ is negative, $p_{i,j}$ will decrease. The relaxation technique aims to find the configuration of labels assigned to an object that maximises the overall probability of a certain label configuration.

Relaxation techniques have the advantage of taking into consideration contextual information in the form of compatibility conditions and are well suited for parallel implementation. Relaxation labelling (probabilistic relaxation) allows only transitions to states of lower energy (and therefore higher confidence of the configuration). These often result in the optimisation process getting stuck in local minima. Relaxation labelling is used in the extension of segmentation techniques like region growing in order to allow higher-level knowledge to be incorporated for scene understanding. Semantic
region growing techniques [6] incorporate context into region merging using a priori knowledge about relations amongst adjacent regions.

Markov Random Field based Approaches

The use of Markov Random Fields (MRFs) for modelling purposes is also very frequent [38]. This is a branch of probability theory which is exploited because it allows the description of spatial neighbourhood or contextual information. This enables the model to incorporate a priori probability of the occurrence of a particular pattern. MRFs theory provides a basis for modelling contextual constraints in visual processing and interpretation. When used with optimisation principles, it enables the development of optimal vision algorithms. MRFs provide a convenient and consistent way of modelling context of spatially correlated features. MRF theory indicates how to model the a priori probability of context dependent patterns, such as a class of textures or an arrangement of object features.

MRFs are used for labelling tasks in computer vision. In probability terms, contextual constraints may be expressed in terms of probabilities. In the presence of context, labels are mutually dependent. How to make a global inference using local information becomes a non-trivial task and this is where MRFs provide a mathematical foundation.

MRF models offer a solution to the problem of representing local information and relationships which are not taken into account by the statistical techniques presented in Section 2.3.3. However, they are computationally involved and practically tractable for segmentation only if the number of labels (segments) is known in advance and is small [39].

Neural Network based Approaches

Pattern recognition is an application area of neural networks [6]. A neural network is a combination of elementary processors, called neurons. Neurons take $n$ inputs and produce one output, as a function of the weighted sum of inputs. Associated to each neuron is a transfer function, which practically generates the output. Each neuron
is wired to other neurons, the structure of the wiring is said to reproduce the neural connections in the human brain. The simplest neural network has only one layer of neurons. Most neural networks are composed of a number of layers and the connections between layers are established by paradigms such as feed-forward. Such networks need a phase of training of the network, obtained using a training set from which the network learns which are the right outputs for a given set of inputs. Self-organising networks avoid the use of training sets.

Fuzzy Sets

In all the techniques that have been presented in this section, the classifier needs a crisp boundary to decide whether to assign a pixel or object to one class rather than another. But most problems related to vision are affected by uncertainty regarding the interpretation of an object (Where does a boundary really lie? How round is a round object?). Fuzzy sets have been introduced to handle decisions in the case of imprecise knowledge of an event [40]. A fuzzy set $A$ is defined as: $A = \mu_A(x_i) | x_i, i = 1, 2, \ldots, n$, where $\mu_A(x_i)$ gives a measure of confidence of the belonging of element $x_i$ to the set $A$. The shape of the fuzzy membership can be changed by fuzzy hedges. A hedge and its fuzzy set constitute a single semantic entity called linguistic variable. In an application like segmentation, the question regards the definition of a region to extract, since its boundaries are ill defined. In this situation, it seems advisable not to commit the decision to any hard criterion of decision, but to allow the segment to be a fuzzy subset of an image. The subsets are characterised by the degree of possibility of belonging to a certain class. In this case, the segmentation process is seen as the process of optimisation of an objective function expressed by weighted similarity measures between fuzzy features. This is the structure of c-means clustering algorithms, the most popular application to fuzzy logic to image segmentation.

2.4 Object-based Segmentation Techniques

Object-based segmentation techniques can benefit from different segmentation approaches, but the ones that seem most interesting for multimedia services are the
region-based segmentation techniques [1, 2]. In fact, edge-based segmentation techniques do not provide a segmentation of the image into regions and therefore objects. Classification-based techniques are not used because they rely on a higher-level of information which is generally difficult to model for generic sequences. Moreover, some paradigms on which they rely, like optimisation or MRFs, are computationally very expensive. Object-based segmentation techniques differ in the features that are exploited for the segmentation, and their number and the way they are combined. It is therefore possible to distinguish three categories of techniques: motion-only segmentation techniques, spatio-temporal segmentation techniques and multiple-feature (or cue) segmentation techniques.

Object-based segmentation was first driven by the so-called second generation coding techniques [41], which incorporate some aspects of the human visual system and therefore aim at the segmentation of the scene into meaningful objects. In regard to this application the objects present in the scene are the objects which share the same motion. This is the reason why the first classifications of object-based segmentation techniques by [42] take into consideration only motion-based techniques. The classification is divided into motion estimation, segmentation and simultaneous motion estimation and segmentation. Also in [9] only motion-based segmentation techniques are taken into account and are classified on the basis of the structure of the algorithms into top-down and bottom up approaches. In [2] object-based segmentation techniques are classified into motion-based segmentation techniques and spatio-temporal segmentation techniques, which combine spatial features like intensity or colour with motion information. More recently, a number of methods have been developed, that incorporate a number of different spatial and temporal features or cues and adopt a more integrated approach to object segmentation. On the basis of the feature space adopted to perform the segmentation in this work the approaches to object-based segmentation are classified into motion-based segmentation techniques, spatio-temporal segmentation techniques and multiple-feature (or multiple-cue) segmentation techniques.

A schematic diagram of possible object-based segmentation techniques is presented in Figure 2.3.
2.4. Object-based Segmentation Techniques

Figure 2.3: State-of-the-art object-based segmentation techniques (simplified from [2]).
2.4.1 Motion-based Approaches

Objects of interest in scenes are moving objects, which bear information about the content of the scene. It is therefore understandable that earlier object-based segmentation methods focused on motion as the only cue of interest. There are different kinds of classification of motion-based methods. In [2] motion-based methods are classified on the basis of the motion model used. Thus, motion-based methods can be distinguished between methods that employ two dimensional or three dimensional motion models. Among the two-dimensional motion models, there are techniques that employ optic flow or change detection information. Among the three-dimensional models, the order of the parameter model makes a difference. Therefore, it is possible to have six-parameter models, eight-parameter models or twelve-parameter models.

In [42], motion-based methods are classified into ones that utilise intensity information, in the form of spatio-temporal gradient information. These methods are called direct methods. Other methods use motion information provided by optic flow and they try to fit it into a parametric model. These approaches are called optic flow approaches. Finally a third category attempts to address the problem of motion segmentation and estimation simultaneously.

In [9], motion-based segmentation techniques are divided into top-down and bottom-up approaches. Top-down approaches exploit global information and iteratively subdivide the sequence into its components, while bottom-up approaches exploit local information and build the components from the lowest level up exploiting local similarity.

In the continuation, motion segmentation techniques are broadly classified into top-down and bottom-up.

Top-Down Approaches

The top-down approaches rely on the outlier detection-rejection paradigm [43]. Top-down approaches iteratively extract a dominant motion. Pixels that do not comply with this motion are then selected for further segmentation. The major advantages of top-down techniques are the exploitation of global information and the low computational
complexity. These methods approximately identify the objects of interest in a scene avoiding over-segmentation, but the extracted objects are inaccurate, especially at the boundaries. In [12], the outliers are defined as points of violation of the assumptions of constancy of brightness or the smoothness constraint of motion, which cause gross measurement errors in the optic flow. These outlier points are detected for further processing.

The objects are sequentially extracted by iteratively determining the successive dominant motion characteristics. Pixels complying with the current dominant motion are assumed to comprise one object. The other pixels are seen as outliers. This is the approach proposed in [44]. One problem with this approach is the difficulty to compute a dominant motion in the presence of multiple motions, known as the generalised aperture problem. In [44, 45], this problem is tackled using a pyramid approach: motion analysis begins at a low resolution level of the pyramid. This allows larger image velocities to be estimated. At each successive iteration, the estimated step is performed on the next higher resolution level of the pyramid. The other problem arises from the outliers influence on the computation of the motion characteristics. In this case, the results can be improved with the use of robust estimators [12]. Top-down techniques are characterised by simplicity and low computational cost.

In [44], the dominant motion model is computed taking into account the error between the intensity difference between frames and a parametric motion model. The parameters that minimise the error are taken as the dominant motion. Once the dominant motion is computed, all the pixels that comply with this motion are segmented in order to eliminate them and to iteratively proceed in the motion analysis. Once the two frames are registered according to the computed motion parameters, the segmentation is simplified in the identification of the regions that are stationary or nearly stationary according to a given parameter that accounts for noise. This analysis is done in a multi-resolution way.

In [46], the graph theoretical framework of Normalized Cuts is used for top-down segmentation using a motion similarity metric. The Normalized Cuts criterion selects the two most dissimilar partitions of a set of data represented in a RAG. By applying this
Chapter 2. Object-based Segmentation of Dynamic Sequences

criterion recursively, it is possible to find an \( N \) region moving object segmentation.

Bottom-Up Approaches

Bottom-up approaches rely on a region-merging procedure to identify meaningful regions. After a set of initial regions is defined, these are merged on the basis of some spatio-temporal similarity measure. The three basic steps are: the creation of the initial regions, the definition of a similarity measure and the application of this measure. This is the approach followed in [9]. The spatio-temporal similarity between the regions is expressed as a hypothesis test. The assumption to be tested regards the similarity of neighbouring regions. The merging strategy is one of graph clustering.

Different techniques can be used to generate a set of initial regions. Some authors use each pixel as an initial region. These are guaranteed to be spatio-temporally coherent, but they may be not meaningful nor the motion estimation based on single pixels reliable. The choice of a single pixel as an initial region is followed in [47], where spatio-temporal segmentation based on MRFs is proposed. A relaxation technique is used to minimise an energy function which depends upon the motion estimate and the number of regions in the segmentation. One-pixel regions are also the initialisation set considered in [48]. In this approach, the flow field is partitioned into connected segments of flow vectors. Each segment is consistent with a rigid motion of a roughly planar surface. At a further stage, the segments are merged together testing the hypothesis that they belong to the same rigid object.

Other approaches suggest that simple quad-tree segmentation can be used to generate the first regions. The spatial delimitation of such regions does not reflect the true spatial structures present in the scene. This disadvantage is overcome by using arbitrary shaped initial regions that are spatio-temporally homogeneous. One example of this approach is given by [49] where the evaluation of the optic flow is based on recursively subdividing the image into squared patches of different size. The size is adjusted adaptively and in such a way to preserve the continuity of the motion between different patches. This is done using quad-tree splines on an adaptive square grid. The splines are designed on a hierarchical basis obtained by a pyramidal decomposition: this allows
2.4. Object-based Segmentation Techniques

a coarse-to-fine motion estimation.

The similarity information is based mainly on the temporal information and is given in a parametric way. An eight parameter model is the one chosen in [50], where a MRF approach is proposed and followed by a Bayesian optimisation, as a criterion to find the best segmentation based on motion. Clustering is extensively used in the literature for this kind of segmentation, but it has the disadvantage not to take spatial relationships into account.

In [51], a layered representation of the objects composing the scene is proposed. Such a representation is useful for video coding applications. The aim is a representation of video information that incorporates depth structure of the scene and accumulates the information throughout the sequence, so that occluded segments belonging to the same object can be recognised as part of the same layer. Each layer [52] is formed by three maps: an intensity and texture map, an alpha map containing the information about the relationship with the other layers and a parametric map describing the motion of the region. In order to obtain such a layered description, a framework is proposed that generates hypotheses of temporal coherence, reinforced by constraints on connectivity and size of the regions. In order to achieve this, an iterative framework generates a set of most likely motion parameters, which are pruned by k-means clustering. Regions are assigned one of the available motion labels by hypothesis testing which aims at minimising the overall distortion. Until the termination criterion is met, the regions are updated by splitting the ones that do not meet a certain threshold and a region filter generates new ones to test and assign.

In [53], the motion estimation is achieved using a robust estimator of the Displaced Frame Difference. The motion model employed for the motion segmentation is a 2-D affine model. The aim of the segmentation is to minimise the difference between the parametric and the true optic flow to a minimum fixed threshold. This is achieved by a MRF framework, which requires the minimisation of a potential function which takes into consideration the displaced frame difference and the gradient of intensity as an indication of the goodness of the motion estimates.

In [54], simultaneous motion estimation and segmentation is achieved by optimisation.
Chapter 2. Object-based Segmentation of Dynamic Sequences

The expression for the probability is reduced to a potential function that has to be minimised. The potential function is composed of three terms: one term expresses the accuracy of the estimates and therefore the goodness of the model fitting the data. The second term aims at minimising the divergence of the parametric model to the optic flow. The third term is related to the goodness of the segmentation.

The definition of region similarity is a challenging issue, especially due to the need to incorporate more than temporal information into a similarity measure. One possible approach is the one proposed in [55]. Another MRF technique is proposed. It takes into consideration the motion estimates and the model of the neighbourhood provided by the MRFs. The assumption is that a pixel having the same intensity as its neighbours is likely to belong to that region. A more complex approach is proposed in [56]. In the framework of MRF, maximisation is defined as a series of constraints relating intensity, boundaries and time. In the intensity constraint, information about smoothness of the change of intensity and temporal coherence is included. The boundary constraints regard the neighbourhood to be modelled by MRF, while the time constraints include a spatial coherence term as well. Similarly in [57], constraints are based on the gradient of the intensity and the distribution of edges alongside the motion information, in this MRF approach. In [58], intensity and motion information are fused into a similarity measure used to perform region merging. In contrast, in [59], a graph theoretical clustering scheme is used. This involves colour information together with motion in order to segment a scene.

It is also possible to use a graph theoretical approach to perform spatio-temporal segmentation, as in [60]. Here, a weighted graph on the image plane connects pixels that are in the spatio-temporal neighbourhood. For each pixel, motion profile vectors are defined to express the probability distribution of the image velocity. The distance between motion profiles is used to assign a weight on the graph edges. Normalized cuts are used to obtain the partitions of the spatio-temporal graph.

Bottom-up approaches [9, 53, 14, 61] rely on region-merging paradigms which employ suitable similarity functions. These approaches exploit spatio-temporal coherence of data at a local neighbourhood level. While a better definition of object boundaries is
achieved, errors in the evaluation of features on a local basis are not uncommon.

In [62] a graph-based region merging technique is presented in order to group regions that present similar motion. This is a bottom-up approach to motion-based segmentation based on the utilisation of the Recursive Shortest Spanning Tree (RSST) and of a six-parameter affine motion model.

In [63], the change detection problem, which labels pixels into two categories, background and foreground, is analysed. The inter-frame difference is modelled by a mixture of two zero-mean Laplacian distributions. At first statistical tests to label pixels with negligible error are used. Then, these labelled pixels are used as starting seeds for a fast marching algorithm where the labels are propagated with different velocities. The novelty of the approach resides in the stopping criterion that avoids two overlapping fronts to form.

In [64], another change detection approach is presented. The inter-frame difference is modelled by a mixture of two zero-mean Gaussian distributions. Then, the labelling problem is formulated as a RMF problem where the potential function has to be minimised. The potential function is composed by a spatial homogeneity term, a motion likelihood term and a temporal coherence term.

### 2.4.2 Spatio-temporal Segmentation

Object segmentation techniques may operate in the spatial (intra-frame) and/or temporal (inter-frame) domain. Methods that address the problem of segmentation in both inter and intra frame domains are called spatio-temporal segmentation techniques. The feature spaces used are intensity or colour information at intra-frame level and motion information at inter frame level.

In [1] these methods can be classified into hierarchical and parallel methods, according to the dynamic interrelationship established between the spatial and the temporal information. Hierarchical methods build the object from pixel or coarse regions up to the object level, using a combination of motion and spatial information simultaneously or changing the features and the criteria for information fusion during the process. Parallel methods process spatial and temporal information separately and build alternative
Chapter 2. Object-based Segmentation of Dynamic Sequences

Figure 2.4: Difference in the structure of the architecture between (a) hierarchical and (b) parallel spatio-temporal methods.

representations of the objects using either colour or motion and only at the end of the segmentation process are the two kinds of information fused.

The different architecture presented by hierarchical and parallel spatio-temporal methods is presented in Figure 2.4.

As pointed out in [2], spatio-temporal methods consist of a spatial and temporal part. The temporal part is strictly related to the motion-based techniques presented in the section above, while the spatial part is related to the baseline still image techniques. It is possible to anticipate [1] that region based techniques are the most used techniques for object-based segmentation in comparison to classification or discontinuity-based techniques.
2.4. Object-based Segmentation Techniques

Hierarchical Spatio-temporal Segmentation Techniques

Hierarchical segmentation techniques build a representation of an image into meaningful objects from the bottom-up. They start at pixel level or using an initial coarse partition of the image (usually using spatial information) and then employ a spatio-temporal (or only temporal) homogeneity criterion in order to merge similar regions and progressively build meaningful objects that share coherent spatial and temporal information. Two main points of interest of this class of approaches are: homogeneity criteria and the creation of an initial partition. The similarity criteria can be probabilistic or deterministic, and according to [1] both approaches yield comparable results. When the similarity (or homogeneity) criteria take into account both spatial and temporal information, the task is more complex due to the need to assign different relative importance to features of different nature. The balance has to be found with some weighting criteria. In most cases, the weighing strategy has to be defined ad hoc or supervised. The initial partitions are usually created on the basis of intensity or colour information and generally based on region-growing baseline segmentation techniques.

These methods are usually computationally expensive, when they make use of statistical methods, and have difficulties coping with highly textured scenes. Additionally, objects are over-segmented and therefore supplementary processing, such as tracking, is required for object identification purposes. Another difficulty lies in the need to use weighting functions. These functions are usually developed ad hoc, are content dependent or rely on parameters, which are user-defined. In order to overcome these problems, a number of parallel rule-based approaches have been proposed recently [13].

In [56], motion and intensity information are fused and used for segmentation using a MRF approach. A set of cliques is defined in such a way that the information related to a pixel is connected to the information of its four-neighbours. The energy terms are a sum of three terms, the intensity model, the motion model and the boundary model. The energy is minimised using simulated annealing. Since the simulated annealing is a very computationally expensive optimisation method, the particularity of this method is that the minimisation is achieved incrementally over a long sequence, accumulating evidence from subsequent frames. This method has the advantage of robustly and dynamically
producing a segmentation which has a lower computational cost than other analogous statistical methods. However, it requires a long sequence and the same quality of segmentation is not available for each and every frame in the sequence.

In [65], a split and merge spatio-temporal segmentation is proposed. The segmentation starts with static segmentation based on intensity or colour using k-means clustering. Each of these regions is assigned a parametric affine model found with a matching technique. If the prediction error is lower than a certain threshold the regions are merged on the basis of their motion similarity otherwise they are split.

A similar structure of the algorithm is presented in [66]. The segmentation starts with static segmentation based on intensity or colour and then the static segments are merged on the basis of motion similarity using an affine motion model. The characteristic of this approach is that the labels are assigned upon a region adjacency graph. The similarity function incorporates motion and geometric constraints and it is minimised with simulated annealing. This technique is a region merging only technique.

In [67], an iterative clustering procedure based on affine parametric motion models and region based colour segmentation is presented. The algorithm presents three similar parts. In each part an initial partition is iteratively refined with the use of a cost function indicating the similarity between the data and the model. The algorithm starts with the motion estimation by block matching. Then, the image is divided into $K$ parts (for example if $K = 4$ the image is divided into four quadrants). Each quadrant is assigned a motion label from the ones available from the motion estimation. These parameters are iteratively updated by assigning every pixel the exact vectors and changing the label according to the real value of the pixels belonging to that region. Then colour segmentation is performed. The motion segmentation map is calculated by affine parameter clustering using colour region-based motion vector matching. Each colour region is assigned a label from the $K$ available classes. The affine parameters are iteratively updated using the squared difference between the affine parameters and the exact motion vectors in the regions. This step can also be repeated using the squared difference between the real intensity and the motion compensated intensity.

In [68], the graph theoretic framework of RSST is used to obtain a colour segmentation.
2.4. Object-based Segmentation Techniques

Then the initial colour segmentation is used in order to build a hierarchical representation and segmentation of the video information using motion similarity. This is done in a user supervised way.

In [69], the method starts with a colour segmentation to be refined by motion information. After an initial coarse segmentation, obtained by k-means clustering, the colour segmentation is refined by MRF modelling represented by cliques in a RAG. The minimisation of the potential function aims at minimising an error in the Gaussian modelling of colour information in the static regions. This colour segmentation is used to initialise a motion segmentation that is done with the framework of MRF as presented in [53], but limited to only two labels, foreground and background.

In [14], a joint spatial and temporal similarity function is used as a merging criterion in a watershed framework. The critical part of every watershed segmentation algorithm is the determination of the markers, which work as seed regions for the segmentation process. In this approach the markers are extracted first in the spatial domain and then coherent markers from the temporal point of view are chosen as seeds for the watershed segmentation. The image profiles for the watershed application are derived by a joint spatial and temporal similarity function. The spatial similarity is given by the intensity difference and the motion similarity is given by the motion vector difference. The two terms are composed in a weighted sum modulated by the weight $\alpha$ and $(1 - \alpha)$. Since the segmentation obtained in such a way is fragmented, coherent segments are merged using motion information, in the shape of a change detection mask.

The same concept of segmentation is presented in [70], where a joint spatio-temporal merging criterion is embedded into a hierarchical segmentation paradigm using the graph-theoretical data structure of a tree. This approach also gives emphasis to the maintenance of a good object shape using mathematical morphology filtering tools for simplification of the partition.

In [71], motion and spatial segmentation are performed in parallel, with the particularity that the motion segmentation is refined by the information provided by multiple frames, which give details about the regions uncovered by motion in the image. Once such a motion mask is built, the colour segmentation is mapped onto the mask and
the object boundaries are built including the colour regions whose majority of pixels belong to the motion mask.

In [72], a temporal memory is introduced in the scheme to enforce temporal consistency. Each frame is initially over-segmented using watershed. Each spatial segment obtained by watershed is then assigned an average motion feature and represented as a vertex in a RAG. The object segmentation is then obtained by a labelling scheme, where the regions are classified into two groups (foreground or background) on the basis of a likelihood which is based on the colour and motion similarity and the consistency of the temporal memory.

In [73], an initial partition is created by morphological filtering (basically colour watershed), performed on an initialisation frame, and then projected on the others. This initial partition represents the universe of possible regions. In order to obtain a video object, a decision has to be made using an optimisation scheme. This is achieved by the minimisation of the following cost function \( C(p_i, R_i) \) of assigning the element \( p_i \) to the region \( R_i \):

\[
C(p_i, R_i) = \alpha_1 \text{dist}_i(p_i, R_i) + \alpha_2 \text{dist}_c(p_i, R_i) + \alpha_3 \text{dist}_d(p_i, R_i)
\]  

(2.1)

where \( \text{dist}_i \) is a measure of the similarity in intensity, \( \text{dist}_c \) is a measure of contour complexity and \( \text{dist}_d \) is a measure of deformation of the region from the projected/tracked object. The structure of this cost function is very similar to the one proposed in [14], only the emphasis is placed on the minimisation of the cost of coding the information, with the attention given to contour complexity and region deformation. The motion, on the other hand, is not well exploited, if not for the projection. The three parameters are user-dependent. This approach has been generalised in [74].

Emphasis on allowing user interaction is also placed in [75], where the user can interact specifying features of interest, contours of the objects or object masks. The segmentation is refined with the use of mathematical morphology, while tracking is done automatically.
2.4. Object-based Segmentation Techniques

Parallel Spatio-temporal Segmentation Techniques

Parallel spatio-temporal methods keep the temporal and spatial part of the segmentation separated, performing them in parallel. From both parts of the segmentation process, they derive a hypothesis of an object. The check for consensus of two alternative representations of the object is derived usually by a set of rules or tests. In this way, the problem related to the weighting of different spatio-temporal features is avoided.

In [13], an integrated parallel spatio-temporal segmentation is presented in the context of the analysis model of the European project COST 211. Motion and colour segmentation are performed separately and in parallel. Colour segmentation is obtained by a graph theoretical method called RSST. Motion segmentation is also obtained by RSST, using the motion information obtained by estimation with Block Matching. Auxiliary motion information is obtained by the computation of the change detection mask and by a memory mechanism storing the motion segmentation obtained in previous frames from the one currently analysed. The information is then fused by a rule processor which can be used in two modalities. The first modality refines the boundaries of the change detection mask by projecting the segments obtained by colour segmentation onto it and deciding by majority voting if the segment belongs to the background or to a moving object. In the second modality, the boundaries of different moving objects are obtained with a similar rule applied to each motion segment and then the objects are tracked with the use of the motion information stored in the memory system.

In [76], an integrated region merging system for object-based segmentation and representation is presented with the use of low level features like colour textures and motion. The segmentation and tracking is obtained in a sequence of $N$ frames. The first frame is segmented spatially using colour and texture features. This segmentation is then extended in time for $N-1$ frames by affine matching the spatial segments in the successive frames. In this fashion, tracking is also performed.

In [77], a semi-automatic segmentation and tracking technique is proposed in the compressed domain. The user specifies in the I-Frame the approximated contours of the object of interest. Given this indication, a search zone is defined and the boundaries of
the object are refined using mathematical morphology. Using this initial segmentation, the object is projected onto the following P-Frames using motion information and the boundaries updated by morphological operations like before. The object is tracked until the next I-frame.

In [78], a semi-automatic segmentation and tracking technique is proposed. The object of interest is selected by the user and the object boundaries are found by a watershed segmentation using a joint similarity measure in a predefined uncertainty region.

In [79], the problem of inter-frame and intra-frame is separated. The intra-frame segmentation is achieved in a semi-automatic way. The user roughly specifies a swath around the object of interest, whose refined boundaries are obtained superimposing an over-segmented version of the frame obtained by a watershed segmentation. The process can be iterated until a satisfactory segmentation is obtained. The tracking is then done automatically. The same concept is used in [80], only this time the user also specifies the relative depth of the objects, so that multiple objects can be easily tracked, and the intra-frame segmentation is obtained by seeded region growing.

In [81], the object mask is obtained from the change detection mask on the thresholding of motion vectors. This mask is refined taking into consideration the contrast at the boundaries of the objects with the use of a rule. The adopted rule sets some threshold for acceptance on the basis of the global variance present inside the change detected object.

Edge-based Approaches

Although not very frequent, it is interesting to note some approaches to spatio-temporal segmentation which rely mainly on edge information.

Perhaps the most interesting approach, also from an historical point of view, is that of [82]. This work has been developed for a static background. The edges of the moving objects are extracted using the Canny edge detector, then closed and simplified using binary morphological filters. Once this template of the moving object is extracted, it is tracked by object matching from frame to frame and updated at each frame to
accommodate changes in scale and shape. The video object plane is obtained by raster scan uniting the first and the last edge pixels in the same row. This can actually create large inaccuracies in the extraction of the video object, as it is evident in the published results.

In [83], the problem is to restrict computation and resources only to the part of the frame which offers ambiguous interpretation, at the boundary of the moving object. The approach is to select interesting points in an initialisation frame. These are corners which show a contrast above a certain threshold. A triangular mesh is then fitted onto the frame. Nodes of the triangles are classified into two categories: moving (foreground) and static (background). Triangles whose nodes belong to different categories are selected for further analysis. The analysis consists of the search for the path of maximum contrast between edges which are fitted to the sides of the closest triangles. Once the object is found, it is projected onto the following frames until a new initialisation frame occurs, in this way tracking of the object is achieved.

In [84], edges are extracted in the three separate channels. Edge differences from two consecutive frames are used to classify edges as moving or background. Moving edges are then merged together and dilated in order to form the moving object plane.

### 2.4.3 Multiple-feature Segmentation

There is evidence that simultaneous consideration of a number of features, such as intensity, motion, texture, colour, depth, and so on, is useful towards reinforcing the spatio-temporal coherence of objects. Several approaches have been developed for the fusion of colour and texture for still images [85, 46, 86, 87]. Unfortunately these can be time consuming and/or tractable only if the number of segmentation labels is limited and known in advance. Most recent attempts resort to simple rules and supervised user intervention [88, 89] instead.

In [85], a region growing segmentation method called JSEG is presented. The spatial segmentation is performed directly on images called J-images that altogether represent colour textures, instead of single pixel colours and parametric texture descriptors. The segmentation criterion is based on the minimisation of a variance measure of the colour
textures presented in the image. Once a spatial segmentation of a frame is obtained, this segmentation is projected to the next frame in order to perform some more merging on the basis of colour and motion similarity, as explained in [76]. Tracking is also obtained by projecting the spatial segments forward and checking the super-imposition between two segments in successive frames. If the majority of pixels of two segments in two successive frames coincide, then the segments are said to be tracked.

In [90], a segmentation scheme which makes use of intensity, motion, edge and texture features is proposed as a clustering method which makes use of self-organising feature maps. These are trained with examples of different video topics (head and shoulders, natural scenes, and so on). Features are normalised and motion and intensity are combined in a unique similarity measure where the motion reliability is weighted against the textural measurements like in [88]. The output of the neural network is rather patchy and over segmented, so these preliminary segments are fused using intensity similarity in a region merging fashion. Additionally, an edge linking procedure is also performed. Given all these additional operations to obtain the final segmentation, the overall process appears to be rather cumbersome.

In [89] multidimensional features are used in a supervised clustering method for segmentation. Features used are motion (optic flow), colour triplets, position and texture. The user indicates the desired segmentation of an initial frame. This frame is then used for training and the clustering follows.

In [91], the use of multiple-features is extended from low-level to high-level semantics. This algorithm makes use of low-level information coded into intensity and motion similarity. The number of labels to assign is also known. Each image is divided into blocks and each block is assigned a random label. The problem is therefore to assign the correct configuration of labels. This is done by mapping the image blocks onto a RAG, so that only neighbours can interact. The labelling is an optimisation of the cost function which is represented by the weighted sum of similarities. The weights represent the confidence of each low level cue and they are obtained by high level semantic represented by a priori knowledge of the objects present in the scene.

Semantics are introduced also in [92]. Here, a very interesting hierarchical multiple-
feature technique is proposed. The hierarchy adopted is related to features more than to image partitions. At the bottom of the hierarchy are grey-level partitions, then a higher level is represented by motion partitions, followed by depth partitions and then semantics. The moving object is defined at the semantic level. Each level of the hierarchy is called a layer. Depth partitions are made up of one or more motion partitions and motion partitions are made up of one or more grey-level partitions. Each partition at a higher level must respect the boundaries established at a lower level of the hierarchy.

2.5 Conclusions

From the excursion into the literature of this chapter, the following issues have emerged:

- The literature is quite rich of spatio-temporal methods of object-based segmentation. Most of these methods claim to be flexible enough to accommodate a number of features, but in practical cases only intensity (and more rarely colour) and motion are considered as sources of information. There are, therefore, very few methods designed to act on a multiplicity of features.

- Although the problem is the segmentation of a video object plane, the segmentation obtained at a frame level is quite poor. Object-based segmentation is reduced to over-segmentation on the basis of intra-frame features, therefore a practical simplification of the spatial content of the frame, delegating the task of the production of a video object to motion tracking.

- The definition of a multiple-feature merging criterion is hampered by the presence of too many user-defined parameters. This is resolved by the claim that fully automatic segmentation of a moving sequence is premature and therefore resorting to user-defined parameters or object planes and/or contours.

Although user-interoperability is one of the main features of emerging multimedia applications, the need is to be able to associate two operational functions: a fully automatic and a semi-automatic one. The fully automatic modality is necessary to process
large-scale information very quickly. It should be based on low-level features and be of
generic applicability, which can be very useful for post-production, archiving and index­ing. The semi-automatic one should be more user-friendly and oriented to creativity.
It is needed for non real-time editing of smaller scale material.
Chapter 3

Methodology

3.1 Multiple-feature Combination

Segmentation of still images is a low-level processing task that has been long investigated [25, 24, 26]. However an acceptable general purpose solution has not been achieved yet [29].

Moreover, the straightforward extension of still image segmentation techniques to video sequences of arbitrary content has proven to be unsatisfactory, due to the necessity of incorporating low-level features, or cues, of different nature, such as position, colour, texture, shape and motion [93].

There is evidence that simultaneous consideration of a number of features (so-called multiple-feature or multiple-cue segmentation), such as intensity, motion, texture, colour, depth, and so on, is useful towards reinforcing the spatio-temporal coherence of objects. Most recent attempts to perform multiple-feature segmentation resort to clustering, neural networks or supervised user intervention [88, 77, 90]. These approaches result in a hierarchical use of features and present the same difficulties envisaged in the hierarchical spatio-temporal techniques, such as over-segmentation, difficulty in the definition of weighting strategies and computational complexity.

A parallel use of features as presented in parallel spatio-temporal segmentation techniques has, on the other hand, shown to simplify these issues. On the other hand,
strictly speaking, parallel paradigms have not been used for multiple-feature segmentation, but only for spatio-temporal segmentation, i.e. for the integration of only two features or cues [2].

In the quest for developing an automatic multiple cue scene segmentation algorithm, two kinds of issues have to be addressed [1].

The first issue is represented by the choice of the segmentation algorithm. This choice in turn determines the kind of representation of the data and influences the design of a model to describe the mutual interrelationship between cues and the relative importance attached to each of them.

The second challenge is represented by the establishment of a hierarchy in one of two possible ways. It is possible to think of a hierarchy of features, or cues, where different relative strength is attributed to each feature to highlight the different perceptual importance. This is the case of hierarchical segmentation techniques. Another way of thinking of hierarchies is in functional terms, that is to say to take into consideration the order in which low-level tasks have to be performed in order to simulate a higher-level process carried out by the human visual system [6].

Proximity to perceptual tasks performed by the human visual system is an important perspective, that is not often taken into consideration in the design of paradigms for object-based segmentation. However, object-based segmentation has as the ultimate goal the extraction of meaningful objects as perceived by the end-users.

In this chapter, the baseline segmentation technique of choice for this work is presented. The extension of this technique to the realm of multiple feature segmentation is proposed. The closeness of the chosen segmentation algorithm to human visual perception is also investigated, by presenting a simplified model of the working of the visual cortex as far as feature analysis and functional processing is concerned.

3.2 Baseline Segmentation Technique

From the literature review presented in Chapter 2, it is possible to notice that region-based segmentation techniques are recognised to be the most useful techniques to per-
form object-based segmentation [1].

The most widely used region-based segmentation techniques for this purpose are two: region growing and watershed segmentation techniques [1, 2]. Watershed segmentation requires the definition of markers, which is problematic in a multi-dimensional feature space. Such techniques are also prone to noise, that is the reason why they are implemented in a multi-resolution fashion, making the process more complicated. Often watershed segmentation is used as means to simplify the image and initialise a set of homogeneous regions that is then incorporated into a hierarchical description of the frame [61, 94].

The representation of the information is also of primary importance and a lot of work has been dedicated to it. The most suitable representation of visual information with regards to object-based segmentation techniques appears to be the one of a hierarchical structure like a tree [1, 95]. In fact this is the terminology used also in the MPEG-4 and MPEG-7 description of composite objects [20, 21].

In this work, a graph-based segmentation technique called the Recursive Shortest Spanning Tree (RSST) has been chosen as the baseline segmentation technique. It is a region-growing approach first proposed by [34] that does not need initial seeds to perform the segmentation. It works on the basis of specified homogeneity criteria, in a step-wise optimal fashion which allows only the two most similar regions to be merged at each iteration of the process. During the segmentation process a data structure called a Shortest Spanning Tree (SST) is also built: this is a big advantage of this technique. Not only is the segmentation in an arbitrary number of regions performed, but also a hierarchical description of the information at an arbitrary granularity is also provided [68].

These reasons render the RSST technique a very promising one to be investigated in the field of object-based segmentation.

3.2.1 Graph-based Segmentation Techniques

Graph theory has been used for cluster analysis [96], in order to produce connected groups of vertices and the concept of Shortest Spanning Tree (SST) introduced to min-
imise the distance between vertices. There are boundary-following techniques [97] for joining edge points based on graph theory. Graph theory is also used in segmentation in the form of quad-trees [98]. Graph-based image analysis techniques represent hierarchically the image at different degrees of resolution and granularity. This provides the basis for the so-called pyramidal representations [99]. The pyramids can be regular, and in this case the hierarchy structure is bounded in advance, or irregular, for a more flexible decomposition of the information. Irregular pyramid representations are achieved by bottom-up clustering that at each stage merges two neighbouring regions according to a given similarity criterion [33].

Graph-based representations of the information are also becoming important in the content-based representation of visual information, where it is necessary to hierarchically describe a structure of the information at various levels of semantic interpretation. The spatio-temporal consistent information has been represented by binary trees in [95] and recursive shortest spanning trees have been used for colour segmentation of videos in [68], in conjunction with binary trees and for motion segmentation in [62] and in the European framework COST 211 in [13].

This method relies on the baseline segmentation technique provided by the Recursive Shortest Spanning Tree (RSST) algorithm [34].

The RSST algorithm presents several advantageous features that make it preferable to many segmentation techniques available in the literature [28]. As a graph-theoretical tool, it incorporates local relational information between elements. Furthermore it exploits global image information and it allows a hierarchical description of the scene. It has a higher degree of autonomy compared with other methods as it does not require initialisation steps such as markers, as in the case of watershed, or seeds, as in the case of region-growing techniques. Additionally, it does not impose external constraints on the structure of the region boundaries, like in the quad-tree methods.

3.2.2 Recursive Shortest Spanning Tree

This segmentation algorithm is based on the graph-theoretic concept of shortest spanning tree (SST) [34].
3.2. Baseline Segmentation Technique

Figure 3.1: Schematic representation of the elements of a graph $\mathcal{G} = (V, E)$.

Elements of Graph Theory

A graph $\mathcal{G} = (V, E)$, see Figure 3.1 is composed of a set of vertices $V_i$ connected to each other by links $E_{i,j}$, where $V_i$ and $V_j$ are the terminal vertices that the links connect. In a weighted graph the vertices and links have weights associated with them, say $V_i$, $V_j$ and $E_{i,j}$ respectively. If each vertex is linked to every other, the graph is said to be complete. A partial graph has the same number of vertices but only a subset of links of the original graph.

A chain is a list of successive vertices in which each vertex is connected to the next by a link in the graph. A cycle is a chain whose end links meet the same vertex. A tree is a connected set of chains such that there are no cycles.

A spanning tree is a tree which is also a partial graph. A shortest spanning tree of a weighted graph is a spanning tree such that the sum of its link weights, or some other monotonic function of its link weights, is a minimum for any other possible spanning tree. A forest is a set of trees and a spanning forest is a forest which is also a partial graph. A graph is planar if it can be drawn on a plane without any links crossing.

3.2.3 Application of Shortest Spanning Tree to Image Analysis

In order to use a graph model to perform the image analysis, the first step to take is to map the image into a planar graph. This mapping is shown in Figure 3.2. Each pixel belonging to the original image is represented by a vertex, which is linked to its
Figure 3.2: Example of one-to-one mapping of an intensity image onto a planar weighted graph. Each pixel in an image is mapped onto a vertex in the graph and links are established between 4-neighbour pixels. The vertex weight is the intensity of each pixel. The link weight is a similarity measure between linked vertices (in this example the absolute difference of the intensity of each pixel/vertex).

4-connected neighbours. If the only cue to be taken into consideration is intensity, then the weight of the each vertex is $V_i = I_{x,y}$, where $I_{x,y}$ is the intensity of the pixel of coordinates $(x,y)$. Assuming the image has $N_x$ rows and $N_y$ columns, each pixel is mapped on $i$, for $i = 1, \ldots, N_x \times N_y$, on a one-to-one mapping. If $V_i$ and $V_j$ are two neighbours, then the link between them can be described as the absolute difference of their intensities: $E_{i,j} = |V_i - V_j|$. The second step, after having mapped the image onto a graph, is to obtain a shortest spanning tree for it. A popular algorithm is presented in [100]. According to this, the graph of the image is thought of as a forest of single vertices. Then, for any tree in the forest, if the link with the lowest weight which connects that tree to another is added and, if it does not form a cycle, then the link will be in the shortest spanning tree. Construction of a SST is shown in Figure 3.3.

3.2.4 Shortest Spanning Tree Segmentation

Finally, the information contained in the shortest spanning tree of the graph is used to form partitions of the graph, which are labelled as regions of the segmented image. A spanning tree can produce these partitions by cutting its links. The forest produced
3.3 Characteristics of SST

Figure 3.3: Building a SST from a graph. In an image of \( N \) pixels, its SST can be built in \((N-1)\) iterations. At each iteration, the two most similar vertices (that do not form a circle) are joined. The links belonging to the SST are shown as full lines. The links that do not qualify for the SST are shown as dotted lines.

The operation is a set of disjoint trees. The set of vertices in the forest still span all the image so that no vertex is left out. If \( T \) is a tree in a forest, a partition \( P_i(T) \) is:

\[
P_i(T) = \begin{cases} 
1 & \text{if } V_i \in T \\
0 & \text{otherwise}
\end{cases}
\]  

(3.1)

Once a partition of a graph has been obtained, the set of partitions has to be mapped back onto the image, in order to form the regions. One way to do this is to generate the segmentation image such that each pixel in one region is assigned a constant intensity value. This value is the mean of all the pixels in the region. Therefore, in order to form \( R \) regions, the SST algorithm can be described as follows:

- Map the image onto a weighted graph.
- Find a SST of the graph.
- Cut the SST at the \( R - 1 \) most costly links.
- Map the forest onto a segmentation image.

3.3 Characteristics of SST

The SST algorithm has some interesting features that are worth highlighting. Unlike some other global methods, e.g. thresholding, the spatial information about the neighbourhood of each pixel or each region is used, therefore all the regions produced are
made of connected components. The edges of the regions produced are well defined and are not jagged like the ones produced by split-and-merge procedures, for example. The regions produced are closed so there is no need for edge linking. Additionally, the edges are one-pixel-wide so there is no need for thinning. The segmentation tree contains all the information needed to segment the image in a hierarchical order. This allows the user to choose the level of detail needed for the particular application. When a new region is added, only that specific region which is split in two is modified. This means that established boundaries do not change as finer details are included. It allows both region-based and edge-based segmentation to be performed.

The most evident drawback of the method is also common to other single linkage methods. If regions are very different, but they are connected by a set of pixels that are slowly degrading from one grade of intensity to another, they will be merged, even though they have a large overall difference. Moreover, no global information is exploited. This is particularly inefficient in the presence of noise, where there are spurious spikes of intensities that are identified as single regions because of their difference with their neighbours. These two characteristics together cause the algorithm to produce a large number of small, insignificant regions while merging the fundamental objects together.

A number of improvements to the basic formulation of the algorithm have been proposed in [34], especially in order to overcome the shortcomings of not using any global information. The most interesting variation is called Recursive Shortest Spanning Tree (RSST) algorithm.

The flowchart of the algorithm is given in Figure 3.4.

3.4 General Formulation of RSST and Application to Multiple-feature Segmentation

The RSST method allows both the weights of the vertices and the weights of the links between vertices to contain any kind of information that the user might see fit to exploit [101]. It is possible to shape the weight of each vertex in such a way that it contains
3.4. General Formulation of RSST and Application to Multiple-feature Segmentation

Figure 3.4: Flowchart of RSST algorithm.
a combination of features related to the visual information of interest. The weighting function between links can be designed as a cost function which represents similarities between regions, or equivalently as a distance measure between feature vectors. In the development of this project, the way to define both weights in order to reflect the dynamic interrelationship between cues from a perceptual point of view will be investigated. This will constitute a new way to express similarities between regions, which is to date an open issue.

A spatially digitised image $I$ can be specified as a two dimensional field of $M \times N$ elements defined on an $m$-dimensional representation space. Any particular ranking in the element space, i.e. raster scan, allows a single-valued indexing of image elements, which can be expressed as:

$$I = [I(0)I(1)\ldots I(MN - 1)]$$ (3.2)

In general we define an element $I(p)$ as a $m$-dimensional vector:

$$I(p) = [I_0(p)I_1(p)\ldots I_{m-1}(p)]$$ (3.3)

where $I_i(p)$ is the amplitude of the $i^{th}$ cue of the $p^{th}$ image element. Examples of cues are grey-level intensities, colour triplets, motion vectors and so on. Cues such as colour are readily available from the image acquisition process while cues like motion vectors require additional computation; typically the application of a motion estimation algorithm is needed.

Initially, an image $I$ is mapped onto a graph $G^I$:

$$f : I \rightarrow G^I$$ (3.4)

A graph $G^I = \{U, L\}$ is a collection of vertices $U = \{U_k\}_{k=0}^k$ and a collection of links $L = \{L_{k,t}\}_{k,t=0}^k$ which join all or some of these vertices. The graph is four-way connected and hence planar. Furthermore, it is weighted by assigning appropriate weights to each link. Finally, the graph evolves with time to allow for a description of the recursion which is central to the segmentation algorithm and this is denoted by $G^I(t)$. 

3.4. General Formulation of RSST and Application to Multiple-feature Segmentation

The algorithm establishes a correspondence between vertices and regions of the segmented image. Consequently, the complete specification of a vertex needs to contain both the description of cues of interest and the spatial description (i.e. the pixel coordinates) of a region. For this purpose two auxiliary quantities, $\bar{V}_k$ (i.e. the vector of the cues of interest) and $E_k$ (i.e. the set of pixel co-ordinates), corresponding to vertex $U_k$, are introduced. The graph can now be expressed in the following form which explicitly contains a time dependence:

\[
G^t = \{U(t), L(t)\} = \{U_k(t)_{k=0}^{MN-1-t}, L_k(t)_{k,l=0}^{MN-1-t}\} = \{\bar{V}_k(t)_{k=0}^{MN-1-t}, E_k(t)_{k=0}^{MN-1-t}, L_k(t)_{k,l=0}^{MN-1-t}\}
\]

The initial conditions are such that each vertex bears a one-to-one correspondence with a pixel in the original image:

\[
\bar{V}_k(0) = \bar{r}(k) = [I_0(k)I_1(k)...I_{m-1}(k)]^T
\]

\[
E_k(0) = k; k = 0, 1, ..., MN - 1
\]

To complete the set of initial conditions, link weights $L_{k,l}$ are assigned according to a specified cost function. The evolution of the graph in time depends primarily upon the identification of the link with the minimum weight:

\[
L_{\text{min}}(t) = \min_{k,l}\{L_{k,l}(t)\} = L_{k,\lambda}(t)
\]

The minimum-weight link connects the pair of vertices $(\bar{V}_k(t), \bar{V}_\lambda(t))$. The corresponding regions will consequently be merged:

\[
E_k(t + 1) = E_k(t) \cup E_\lambda(t)
\]

The new region will be reconfigured in space and assigned a new representative value for each of the cues considered. If the average is used, for instance, then:
58 Chapter 3. Methodology

\[ \bar{V}_k(t+1) = \frac{||E_\kappa(t)|| \bar{V}_k(t) + ||E_\lambda(t)|| \bar{V}_\lambda(t)}{||E_\kappa(t)|| + ||E_\lambda(t)||} \]  

(3.11)

where \( ||E|| \) is the cardinality of set \( E \).

Vertices not connected with the minimum weight link are not modified:

\[ \text{all } k \neq \kappa, \lambda : \bar{V}_k(t+1) = \bar{V}_k(t) \]  

(3.12)

\[ \text{all } k \neq \kappa, \lambda : E_k(t+1) = E_k(t) \]  

(3.13)

Once a new region is formed, link weights need to be recalculated where appropriate. Links not connected with the vertex, which corresponds to the newly formed region need not be updated:

\[ \text{all } (k, l) : \{ k \neq \kappa \} \land \{ l \neq \lambda \} : L_{k,l}(t+1) = L_{k,l}(t) \]  

(3.14)

The weights of the links connected to the newly formed region are updated according to the corresponding cost function. It should be noted that up to this point no choice of cost function has been specified. At the next iteration of the algorithm the weights of all the links that are connected to the new region are recomputed before the new minimum-weight link is selected. The links chosen in this way define a spanning tree on the original graph and the order in which links are chosen defines a hierarchy of image representations.

3.4.1 RSST Performance Issues

The RSST framework provides an automatic way of obtaining a hierarchical description of the picture for an arbitrary number of regions, exploiting both local and global information. It can be flexibly adapted to perform segmentation with the use of different features and the joint-use of features. An example of a hierarchical description obtained with the RSST is given in Figure 3.5. The results have been produced using only
3.4. General Formulation of RSST and Application to Multiple-feature Segmentation

Intensity $I_{x,y}$ as a cue for segmentation, as in the original method proposed by [34], the notation of which has been reported in Section 3.2.4. The vertex $V_i$ representing the region $\mathcal{V}(x,y)$ has weight:

$$V_i = \frac{\sum_{x,y} P_{x,y}(i)I_{x,y}}{\sum_{x,y} P_{x,y}(i)}$$  \hspace{1cm} (3.15)

At the beginning each pixel is considered to be a separate region and at each stage of the iteration the recursion merges two regions together. At each stage the most similar vertices, $V_i$ and $V_j$ are merged together and the new vertex $V_k$ representing the new region $\mathcal{V}(x,y)$ has weight:

$$V_k = \frac{\sum_{x,y} P_{x,y}(i)I_{x,y} + P_{x,y}(j)I_{x,y}}{\sum_{x,y} P_{x,y}(i) + P_{x,y}(j)}$$  \hspace{1cm} (3.16)

The new partition is $P_{x,y}(k) = P_{x,y}(i) + P_{x,y}(j)$. The link that joined vertices $V_i$ and $V_j$ is removed and saved.

The test image used here is a frame from test sequence Renata. In this work, three test sequences have been chosen to represent general results. Information about this material can be found in Appendix A.

As observed in [102], this method tends to be robust to noise, since noisy pixels are reduced to single leaves in the tree and they therefore do not influence the growth of main regions. This however produces a number of small unnecessary regions that have to be cancelled by means of post-processing. It can also cause over-segmentation in the presence of finer textures that tend to have the characteristics of noise.

The use of global information notwithstanding, this method is particularly sensitive to the presence of weak boundaries between regions. However, this is a global disadvantage of region growing methods.

In this section, the characterisation of the RSST is addressed with emphasis on the complexity, the performance under noisy conditions and the avoidance of over-segmentation without recourse to post-processing.
Figure 3.5: Hierarchical representation of the information. The RSST algorithm allows the user to choose the detail of the segmentation. In this case, the sequence is represented with the respective edge maps at 10000 regions in (a) and (b), 5000 regions in (c) and (d) and 1500 regions in (e) and (f).
3.4. General Formulation of RSST and Application to Multiple-feature Segmentation

Complexity Issues

The RSST algorithm employed for baseline segmentation is a stepwise optimal algorithm [103]. The link weight function reflecting similarity between regions is a measure of the error committed at each merging step as the region growing evolves. By minimising the error at each iteration of the algorithm, stepwise optimality is guaranteed.

The algorithm has a nominal computational complexity of approximately $o(n^2)$ for an image of $n$ pixels [104]. Optimising algorithm performance with respect to speed has not been a priority issue for this investigation. Nevertheless, fast versions have been proposed in the literature [105]. A non-optimised implementation of the baseline algorithm running on a 400 MHz Ultra Sparc 4 CPU server takes approximately 30 seconds for a ITU Rec.601 frame (i.e. $720 \times 576$ pixels). For a detailed analysis of the complexity of RSST, the reader may refer to [104, 102].

Performance in the Presence of Noise

As a test of efficiency of the algorithm, a testing procedure has been designed. The test image is a checkerboard (therefore the number of regions in known in advance and equal to the number of squares), corrupted by two sources that occur quite often in real pictures: noise (Gaussian additive noise) and blurring. The effect of the changing of the number of segmentation regions, on the segmentation results, has been tested, addressing the cases of over-segmentation and under-segmentation. The evaluation has been carried out by comparing the amount of noise which is given as an input to the segmentation tool and the segmentation error. The noise has been calculated as Peak-Signal-to-Noise-Ratio (PSNR), the definition of which is given in Appendix B. Figure 3.6 summarises the results of this experimentation. In the axis labelled $err.$ in the amount of noise given as input is shown as PSNR, measured in $dB$. The same applies for the segmentation error, $err.$ out. In the regions axis, the number 0 indicates that the segmentation provides the number of regions equals to the regions contained in the input image, the negative values indicate under-segmentation. The regions are one less and two less than the natural number of regions. The same applies to the positive
Figure 3.6: Assessment of algorithmic performance: response of the algorithm (noise produced as output, computed as PSNR in dB) when different levels of (a) Gaussian additive noise, (b) blurring or (c) the two sources combined are corrupting the image. It is possible to notice a critical level of noise beyond which the algorithm breaks down.
range in the case of over-segmentation. From the diagram, it is possible to notice that there is a critical level of noise in which the algorithms breaks down.

3.5 Relationship to Human Visual Perception

The principal problem affecting existing segmentation algorithms is that the segmented regions are usually not perceptually meaningful. This represents a particularly critical difficulty when the segmentation has to be used as a basis for object-oriented video coding.

In this section, a necessarily brief overview of the functional structure of the visual cortex is given. The purpose is to provide a simplified model of the Human Visual System (HVS) and to verify the correlation of the chosen mathematical model to the processing of a biological viewer. The mathematical details of the visual signal processing are omitted since, except for the first region of the visual cortex, the mathematical model is not available yet [106]. However, some of the task division attained in the visual cortex and the functionalities of some areas are well-known in the psycho-visual community and yet not taken into account in the elaboration of new strategies for image processing or computer vision.

In this work, it is important to take into consideration what is to date known of the HVS at least as a useful reference, since the main goal of object-based segmentation is to extract meaningful objects for the end-user.

3.5.1 Psycho-visual Perspective

Much of the primate cortex is devoted to visual processing.

The visual stimulus is received by photoreceptors in the retina of the eye and transmitted through the optic nerves to one of the lateral geniculate nucleus (LGN), see Figure 3.7. LGN functions as a connection between the biological imaging device (the eye) and the biological visual signal centre, the visual cortex, depicted in Figure 3.8. All the information collected by the retina enters the visual cortex from an area called V1 or the striate cortex, so called because of its striped appearance.
Chapter 3. Methodology

Figure 3.7: LGN acts as a bridge between the eyes and the visual cortex. Simplified diagram from [3].

Figure 3.8: Location of functional areas of the visual cortex. Adapted from http://www.biols.susx.ac.uk/home/GeorgeMather/.
3.5. Relationship to Human Visual Perception

Hubel and Wiesel [107] were the first to study this area and to publish a detailed description in 1962. The striate cortex acts as a hierarchical array of feature detector cells. These cells have a structure varying from simple to complex and hyper-complex. They are receptive to edges and bars. The structure of V1 is a layered one, with layers responding selectively to particular orientations and a hierarchy of feature detectors that progressively respond to more and more complex stimuli.

Hubel and Wiesel's theory was that this hierarchy of feature detectors progressively provides a description that spans from the simple to the complicated object, from edges, to corners, to surfaces, to more symbolic and abstract objects, such as a Volkswagen or a grandmother. This theory contains some flaws that have been exposed. In particular, the hierarchy of the processing does not appear to be a simple one, but there is evidence of feedback from higher centres of elaboration (feature detectors) to lower ones, i.e. the outputs of lower feature detectors can be progressively refined by some higher level of processing (Ferster and Lindstrom, 1983 [107]). So there is evidence of the presence of feature detectors that collect information from different visual features or cues. The
question now is in which kind of pathways the information is actually processed and is this processing really a hierarchical one? To find the answers to these questions psycho-visual scientists turned their interest beyond the striate cortex towards the extrastriate cortex, so called because it does not contain striped cells, like the striate cortex. The extrastriate cortex contains a set of visual areas, which contain retinotopic maps of particular visual features or cues as they manifest themselves in the pattern of neuronal activity initiated by light response. Some visual areas are straightforward to locate and identify, some others are more problematic, therefore the actual number, location and function of all visual areas is not yet known. However, a study of the extrastriate cortex of the macaque monkey by Maunsell and Newsome, 1987 [107], revealed 19 visual areas. The map of the visual areas and connection are shown in Figure 3.9. The cascade of connections is not one of a simple hierarchical chain of processing, each area sends outputs to several others and all connections from visual areas are matched by reciprocal ones running in the opposite direction. Van Essen (1985), [107] introducing a very detailed description of visual pathways, counted at least 92 regions. Each pathway can be classified according to the cortical layer where they originate or terminate as ascending (from V1) or descending (towards V1). The processing made by this connection presents a clear hierarchical organisation through different layers. Therefore the extrastriate cortex seems to have a hierarchical structure, but an unusual one, with different centres of elaboration at each layer (or level) and extensive feedback for refinement of outputs. What kind of functions and structure of processing can be envisaged in the layered structure of the hierarchical pathways? Maunsell and Newsome 1987 and Van Essen 1985 [107] have carried out extensive research.

It leads to functional distinction between visual areas. V1 and V2 are areas of simple representation. Extrastriate pathways are divided into two main directions that operate in parallel and are related to motion analysis on one side and spatial layout analysis on the other side with colour, shape and object recognition, see Figure 3.10. The motion pathway, also called dorsal pathway because it is located in the upper part of the visual cortex, runs from V1 to medial temporal (MT) and finally to and medial superior temporal area (MST) and 7a in the parietal lobe. The pathway concerned
3.5. Relationship to Human Visual Perception

Figure 3.10: Dorsal and ventral pathways stem from areas V1 and V2. They have complementary functionalities. Simplified diagram from [4].

with colour and form, also called ventral pathway because it is located in the lower part of the visual cortex, goes from V1 to V4 and then to posterior inferotemporal (PIT) and finally anterior inferotemporal (AIT). The definition of these areas depends on their location: the temporal area of the brain is at the side of the forehead, while the parietal area is at each side of the head. The temporal and parietal areas are connected to the occipital area in the back of the head, where the primary visual cortex resides. Anterior indicates the frontal part of an area, posterior indicates the back of an area, superior indicates the upper part of an area, medial indicates the middle part of an area and inferior indicates the lower part of an area. The segregation between these layers is not clear cut and there are communications and overlapping, see Figure 3.11.

To summarise, the signal travels from the eye, through the optic nerve, to a LGN. The V1 area (striate cortex) receives the information from the LGN. It roughly analyses region by region the retinal contours and is connected to V2, where the extrastriate cortex begins. V1 acts as a bank of edge detector filters and it is organised in a hierarchical way. Some of the cells in V1 show a very high selectivity to orientation, others have a rather more concentric receptive fields, which appear like blobs and are more sensitive to opponent-colour [108]. These cells provide a rather patchy input to the following area in the extrastriate cortex indicated as V2 [109]. V2 contains an
Chapter 3. Methodology

Figure 3.11: Temporal (T), parietal (P) and occipital (O) lobes locations. The lobes are interconnected. Simplified diagram, originally found in [5].

approximate retinotopic map of the features presented in V1. From V2 two pathways arise: the ventral pathway and the dorsal pathway [110]. The ventral pathway consists of V3, V4 and the infero-temporal cortex. V3 is concerned with the perception and localisation of textures [111]. V4 performs colour processing, enforcing colour constancy in the opponent-colour domain [112]. The infero-temporal cortex is involved in object recognition and visual memory. The dorsal pathway is constituted by V5 (also called MT), MST and 7a [112]. V5 and MST are instrumental to motion detection, the MST being more sensitive to angular motion. The ventral intraparietal (VIP) and 7a regions are involved in visual attention and 7a also provides an attention mechanism to motion detection.

Is the elaboration carried out by each of the parallel pathways hierarchical? According to Zeki (1980) the pathway concerned with form and colour analysis ranges from very selective towards small differences in wavelength to less selective and a degree of colour constancy is shown. So there is a separation between luminance and chrominance components. The theory which has more respect to date is that the processing of information is carried out in a parallel way in separated pathways (primal visual
3.5. Relationship to Human Visual Perception

pathway theory). In each area of layer of processing the information is processed in a hierarchical way. Additionally, the parallel pathways are not to be considered separately because there are extensive connections between areas, with feedback to lower areas of processing, that presumably help in refining the elaboration [107].

All the elaboration carried out by these visual areas is also influenced by memory attention and organisation of the behaviour. So that psychologists suggest that there are actually two kinds of segmentations, one is an attentive and the other is non-attentive [113]. However, we are interested in non-attentive processing, as also among psychologists there is agreement that early vision is more concerned with non-attentive segmentation [114, 107]. This should also be the realm of a general-purpose object segmentation algorithm.

The problem of the combination of the outputs of different visual areas is still an open one in the psycho-visual community. It is often referred to as the binding problem. In [115] an attempt of an answer is provided: the way of combining outputs from different visual areas would be the result of a set of rules learned through experience and evolution and wired in the folds of connections between different visual areas.

All these concepts are represented in Figure 3.9, where the structure of the visual cortex has been simplified to include only the areas that have been more widely studied and the path of connection has been pruned to include only the connections between these latter areas. As far as a non-attentive segmentation mechanism is concerned, the areas of interest end with V4 and V5.

From this description, it is perhaps possible to appreciate the complexity of the structure of the visual cortex. Modelling of the HVS for reverse engineering purposes is attracting increasing interest from the image processing community. However, much work needs to be devoted to:

- The identification of the functions of each of approximately 92 visual areas.
- The identification of the network of connections between visual areas and their mutual influence.
- The mathematical modelling needed for description, engineering and simulation.
In this work, a limited number of visual areas have been taken into consideration for analysis, since their function seems understood. A limited number of connections is taken into account. No mathematical modelling is considered, since no mathematical framework is available after the area V1. Only areas concerned with pre-attentive segmentation are of interest for this work, since no a priori information should be incorporated in a generic model. The work will be therefore based on concepts gathered from psycho-visual perception.

These concepts can be summarised as follows, for the structure of the early human visual system:

- Spatial and temporal processing are carried out in a separate and parallel fashion.

- There are different visual areas, each of them concerned with the elaboration of different features. In particular, motion information is processed in parallel to and texture information.

- There is a layered structure with hierarchical processing of the features inside visual areas and visual pathways.

- There is feedback between different layers and different visual areas.

Parallel methods for object based segmentation of the information seem more motivated from the psycho-visual perception point of view. Hierarchical methods of segmentation can be seen as perceptually motivated but only in the case of single-feature segmentation: in fact single visual areas and pathways have a hierarchical structure. However, as far as the processing of multiple-features is concerned, the prevailing psycho-visual theory is the one of parallel independent processing of spatial information, i.e. texture and colour, and temporal information, i.e. motion, although there is extensive interaction between the two pathways. It is also useful to remember that the two pathways act on common information which is supplied by V1 in terms of edges and V2 in terms of colour patches.
3.6 Conclusions

In this section, the RSST framework has been reviewed as an effective method to obtain segmentation and representation of video information.

The RSST framework has been presented in [34] and then further adapted as a tool to perform region-based coding in [116], colour segmentation in [117], motion information in [62] and texture segmentation in [118].

In this section, an extension of the RSST framework has been presented in order to achieve multiple-feature segmentation by means of the association of a feature vector to each vertex of the graph and the use of a multi-dimensional cost function. The novelty of this approach is related to the extension of the RSST framework to the multiple-feature segmentation application [93, 101].

The correlation of the RSST framework to the HVS has been investigated by presenting a simplified model of the HVS functionalities, based on well-accepted evidence of psycho-visual studies into mammalian visual perception. From the comparison of these two models, it looks evident that, while a graph-based segmentation method is well-tuned with current state-of-the-art endeavours to develop a hierarchical object-based representation, this method does not share any similarity with biological processes of obtaining meaningful objects.
Chapter 4

Feature Extraction and Segmentation

4.1 Introduction

Object-based segmentation of video objects is effectively segmentation of moving objects for this application, since it is difficult to identify still meaningful objects without the use of higher level semantics.

While motion is a fundamental source of information, in multiple feature segmentation, motion, which prescribes a global property of the object, has to be fused with local properties like colour or texture.

This chapter discusses how to extract features of interest and how to obtain meaningful segmentations of video sequences using single features. The requirements needed for feature extraction and segmentation are:

- generality
- simplicity
- robustness

73
4.2 Motion Segmentation

Motion is a cue of fundamental importance both for the definition and the localisation of objects. It has been pointed out that, for video content extraction, it is very difficult to define a meaningful object without a priori knowledge of the application and the scene itself, unless motion is present. Moving objects are meaningful entities, articulated in different parts. Motion information gives also a clear indication of the position of such objects in the scene, restricting the area of search for entities that share common meaning and properties.

In this section, the problems of motion estimation and motion segmentation are addressed. Particular emphasis is directed at labelling under very broad assumptions, therefore enforcing the segmentation with some information about the meaning of the objects themselves. The verification of the consistence of such labelling and a novel use of the information provided by model outliers have also been addressed.

4.2.1 Role of Motion

Motion is a powerful cue used by viewers to extract objects of interest from irrelevant details. In this sense, a sequence of images is more meaningful than a still image because motion strongly indicates the presence of a meaningful object. An indication of the importance of motion, as a source of information about the objects present in a sequence, is the fact that the first object-based segmentation techniques rely on the motion information only.

In imaging, motion arises from relative displacement between the sensing system and the view that is being observed.

A classification of methods used for motion estimation can be achieved on the basis of motion models, estimation criteria, and search strategies [106, 119].

Some preliminary notation is needed. It is possible to refer to both continuous and discrete representations of motion and images, with bold characters denoting vectors. Let \( \mathbf{x} = (x, y)^T \) be a spatial position of a pixel in continuous coordinates, i.e. \( x \in \mathbb{R}^2 \) within image limits. Let \( I_t \) denote the image intensity at time \( t \). Before the images can
be digitally manipulated they have to be sampled and quantised. Let \( \mathbf{n} = (n_1, n_2)^T \) be the discretised spatial position of the image corresponding to \( \mathbf{x} \). Similarly let \( k \) be a discretised position in time, denoted \( t_k \). The triplet \( (n_1, n_2, k)^T \) belongs to a 3-D sampling grid, for example a 3-D lattice. Motion in continuous images can be described by a velocity vector \( \mathbf{v} = (v_1, v_2)^T \), where \( \mathbf{v}(\mathbf{x}) \) denotes the velocity field or motion field. The computation of this dense field is often substituted by the computation of a small number of motion parameters \( \mathbf{b}_t \), by means of a known transformation. For discrete images, the notion of velocity can be substituted by displacement \( \mathbf{d} \), given that the sampling period is constant and known a priori.

### Motion Models

The goal of motion estimation is to estimate the motion of image points, that is to say 2-D motion or apparent motion. Such a motion is due to the 3-D motion of a camera and/or the motion of 3-D objects in a scene.

Two simple models are successfully used.

- The velocity can be expressed by two parameters as:

  \[
  \mathbf{v}(\mathbf{x}) = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}
  \]

  This is a 2-D translational model where parameters \( \mathbf{b} = (b_1, b_2)^T = (v_1, v_2)^T \). This model can approximate very closely the motion field in most natural images and it is widely used for compression.

- The velocity can be expressed by six parameters as:

  \[
  \mathbf{v}(\mathbf{x}) = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} + \begin{pmatrix} b_3 & b_4 \\ b_5 & b_6 \end{pmatrix} \mathbf{x}
  \]

  This is known as affine motion model.

The translational model in Equation 4.1 is a particular case of the Equation 4.2.
Temporal Motion Models

Temporal motion models aim to model the trajectory followed by a point $x = (x,y)$ through time. The simplest model is to assume that the trajectory is linear and the point $x = (x,y)$ is moving with constant velocity between the times $t$ and $\tau (\tau > t)$. Therefore, the position of a point at time $\tau$ can be expressed according to Equation 4.3:

$$x(\tau) = x(t) + v(t)(\tau - t) = x(t) + d_{t,\tau}(x)$$  \hspace{1cm} (4.3)

Where $d_{t,\tau}(x)$ is a displacement vector measured in the positive direction of time. The use of a linear trajectory, which embeds the 2-D translational model presented in Equation 4.1, is used very effectively in practice for motion estimation. In this case, the goal is achieved estimating the two parameters of the velocity $\mathbf{b} = (b_1, b_2)^T = (v_1, v_2)^T$.

Region of Support

The set of points $x$ to which a spatial or temporal model is applied is called the region of support, $\mathcal{R}$. Typically, four kinds of region of support are considered:

1. $\mathcal{R} = \text{whole image}$. This is suitable for the camera motion analysis.

2. $\mathcal{R} = \text{one pixel}$. Typically, for this region of support, the 2-D translational model represented by Equation 4.1 is used together with the temporal model expressed by Equation 4.3.

3. $\mathcal{R} = \text{rectangular block of pixels}$. Typically, for this region of support, the 2-D translational model represented by Equation 4.1 is used together with the temporal model expressed by Equation 4.3. This has been successfully applied for all digital compression standards.

4. $\mathcal{R} = \text{irregularly shaped region}$. Regions $\mathcal{R}$ are expected to be object projections. Typically, for this region of support, the 2-D translational model represented by Equation 4.2 is used together with the temporal model expressed by Equation 4.3.
Observation Models

Motion arising in a sequence of images is perceived by the human viewer as a variation in intensity or colour. Therefore, the relationship between the variation in intensity and trajectory of a point in an image carries information for the estimation of the motion. The general assumption is that the intensity of a point remains constant along the motion trajectory. For temporally and spatially sampled images, this means:

\[ I_k(n) = I_{(k-1)}(n - d) \]  \hspace{1cm} (4.4)

where \( I_k(x(t_k)) = I_k(n) \), \( t = t_{k-1} \) and \( \tau = t_k \) according to the notation used for Equation 4.3.

In Section 4.2.2, the intensity constancy constraint will be expanded and analysed in more detail. It is however interesting to notice that the intensity constancy constraint does not hold in the case of variation of illumination of the scene. In this case, constraints based on the constancy of the spatial gradient in the direction of the motion can be formulated.

Estimation Criteria

The goal of motion estimation is to fit the observed data to a model that approximates the real motion field as accurately as possible, according to the specifications of a particular application. The models presented have to be described through an objective function, representing the error made in the approximation or the closeness of the approximation to the real solution. The estimation criteria have then to be optimised.

Since a dense motion field has been used in this work, let us review the pixel-domain estimation criteria. They are usually produced by the intensity constancy criteria seen in Equation 4.4. These criteria aim to minimise an error function, for example the one expressed by Equation 4.5:

\[ \epsilon_k(n) = I_k(n) - \hat{I}_k(n) \]  \hspace{1cm} (4.5)
Where \( e_k(n) \) is the error made approximating the observed data \( I_k(n) \) with the model \( \tilde{I}_k(n) \). The error made in the approximation is derived from the intensity constancy and the smoothness constraints. This will be discussed in the Section 4.2.2, together with the choice of the estimation criterion, which can be made robust to errors arising from the choice of the model.

Search Strategies

Once the models have been fitted to the estimation criterion, it is necessary to compute the estimates of the motion parameters by means of a search strategy. Instead of giving a list of search strategies, which would be long and incomplete, two of the most widely used strategies, which are of interest for this work, are mentioned here:

- **Matching** is a way of minimising the approximation error where motion candidates are compared with the original observation within a certain region of support. The best match becomes the optimal approximation.

- **Gradient-based techniques** are developed into Taylor expansions of the image functions \( I_{k-1}(n - d(n)) \). It is necessary to take into account that the Taylor series expansion is valid only in the immediate vicinity of the initial \( d \).

4.2.2 Robust Optic Flow Estimation

In [120], computation of correspondences between points of interest and computation of optic flow are presented as complementary approaches to dynamic image analysis.

Optic flow computation requires a small time difference and no significant changes between consecutive frames. It produces a vector indicating motion direction and velocity for possibly all image points (dense optic flow) using the minimum amount of information regarding the objects of interest (it is suitable for object model-independent analysis). It is therefore a technique well suited to produce the motion information required for the specifications of this project.
Optic flow reflects the changes occurring in an image due to the motion in a interval of time \( dt \). It produces a motion field as a two dimensional projection across the image of a three dimensional motion [6].

Optic flow computation is based on the following two assumptions:

- The observed intensity of any object point is constant in time (intensity constancy constraint).
- Nearby points in the image plane move in a similar manner (velocity smoothness assumption).

The intensity constancy constraint is represented by the Equation 4.4. Let \( s \) be a variable along the motion trajectory. The intensity constancy constraint can be rewritten in the continuous case as expressed by Equation 4.6:

\[
\frac{dI}{ds} = 0
\]  

(4.6)

Therefore, the intensity constancy constraint requires the directional derivative in the direction of the motion to be zero. Since \( I = I(x, y, t) \) this equation can be expanded, using Taylor expansion truncated at terms higher than the first, as in Equation 4.7:

\[
\frac{\partial I}{\partial x}v_x + \frac{\partial I}{\partial y}v_y + \frac{\partial I}{\partial t} = (\nabla I)^T v + \frac{\partial I}{\partial t} = 0
\]

(4.7)

Where \( \nabla = (\partial/\partial x, \partial/\partial y)^T \) is the spatial gradient and \( v = (v_x, v_y)^T \). Equation 4.7 can be used to compute the velocity \( v \). One wants to find the parameters of the model of \( v \), in such a way that fitting the model of the velocity to the data available (in this case the intensity gradients) approximates Equation 4.7 best. This means that the error \( \varepsilon_D(v) \) caused by the approximation of the optic flow with the use of the intensity constancy constraint should be minimised. Therefore, Equation 4.7, in the continuous and discrete form, is called the data conservation constraint:

\[
\varepsilon_D(v) = \left( \nabla^T I(x)v(x) + \frac{\partial I((x))}{\partial t} \right)
\]

(4.8)
Equation 4.8 does not specify the optic flow completely, so the *smoothness constraint* is introduced. Full details can be found in [42]. The idea of the smoothness constraint is that the velocity should change slowly in a given neighbourhood, that is to say, neighbour pixels should share the same or very similar velocity. The variation of the velocity, $\varepsilon_S(v)$ can be expressed by the squared gradients in the $x$ and $y$ of the horizontal and vertical components of the velocity, here indicated as $v_1$ and $v_2$:

$$
\varepsilon_S(v) = \left( \| \nabla(v_1(x)) \|^2 + \| \nabla(v_2(x)) \|^2 \right) \quad (4.9)
$$

The variation of the velocity $\varepsilon_S(v)$ should be as small as possible. This consideration gives the idea of minimising an objective function representing the error made by the approximations introduced by the intensity and smoothness constraints. The objective function to be minimised can be written as Equation 4.10:

$$
\varepsilon(v) = \varepsilon_D(v) + \lambda \varepsilon_S(v) \quad (4.10)
$$

This is a formulation which seeks to minimise the weighted error of the optic flow equation and the pixel-to-pixel variation of the velocity field. The brightness constraint can be violated by motion boundaries, shadows, reflections or changes in illumination. The smoothness constraint forces the local flow to be close to that of its neighbour. When a motion discontinuity occurs, a smoothing along the boundary is present. This reduces the accuracy of the information obtained by the motion estimator at the boundary. The violation of the constraints results in gross measurement errors that are referred as outliers. There is a need for techniques that are robust to outliers and can therefore cope with violation of the model chosen to represent the problem. Robust estimation is designed to decrease the sensitivity of the measurement to violation of the model [121]. This is achieved by means of robust statistics, whose goals are [12]:

- To describe a model that fits the data better.
- To select data points that violate such a model for further processing, if needed.
4.2. Motion Segmentation

Figure 4.1: Differences in the influence of non-robust (first row) and robust (second row) estimators.
Chapter 4. Feature Extraction

The problem of fitting a certain model to the data is approached by minimising a function of the error made when approximating the observed data with a set of given assumptions. If $e$ is the error function presented in Section 4.2.1, the minimisation criterion can be summarised as in Expression 4.11:

$$\min \sum_{n \in \mathcal{R}} \Phi(e(n))$$

(4.11)

Where $\Phi$ is the estimator. When the errors are normally distributed, the optimal estimator is a quadratic one, which corresponds to the maximum likelihood estimation criterion. This formulation gives rise to the standard least-squares estimation problem. Defining as outliers the points that do not fit to the actual model and therefore generate a large error in the approximation, the quadratic estimator is not robust to the presence of outliers. This can be shown by an influence function $\Psi$, which is the derivative of $\Phi$, as shown in Figure 4.1. A robust estimator is an estimator which is insensitive to small deviations from the assumptions on which the minimisation is based [122] (what is usually indicated by small deviation is that a small number of data exhibit a large deviation) [122]. Intuitively then, the function $\Psi$, as a derivative of the estimator $\Phi$ according to the estimation error $e$, shows the sensitivity of the estimator to increasing values of error, hence the influence of the outliers [123]. Mathematically speaking (the interested reader is referred to [124] among other texts for the complete mathematical treatment), the function $\Phi$ is a weight factor of the function, dependent on the estimation error, which needs to be minimised. If $\Psi$ increases with an increase in the error, then the influence of outliers becomes notable. For a robust estimation, the influence function $\Psi$ has to remain constant or decrease for large values of the error.

This behaviour is illustrated in Figure 4.1. In the case of the quadratic (least-squares) estimator, the influence of the outliers increases linearly without bound. In contrast to this, in the case of the Lorentzian estimator, the influence of outliers increases only up to a fixed threshold and then the influence goes to zero. Such estimators are presented as robust estimators.
4.2. Motion Segmentation

<table>
<thead>
<tr>
<th>Motion Model</th>
<th>Affine (Equation 4.2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temporal model</td>
<td>Linear (Equation 4.3)</td>
</tr>
<tr>
<td>Estimator</td>
<td>Quadratic</td>
</tr>
<tr>
<td>( N )</td>
<td>0</td>
</tr>
<tr>
<td>( \sigma )</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 4.1: Parameters for the LS experiment.

Experiment: Least Squares versus Robust Estimation

In this project, the dense optic flow field has been estimated using a least-squares (LS) estimation formulation as in [125] and a robust estimation (RE) formulation with a Lorentzian kernel as in [12].

In [125], a gradient-based technique is applied. The data constraint of Equation 4.8 is applied to a patch of an image of dimension \( N \times N \). An affine model is fitted to the data using a quadratic estimator. Each measurement is assumed to carry errors distributed according to a Gaussian of given \( \sigma \). The parameters used for the LS experiment are summarised in Table 4.1.

In [12], a gradient-based technique is applied. In contrast to [125], both data and smoothness constraints (Equation 4.8 and Equation 4.9 respectively) are used to find the estimator error \( \epsilon \). A robust Lorentzian estimator \( \Phi = \log \left( 1 + \frac{\epsilon^2}{2\sigma^2} \right) \) is used. This approach is a multi-resolution approach. The following is a description of the multi-resolution method proposed in [12], the software of which has been used to obtain a dense optic flow in this work. However, the reader should be aware of a theoretical approach to multiresolution analysis, called the Renormalization Group Transform [126], where a model needs to be defined only for one level of the pyramid, since this is sufficient to define the 2 at each other level. In [12], at first a Gaussian pyramid consisting of \( L \) levels is built. At the coarsest level of the pyramid, the motion model parameters are set equal to zero. At finer levels of the pyramid the estimated motion parameters are propagated. This method requires the determination of three parameters, \( \sigma_1 \), \( \sigma_2 \) and \( \sigma_3 \), for each level of the pyramid. Parameter \( \sigma_1 \) is related to the estimator of the data constraint and parameters \( \sigma_2 \) and \( \sigma_3 \) are related to the smoothness constraint for
Chapter 4. Feature Extraction

Motion Model
Temporal model Linear (Equation 4.3)
Estimator Lorentzian

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L$</td>
<td>5</td>
</tr>
<tr>
<td>$S_1$</td>
<td>10.0</td>
</tr>
<tr>
<td>$S_2$</td>
<td>1.0</td>
</tr>
<tr>
<td>$S_3$</td>
<td>1.0</td>
</tr>
<tr>
<td>$s_1$</td>
<td>10.0</td>
</tr>
<tr>
<td>$s_2$</td>
<td>0.05</td>
</tr>
<tr>
<td>$s_3$</td>
<td>0.05</td>
</tr>
<tr>
<td>$f$</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Table 4.2: Parameters for the RE experiment.

The horizontal and vertical components of the optic flow. Let $S_1$, $S_2$ and $S_3$ indicate parameter values for the first level of the pyramid and $s_1$, $s_2$ and $s_3$ parameter values corresponding to the last level of the pyramid. At each level of the pyramid, the parameters $\sigma$ are decreased by a factor $f$. This method outputs a map of outliers to the data and the smoothness constraint [43]. The definition of outliers depends on the values of $s_1$, $s_2$ and $s_3$. The parameters used for the RE experiment are summarised in Table 4.2. This algorithm can also be used for top-down motion segmentation of multiple moving objects [43]. At each level of the pyramid, it outputs the dominant motion and selects the outliers for further analysis.

The results obtained are shown in Figure 4.2. In this Figure, the horizontal component of the displacement field for a couple of adjacent frames of the test sequence Renata is shown in shades of grey.

It is possible to see that the motion boundaries are rendered more accurately by the RE method. The LS method suffers from over-smoothing at the motion boundaries. The robust method shows sharp motion boundaries, while being smoother in the interior of the object.

Therefore, the robust formulation has a higher potential for accurate motion segment-
4.2. Motion Segmentation

Figure 4.2: Example of horizontal optic flow component. The translational motion flow is obtained in (a) using LS method and in (b) using RE method. Segmentation using the RSST framework for 500 regions in (c) in the case of LS method and in (d) in the case of RE method.

4.2.3 Motion Segmentation using the RSST

The dense optic flow produced by a motion estimator can be used for motion segmentation within the framework of the RSST. As an initialisation step, RSST requires a one-to-one mapping between any image pixel \( l \) and a feature vector \( \vec{v}_l^M(0) \), describing various attributes of the pixel at the iteration 0 of the process.

For motion-based segmentation, every pixel \( l \) is assigned horizontal and vertical components, respectively \( u_l(0) \) and \( v_l(0) \), of the dense optic flow [101], as expressed in Equation 4.12:
Chapter 4. Feature Extraction

\[
\tilde{V}_i^M(0) = [u_i(0), v_i(0)]
\]

(4.12)

It is worth noticing that, assuming \( I = n = (n_1, n_2) \), \( u_i(0) \equiv d_1(l) \) and \( v_i(0) \equiv d_2(l) \), according to the notation used in Section 4.2.1. The reason for this change of notation is the need to incorporate indices for the different iterations of the algorithm.

Each link between two generic vertices \( I \) and \( k \) at a generic iteration \( t \), is given by \( L_{k,l}^M(t) \) and defined according to Equation 4.13:

\[
L_{k,l}^M(t) = |u_k(t) - u_l(t)| + |v_k(t) - v_l(t)|
\]

(4.13)

Where \( | \cdot | \) is the \( L_1 \) norm. As indicated before, \( \tilde{V}_i^M(t) \) and \( L_{k,l}^M(t) \) relate to single feature or cue segmentation using motion measurements. Horizontal and vertical components of the optic flow field \( u(t) \) and \( v(t) \) have been normalised in order to have the same dynamic range. Choice of cost functions to represent link-weights and feature normalisation will be discussed in more detail in Sections 5.2.1 and 5.2.2 respectively.

Motion segmentation with the RSST framework is achieved as a bottom-up iterative process. At each iteration, the two most similar vertices in the graph \( G(t) \) are merged. This is achieved minimising the cost function representing the link between vertices:

\[
\min_{(k,l)} L_{k,l}^M(t) \quad \forall (k,l) \in G(t), \ (k \neq l)
\]

(4.14)

The vertices \((k, l)\) corresponding to \( \text{arg}(\min_{k,l}) \) are merged and vertex values are updated as in Equation 3.11. The links to the vertices adjacent to the newly formed vertex are also updated and circles within \( G(t) \) are deleted.

Experiments have been carried out within the RSST framework using a dense motion field from a least-squares motion estimator [125] and a robust motion estimator [12]. In Figure 4.3 the results of the segmentation performed on the test sequence Renata are shown. In Figure 4.3 (a) the result of the segmentation obtained using the RSST and a LS estimator is presented. In Figure 4.3 (b) the result of the segmentation obtained using the RSST and a RE technique is presented. As it is possible to notice from the
4.2. Motion Segmentation

Figure 4.3: Segmentation of dense motion field in (a) using a LS method and in (b) using a RE method. Both segmentations contain 100 segments.

Inspection of Figure 4.3, more meaningful results are obtained with the use of a robust motion estimator.

In Figure 4.4, the moving object segmentations obtained for different frames of test sequences Renata, Mobile and Calendar and Garden are shown. These segmentations have been obtained using a robust motion estimator. The segmentations are generally meaningful and consistent, especially the segmentations of the garden sequence, since the only motion is a translational motion of the camera. In the Mobile and Calendar sequence, different motions are present. Together with translational camera motion, the train is translating horizontally and the calendar is translating vertically. The ball is translating horizontally and rotating and, at around frame 40, it stops and it is moved by the train. All these objects are located by the segmentation.

More challenging for the analysis of motion is the test sequence Renata. There is only one foreground object in the scene, but its motion changes from purely translational
at a constant velocity to decelerating, stopping and then moving at almost the same speed as the camera. There is also rotational movement of parts of the object. This sequence proved to segment. The difficulty appears at around frame 40, where the object is rotating and translating at almost the same speed as the camera and where there is a global zoom of the camera.

The diagrams in Figure 4.5 show the error made during the segmentation of different frames belonging to the described test sequences. The performance is evaluated with the use of the MeanSquared Error (MSE) metric (see Appendix B, Definition B.1). The diagrams demonstrate that the performance decays as the number of regions approximating the scene decreases. They also provide a way of estimating the complexity of the sequence in terms of motion segmentation. In fact, they highlight the frames where the segmentation becomes problematic, therefore signalling when the assumptions encoded in the segmentation criteria are insufficient. This is particularly evident in Figure 4.5 (a), where the error made around frame 40 of the test sequence Renata is much higher than in the other frames of the same sequence.

From Equation 4.13, it can be seen that the segmentations presented within this framework are a piecewise optimal solution to motion segmentation. In this case, the motion model is represented by Equation 4.1. This assumption holds true in many practical situations. However, this model will not perform very well for object rotation or acceleration. The latter happens around frame 40 of the sequence Renata in Figure 4.4, for example. This problem could be addressed by using a more complex motion model as reported by [62]. In the cited work, motion segmentation is addressed within the RSST framework. During each iteration, a function representing a 2-D affine model, like in Equation 4.2, is minimised.

The relationship between these two different approaches to motion segmentation with RSST is to be found in the nature of RSST as a framework for producing piecewise approximations of the image attributes. In the approach presented by [62], the search is for an optimal piecewise-smooth approximation of the motion field, which is obtained by a higher order model such as the 2-D affine model. In the proposed approach, the search is for an optimal piecewise-constant approximation of the motion field, which is
Figure 4.4: Results of the segmentation using RSST on the basis of translational components of optic flow for test sequences Renata (first column), Mobile and Calendar (second column) and Garden (third column).
Figure 4.5: MSE of segmented motion fields for test sequences: (a) Renata, (b) Mobile and Calendar and (c) Garden.
4.2. Motion Segmentation

consistent with the assumption of a 2-D translational motion (compare with Equation 4.1).

4.2.4 Derivation of Motion Features from Translational Optic Flow Components

In Section 4.2.3, it has been highlighted how more complex models are to be used for covering the occurrence of motions that are not described completely by the 2-D translational model. The common approach to solving this problem is to use higher order parametric models. These have also been employed with the RSST framework. The main drawbacks of these models are two. They require a larger region of support than a single pixel, as used in the initialisation grid for segmentation. The computational complexity of updating the model at every interaction of the RSST procedure increases with the number of parameters in the motion model. For this application, it is necessary to describe more complex motions. It is required to have a dense field of motion features, so that the RSST can be initialised at the finest level. For computational reasons, the motion features should be calculated beforehand and then included in a simple optimisation criterion.

In [127], the use of features derived by translational components of the optic flow and describing rotation, translation and zoom in the compressed domain are reported. The features have been applied to the problem global motion compensation. In that work, the translational components of the optic flow are used to derive a descriptor of global rotation and zoom. In the present work, the same principles are applied to obtain a dense indicator of rotation. Let us take into consideration the rotation of an object in the 2-D plane of the image.

With reference to Figures 4.2.4 and 4.7, if a point belongs to an object undergoing rotation, it will follow a circular trajectory dependent on the distance of the point from the centre of rotation. The vector \( v = [v_x, v_y] \) representing the velocity of the point will be tangent to the trajectory. Let us consider two points \( P_1 \) and \( P_2 \) that are neighbours in a regular 2-D image grid and aligned in the horizontal direction. Let \( l \) be the distance between two adjacent pixels (in the vertical and horizontal direction) and let \( n l \) be the
Figure 4.6: Optic flow of a rotating object, taken from [6].

Figure 4.7: Schematic representation of rigid 2-D rotation of points $P_1$ and $P_2$ aligned horizontally.
distance of point $P_i$ from the origin of the reference system, so that $n$ is the abscissa of pixel $P_i$. Using $\phi$ as the angle of rotation:

$$v_1 = nl \tan(\phi)$$ (4.15)

and

$$v_2 = (n + 1)l \tan(\phi)$$ (4.16)

Thus,

$$v_1 - v_2 = l \tan(\phi)$$ (4.17)

and

$$\phi = \arctan \left( \frac{v_2 - v_1}{l} \right)$$ (4.18)

That is to say, in continuous terms and with sampling step $l$ in the image grid:

$$\phi = \arctan \left( \frac{\partial v_y}{\partial x} \right)$$ (4.19)

In the case of neighbours aligned in the vertical direction:

$$\phi = \arctan \left( \frac{\partial v_x}{\partial y} \right)$$ (4.20)

This leads to the physical measure of the curl of a vector field. It characterises the amount of rotation that the field is undergoing and it can therefore be exploited as a measure of rotation for the purposes of this application. For a field $F$ corresponding to a uniform rotation in the 2-D plane of the image as in Figure 4.8, the curl is expressed by Equation 4.21:

$$\text{curl} F = \left[ \frac{\partial F_x}{\partial y} - \frac{\partial F_y}{\partial x} \right] \hat{k}$$ (4.21)
Figure 4.8: Optic flow generated by uniform rotation.

Figure 4.9: Optic flow generated by uniform divergence.
where $F_x$ and $F_y$ are the components of the field in the horizontal and vertical direction and $\vec{k}$ is the direction perpendicular to the plane of the image.

The use of gradients (like curl and divergence) of the optic flow has been advocated to interpret three dimensional modelling of the scene by the HVS [128, 129].

In computer vision, first-order gradients of the optic flow have been employed for motion analysis [130]. In [127], the translational components of the optic flow are used to calculate the global camera zoom parameter. A field like the one occurring in the case of a global zoom (see Figure 4.9) occurs for an object moving in a direction perpendicular to the one of the image plane or for an object rotating about an axis lying on the image plane. It is therefore useful to obtain a quantity describing the occurrence of this effect locally in the flow, in order to characterise the object motion better. This is equivalent to computing the divergence of a field $F$ (a scalar quantity), which is expressed, for a two dimensional field, by Equation 4.22:

$$\text{div} F = \left[ \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} \right]$$

(4.22)

In addition to these motion features that broadly represent rotation and are computed on the optic flow associated to a single frame, it is possible to define a feature representing acceleration. Acceleration computed on an inter-frame basis according to Equation 4.23:

$$a(x) = \frac{\partial v(x)}{\partial t} = \left( \frac{\partial v_x}{\partial t}, \frac{\partial v_y}{\partial t} \right)$$

(4.23)

Inter-frame and intra-frame motion features will be used in conjunction with the translational components of the optic flow to complete the criteria for the segmentation within the RSST framework. The following notation will be used. Feature $c$ is the curl, computed using Equation 4.21 applied to the discrete image domain at a location $n$. Feature $d$ is the divergence calculated using Equation 4.22. Feature $a_x$ is the horizontal component of the acceleration and feature $a_y$ is the vertical component of the acceleration as defined by Equation 4.23. The subscripts $k$ and $l$ refer to two generic image pixels or image regions, as vertices in a graph. All the motion features have
been normalised so that they have the same dynamic range. In the definition of the cost function, the argument $t$, indicating the iteration, is omitted for brevity. The cost function representing the combination of different motion features is now expressed by Equation 4.24:

$$L_{k,t} = |u_k - u_t| + |v_k - v_t| + |c_k - c_t| + |d_k - d_t| + |a_{uk} - a_{ut}| + |a_{vk} - a_{vt}|$$

The cost function of Equation 4.24 has been used to perform a series of experiments to evaluate the information gained using rotational intra-frame and acceleration inter-frame motion features in addition to the usual translational components of the optic flow. The experiments presented in Section 4.2.3 have been repeated using the cost function of Equation 4.24. The results obtained have been compared with the ones obtained using cost function of Equation 4.13.

In the discussion, Equation 4.13 will be indicated as Equation A and Equation 4.24 will be indicated as Equation B. The results relative to Equation A will be indicated as obtained in the presence of translational motion features only. The results relative to Equation B will be indicated as obtained in the presence of multiple motion features.

The performance evaluation obtained for test sequences Renata, Mobile and Calendar and Garden is reported below. Figure 4.10 shows the comparison of the MSE obtained from the difference between the original dense optic flow and the segmented optic flow, indicated by $E_T$. $E_T$ is the error made approximating the translational components of the optic flow. It can be seen that the performances obtained from the use of Equation A and Equation B are quite similar. However, Equation A yields better approximations of the translational components of the optic flow.

Let us globally indicate curl and divergence of the optic flow as rotational features of the optic flow. Let us indicate with $E_R$ the MSE obtained from the difference between the original rotational features of the dense optic flow and the segmented rotational features of the optic flow. Let us indicate with $E_A$ the MSE obtained from the difference between the original acceleration calculated from the dense optic flow and the segmented acceleration.
Table 4.3: MSE of the segmentation using motion features for test sequences Renata, Mobile and Calendar and Garden.

Figure 4.11 shows a comparison of the MSE relative to $E_R$ and $E_A$ in the case of segmentations obtained from the use of Equation A and B. In both cases, the use of the cost function expressed by Equation B yields a more accurate segmentation of the original motion features.

Therefore, taking into account the overall error $TE$ obtained from the sum of errors $E_A$, $E_R$ and $E_A$, the use of multiple motion features results more advantageous than the sole use of the translational components of the optic flow. This is summarised in Table 4.3.

Finally, in Figure 4.12, the moving object segmentation of frames 34 – 38 of the test sequence Renata is presented. In the column on the left, the segmentation obtained with the use the cost function of Equation A is presented. In the column on the right, the segmentation obtained with the use the cost function of Equation B is presented. In this section of the sequence, the moving object presents a complex non-rigid rotational motion along an axis lying roughly in the vertical direction of the image plane. This movement is very gradual. It can be seen from inspection of Figure 4.12 how the segmentation with the use of the translational components only loses accuracy gradually. This is apparent at the boundaries on the right hand side of the object (from the observer point of view). With the use of the cost function of Equation B, the determination of the left hand side boundaries of the object improves notably.
Figure 4.10: MSE of translational optic flow components: comparison of performance of the cost function of Equation A (translational features) and of the cost function of Equation B (multiple motion features). Test sequences: in (a) Renata, in (b) Mobile and Calendar and in (c) Garden.
Figure 4.11: Comparison of performance of the cost function of Equation A (translational features) and of the cost function of Equation B (multiple motion features). $E_R$: in (a) Renata, in (c) Mobile and Calendar and in (e) Garden. $E_A$: in (b) Renata, in (d) Mobile and Calendar and in (f) Garden.
Chapter 4. Feature Extraction

Figure 4.12: Moving object segmentation obtained with the use of RSST: using the cost function of Equation A (translational features only) and using the cost function of Equation B (multiple motion features).
4.2.5 Estimation of Number of Moving Objects

A crucial decision relating to every segmentation procedure is the determination of the number of segments that the final segmentation will generate. In the case of segmentation within the RSST framework, this decision needs not to be taken in advance to the segmentation process. The representation provided by the SST contains all the data necessary to produce all possible segmentations for the given image, from the coarsest, at image level, to the finest, at pixel level.

The decision regarding the number of segments produced is not easy to define and is dependent on the target application. In many applications, like compression, the decision is related to the requirement of a minimum overall bit rate or a minimum acceptable distortion. The RSST offers a very flexible tool in this regard, because whatever level of distortion or bit rate required, the framework can provide the required segmentation, without any need to re-run the overall process.

In Figure 4.10, it is possible to see that a certain level of distortion is reached by different sequences and different frames in a sequence at different levels of segmentation (number of regions), as conditions can change with time. One possible approach is to fix the minimum error allowable by the segmentation and determine the granularity of the segmentation (number of regions) on that basis. This is a consideration that makes RSST highly advantageous for either unsupervised applications, where the specification is a given maximum distortion, or for supervised applications, where the user can specify the granularity of the segmentation according to his particular needs.

However, for object-based segmentation or for content-based applications, it is important to estimate the number of meaningful objects composing the scene, not the number of segments. This is a challenging task, as it is dependent on the definition of an object. An approach, that has been followed during this work, is to determine the number of objects by means of histogram mode seeking. In Figure 4.13, the histograms of the horizontal and vertical components of the displacement field ($d_1, d_2$) are plotted. They show that the estimation of the number of objects, on the basis of histogram information, can be quite difficult. This technique suffers from various well known drawbacks. For example, the peaks corresponding to objects must be well separated or very small.
Figure 4.13: 2-D histograms calculated on the basis of translational optic flow components (horizontal and vertical displacements). Test sequences: in (a) Renata, in (b) Mobile and Calendar and in (c) Garden.
objects become confused with the overall noise. For objects that are characterised by a rotational motion, the histograms of the optic flow will not offer any information. Even supposing that the peaks of the histograms are well defined, there is no one-to-one correspondence between the number of peaks detected and the number of regions to be specified in the segmentation process. For example, in the case of Garden, the number of dominant peaks is two. This is because the moving object divides the background in approximately two halves. The number of regions that needs to be specified to the program to obtain the segmentation into background and tree is therefore three.

The number of moving objects can also vary during a sequence, e.g. in the case where a moving object stops or one starts to move. This is exemplified in Mobile and Calendar, where on frame 40 the ball stops and then starts again, being moved by the train at the end of the sequence.

These examples show that, in order to determine the number of meaningful objects, it is practically very difficult not to resort to user interaction.

### 4.2.6 Background/foreground Labelling and Object Characterisation

A minimum requirement for dynamic sequence analysis is the detection of background and foreground objects, which often amount to classifying pixels as stationary or belonging to a moving object. The most common approach to obtaining such information is to derive a change detection mask from thresholding of frame differences [13, 81].

To achieve this, background pixels should be stationary in the absence of camera motion. Camera motion, however, frequently occurs, e.g. global translation, rotation or zoom [127]. It is possible to compensate for the global camera motion. In this project, an extension of the method proposed in [127], for the estimation of global camera motion parameters using a block-based motion estimator, is applied to a dense optic flow field. The global rotation angle is computed from the curl of the optic flow, while the global zooming factor is computed from the divergence of the optic flow. The global translation is apparent from the translational flow itself. The analysis of the histograms of translational flows, as well as curl and divergence fields, leads to a peak in the histogram corresponding to the global camera parameters. This is shown in Figure 4.14,
where a very clear peak corresponding to curl and divergence values of the background is present. In Figure 4.13, it is possible to see very clear peaks corresponding to the background translational movement. From the knowledge of the global camera parameters it is possible to compensate for the camera motion and therefore obtain a change detection mask by means of thresholding.

The approach followed in this work is to take into consideration the peak obtained from the 2-D histograms corresponding to: (a) the translational components of the optic flow, (b) the rotational features of the optic flow (calculated on the basis of curl and divergence) and (c) horizontal and vertical components of the acceleration. In the above examples, the histograms exhibit a clear peak $P$ in a 1-D domain. Let us assume the peak $P(x_p, y_p)$ corresponds to the horizontal and vertical components of values $(x_p, y_p)$ in a 2-D domain (the case for the 1-D domain is similar and simpler). Let us define a rectangular domain $D \in \mathbb{R}^2$ so that:

$$D = (x, y) \in \mathbb{R}^2 : |x - x_p| \leq T_x, |y - y_p| \leq T_y$$

(4.25)

Let $M_t$ be the partition of the image into $N$ segments, obtained by motion segmentation of a frame $F_t$, where $t$ indicates the position in time of the frame $F$. It is therefore $M_t = \bigcup_{i=1}^{N} M_{t,i}$, where $M_{t,i}$ are the segments constituting the segmentation, for $i = 1, \ldots, N$.
4.2. Motion Segmentation

and $M_{t,i} \cap M_{t,j} = \emptyset$. Let $A(M_{t,i})$ be the area of a generic segment $M_{t,i}$. It is possible to label each segment $M_{t,i}$ as belonging to the background or to the foreground, following the rule:

$$A(D \cap M_{t,i}) \geq sA(M_{t,i}) \rightarrow M_{t,i} \in B$$

where $s$ is a constant percentage that indicates a fractional part of the total area $A(M_{t,i})$.

In other words, if the size of the intersection of the segment $M_{t,i}$ with the domain $D$ is bigger than a percentage $s$ of the total area of the segment $M_{t,i}$, then the segment is said to belong to the background $B$. The background $B$ is given by the union of all the segments $M_{t,i}^B$ that are ruled to be in it by Equation 4.26. Therefore, the background $B$ results to be $B = \bigcup M_{t,i}^B$.

Thresholding is not a very accurate method on its own. In fact, it has to be linked to other refinement techniques to obtain an accurate mask. On the other hand, the approach used here is a projection of the change detection mask onto a given motion segmentation for labelling purposes. This means that the mask does not have to be particularly accurate. It only needs to locate the objects, whose boundaries will be found accurately using the correspondent motion segmentation.

In Figure 4.15, the application of Equation 4.26 for labelling background and foreground is explained for test sequence Garden. The labelling is obtained in this case using only the 2-D histogram of the translational components of the optic flow. Moreover, the labelling has been iterated for different, gradually widening values of the thresholds $T_x$ and $T_y$. This provides a layered segmentation of the sequence, where different labels correspond to different layers. The darker the label, the closer the motion of the layer to that of the background (shown in black). The boundaries between different layers are presented in black. It can be seen that by using segments for the labelling instead of acting directly on the domain $D$, noisy, irrelevant segments that are frequent in thresholding-based labelling are avoided.

Equation 4.26 can be used for labelling of objects that present a rotation or acceleration different to that shown by the background. This application is shown in Figure 4.16.
Figure 4.15: Layered representation of test sequence Garden with labels for different layers (first row) and borders of layers superimposed on original frames (second row).

Figure 4.16(a) shows the labelling obtained with the use of translational components of the optic flow for the test sequence Mobile and calendar. Figure 4.16(b) shows the labelling obtained with the use of acceleration for the test sequence Mobile and Calendar. Figure 4.16(c) refers to a later stage in the sequence, where another object, a mobile rotating on an axis positioned horizontally on the image plane appears at the left side of the frame. This object is localised accurately by the divergence feature, while the shape of the ball is also determined accurately, especially in the lower part. The difference in localisation accuracy between the upper and lower parts of the ball are due to the illumination of the scene. The information retrieved by the background/foreground labelling using multiple motion features can contribute to a broader characterisation of the motion of the objects. This is shown in Figure 4.17. Having three binary masks corresponding to the labelling according to three categories of motion (i.e. translation, rotation and acceleration), each object in the scene can have one of \(2^3 = 8\) possible labels, which characterise the motion. In Figure 4.17, black represents the background. The other labels correspond to either translation, rotation and acceleration or any mixture of these motions. From Figure 4.17, it is possible to gather, for example, that the train is translating and accelerating and the ball is translating, accelerating and
4.2. Motion Segmentation

4.2.7 Consistency Check

Until now, the only temporal information that has been exploited to obtain a moving object segmentation is the one produced by two consecutive frames. There can be instances in which this information is not reliable or becomes corrupted during the sequence. An example of this is given by the analysis of Figure 4.10(a), where it is possible to see that for the test sequence Renata, for frames 40 – 45, the error made during the segmentation increases notably and more rapidly than in the previous frames. This behaviour can be considered an indication of motion estimation unreliability.

One way to improve the consistency of the motion segmentation is to resort to the previous history of the segmentation, i.e. to store the results of the segmentation in a memory. This is the procedure adopted in [13] among others, where only pixels that were found to belong to the moving object in a fixed number of frames $n$ are labelled as belonging to the change detection mask. The use of a fixed length memory can be cumbersome, especially if the segmentation is reliable, and can only occasionally cause mistakes.

In the presented approach, the memory from a previous frame is used only when the segmentation error exceeds a user-dependent threshold. The memory consists of the projection of the frame at time $t$ onto the frame at the $t + 1$, assuming that the projection is a reliable approximation of object motion. The projection is compared with the actual segmentation at time $t + 1$. If the projection is more accurate than the actual segmentation, according to a certain error measure, then the projection is used to improve the segmentation. The projection is also used to generate another projection at the time $t + 2$ to iterate the process.

Let $M_t$ be the partition of the image of $N$ segments, obtained by motion segmentation of a frame $F_t$, where $t$ indicates the time reference of frame $F$. It is therefore $M_t = \bigcup_{i=1}^{N} M_{t,i}$, where $M_{t,i}$ are the segments constituting the segmentation, for $i = 1, \ldots, N$ and $M_{t,i} \cap M_{t,j} = \emptyset$. 
Figure 4.16: Background/foreground labelling using: (a) translational optic flow components, (b) acceleration and (c) rotational features of the optic flow.
4.2. Motion Segmentation

Let $M^O_t$ be a binary mask, which reliably labels stationary and moving pixels. Since $M^O_t$ is a labelling descriptor, its pixels can be defined as triplets $n(l) = (n_1, n_2, l)$, where $l \in \{0, 1\}$.

If $L$ is a generic binary mask, then $L = L_1 \cup L_0$, where $L_1 = \bigcup n(1)$ and $L_0 = \bigcup n(0)$.

Let us also define a generic segmentation $R = \bigcup_{g=1}^G X_g$, where $X_g$ is one of the $G$ segments composing the partition $R$. Let us define an area operator $a(\cdot)$, which returns the number of pixels belonging to a given set, segment or partition.

Let us also define a projection operation $\mathcal{P}(L, R)$ between a generic binary mask $L$ and a generic segmentation $R$, such that:

$$\mathcal{P}(L, R) \equiv L^P$$

(4.27)

and

$$L^P = \begin{cases} L^P_1 &= \bigcup_{g} X_g : a(X_g \cap L_1) > a(X_g \cap L_0) \\ L^P_0 &= \bigcup_{g} X_g : a(X_g \cap L_0) > a(X_g \cap L_1) \end{cases}$$

(4.28)

Let us suppose that the labels of $M^O_t$ are reliable and that the objects move from $t$ to $t + 1$ with the same velocity they moved between $t - 1$ and $t$. It is possible to predict a labelling for the frame $F_{t+1}$, indicated as $M^{OP}_{t+1}$. Pixel $n$ of $M^{OP}_{t+1}$ is then obtained as:
\[ n_{t+1} \in M_{t+1}^{OP}, \quad n_t \in M_t^O \quad \Rightarrow \quad n_{t+1} = n_t + d_t \quad (4.29) \]

where \( d_t \) is the displacement corresponding to pixel \( n_t \) at time \( t \).

This prediction relies on two assumptions:

1. The temporal model describing the trajectory is the linear model described by Equation 4.3.
2. The velocity of the moving object does not change substantially between sets of adjacent frames.

Using the predicted mask \( M_{t+1}^{OP} \), a hypothetical labelling of the frame \( F_{t+1} \) can be obtained, namely \( M_{t+1}^{OH} = P(M_{t+1}^{OP}, M_{t+1}) \).

It is possible to use the information contained in \( M_{t+1}^{OH} \) and \( M_{t+1}^{OP} \) to evaluate the reliability of the labelling provided by \( M_{t+1}^{OH} \). Let us calculate the error of the approximation of \( M_{t+1}^{OH} \) with \( M_{t+1}^{OP} \). This error is indicated as \( \epsilon(M_{t+1}^{OP}, M_{t+1}^{OH}) \). If \( T \) is a given threshold of the admissible error, the decision rule can be defined as:

\[ \text{if} \quad \epsilon(M_{t+1}^{OP}, M_{t+1}^{OH}) < T \quad \Rightarrow \quad M_{t+1}^O \equiv M_{t+1}^{OH} \quad (4.30) \]

and

\[ \text{if} \quad \epsilon(M_{t+1}^{OP}, M_{t+1}^{OH}) \geq T \quad \Rightarrow \quad M_{t+1}^O \equiv M_{t+1}^{OP} \quad (4.31) \]

If the objects in a scene change their motion smoothly, then the current motion segmentation is consistent and it is possible to exploit the previous frame labelling in order to generate a new labelling for the current frame. In case the difference between the predicted and hypothetical mask is above a given threshold, i.e. an indication of a sudden failure of the motion estimator, it is advisable to use the predicted (and presumably reliable) labelling.

These rules compensate for large failures of the motion estimator. If errors occur locally, especially along the motion boundaries, the consistency of the trajectory followed by
4.2. Motion Segmentation

the object can be analysed. Given a set of three consecutive frames $F_{t-1}$, $F_{t}$ and $F_{t+1}$, the minimum requirement for a consistent trajectory involves the conditions:

\[
\mathbf{n}_{t-1}(n_1, n_2, l = 1) \otimes \mathbf{n}_{t+1}(n_1, n_2, l = 1) \rightarrow \mathbf{n}_t(n_1, n_2, l = 1) \quad (4.32)
\]

and

\[
\mathbf{n}_{t-1}(n_1, n_2, l = 0) \oplus \mathbf{n}_{t+1}(n_1, n_2, l = 0) \rightarrow \mathbf{n}_t(n_1, n_2, l = 0) \quad (4.33)
\]

According to Rules in 4.32 and in 4.33, if a label $l \in \{0, 1\}$ is assigned consistently to a pixel at times $t - 1$ and $t + 1$, then these pixels will keep the same label at time $t$.

Let us take as an example the test sequence Renata, which exhibits a sudden increase of the motion segmentation error during frames 40 - 45. This is due to the complex motion and the changes in scene illumination, which render the motion estimation unreliable. This is shown by the peak of the error, in Figure 4.10(a). In Figure 4.18(a), the same effect is shown, this time in terms of the degradation of the properties of the moving object. In this figure, the size of the moving object, in pixels, is plotted against time or frame number. The average size of the object is calculated, taking into consideration all the frames except those in the interval 40 - 45. In fact, from inspection of the variation of the MSE, this section of the sequence is characterised by an unreliable motion field and the corresponding segmented objects would corrupt the overall estimate of the object size. The average object size is represented by the plain magenta line in Figure 4.18. Again, in this plot, it can be seen quite clearly that a large deviation of the object size from the average occurs. Let us therefore see whether checking for the consistency of the segmented object can be of help. In Figure 4.18 (b), the size of the projected objects is represented with a green line. It is possible to see that the approximation in size given by the projected objects is closer to the average object size than that of the segmented objects themselves. The approximation given by the projected object degrades over time. When the segmented object provides a closer fit to the average object size than the projected object, it is used as a reference for the motion segmentation again.
Chapter 4. Feature Extraction

<table>
<thead>
<tr>
<th>frame</th>
<th>$a(S)$</th>
<th>$a(P)$</th>
<th>$\varepsilon_S$</th>
<th>$\varepsilon_P$</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>48872</td>
<td>42657</td>
<td>5894</td>
<td>-321</td>
</tr>
<tr>
<td>41</td>
<td>51744</td>
<td>41406</td>
<td>8766</td>
<td>-1572</td>
</tr>
<tr>
<td>42</td>
<td>34576</td>
<td>40143</td>
<td>-8402</td>
<td>-2835</td>
</tr>
<tr>
<td>43</td>
<td>36160</td>
<td>38783</td>
<td>-6818</td>
<td>-4195</td>
</tr>
<tr>
<td>44</td>
<td>44088</td>
<td>37276</td>
<td>1110</td>
<td>-5702</td>
</tr>
<tr>
<td>45</td>
<td>44272</td>
<td>35890</td>
<td>1294</td>
<td>-7088</td>
</tr>
</tbody>
</table>

average size = 42978 pixels

Table 4.4: Size and error displayed by segmented and projected moving objects relative to the frames 40 – 45 of test sequence Renata.

This is summarised in Table 4.4, where sizes of the segmented and the projected objects are shown against the error made in the approximation of the object. This error is represented by the difference $\varepsilon_S = a(\cdot) - \bar{a}$, where $a(S)$ is the size (in pixels) of the area of the segmented object, $a(P)$ is the size of the area of the projected object, $\bar{a}$ is the average size of the segmented object calculated in all frames but 40 – 45, $\varepsilon_S$ is the error made using the segmented object mask and $\varepsilon_P$ is the error made using the projected motion mask.

A qualitative view of the gain achieved by switching from segmented to projected objects, for the problematic interval of frames, is shown in Figure 4.19. In the first column, the motion segmentation masks obtained with RSST segmentation using the cost function of Equation 4.24 are shown. It is possible to see that not even the use of a more complex motion model helps in such situations. The object actually stops moving or it moves at roughly the speed of the camera, so the motion estimator is unable to produce information. In the second column of Figure 4.24, the motion masks obtained by projection of a previous frame in the related frame interval is shown. Visually, the latter masks seem more accurate in the description of the object.

To summarise, in this approach it is suggested that the projection of the segmentation can provide more accurate information than the segmentation itself. There is no need to introduce a permanent memory in the system. The need for a consistency check is
4.2. Motion Segmentation

Figure 4.18: Size of the segmented moving object against the average size for frames 18 to 48 (a) and zoom of size of segmented and projected moving object against the average object size (b).

signalled by a change in the measure of the error. Using the error, it is also possible to judge if the projection is more accurate of the segmentation and vice versa. In this discussion, only two very simple measures of error have been considered, namely the MSE and the size of the object. However, more research should be directed towards the design of an effective measure of global properties of an object.

4.2.8 Role of Outliers

As seen in Section 4.2.2, one by-product of using a robust motion estimation method is the map of outliers. In the case of the robust motion estimation method used in this work (in [12]), the outliers are data which violate brightness and smoothness constraints. In [12], a Lorentzian robust kernel is used. If $\sigma_B$ and $\sigma_S$ are the scale parameters associated with the brightness and the smoothness constraints errors respectively, brightness and smoothness outliers are data with error exceeding $\tau_B = \sqrt{2}\sigma_B$ and $\tau_S = \sqrt{2}\sigma_S$ respectively [43]. In a multi-resolution top-down approach like the one in [12], where a dominant motion is selected at each level of the pyramid, brightness outliers are data which do not conform to the dominant motion model and are therefore selected for analysis in the next level of the pyramid. Brightness outliers can occur in
Figure 4.19: Moving object segmentation obtained with conventional labelling (first column) and using the projections (second column).
the presence of a scene illumination change or in the case of a rapid motion (since the brightness constraint is derived by a Taylor expansion of the intensity function which presumes that the \( \frac{da}{dt} \) and \( \frac{du}{dt} \), are small). More interesting for the accurate definition of motion boundaries are the smoothness outliers. The smoothness constraint forces each data point to have a similar motion to the average motion of its neighbourhood of a given size. Clearly, this constraint is violated at the boundaries of differently moving objects. Therefore, motion discontinuities are treated as smoothness outliers by robust methods. This kind of outliers can therefore be exploited in order to increase the accuracy of the definition of moving objects at their boundaries.

In the experiments carried out in this work with motion estimation process, outliers occur along motion boundaries and particularly along boundaries that are perpendicular to the motion.

Another useful characteristic of the outliers is that they represent a map of the occlusion and uncovering taking place during the motion of the objects.

Figure 4.20 exemplifies these concepts. In Figure 4.20 (a), the map of data outliers is presented, with the outliers in black. In Figure 4.20 (b), the position of the same outliers is presented in white and superimposed onto the original Renata sequence frame. It is possible to see that these outliers occur along the boundaries in the direction of the normal to the motion and that they show regions that have been uncovered or occluded by motion. It is also worth mentioning that outliers occur in textured regions of the image. Similar observations have also been made using the other test sequences.

Once a reliable labelling of the motion changes is available, as in Section 4.2.7, it is possible to use the knowledge of the location, the direction and the orientation of the motion boundaries to refine the binary masks corresponding to the labelling.

The concept used to refine the mask bears resemblance to the one explained in [7]. In this method, two adjacent frames \( F_t \) and \( F_{t+1} \) at times \( t \) and \( t+1 \) and the corresponding dense optic flow between them is considered. The change detection mask corresponds to the area of uncovered background and moving object in \( F_{t+1} \), while it also corresponds to the area of occluded background and moving object in \( F_t \). From the knowledge of the optic flow and the direction of the motion of the object, it is possible to label the
Figure 4.20: Example of the influence of outliers on the segmentation. In (a) map of outliers (in black) and in (b) their position in relation to the original frame (in white). In (c) moving object segmentation without taking into consideration the motion outliers (c) and in (d) with the use of a rule for the inclusion and/or discarding of outliers.
Figure 4.21: Technique for labelling uncovered and occluded areas, according to [7].

areas of uncovered and occluded background. Start and end points of the displacement vectors at the boundaries, along the direction of motion, identify the uncovered area from $F_t$ to $F_{t+1}$ and the occluded area from $F_{t+1}$ to $F_t$, as shown in Figure 4.21.

In the proposed work, uncovering and occluding areas in the labelled motion masks are localised using this method. They are not, however, immediately discarded. They are considered as search areas, i.e. areas in which outlier information should be taken into account in order to refine the segmentation. The robust motion estimator gives a strong indication of the reliability of the estimates in these search areas. The pixels that are classified as outliers are labelled outside the object area when they fall into the uncovering search area or inside the object area if they fall into an occlusion search area.

Let $B(n_b)$ be the oriented boundary of the moving region or regions of the current frame. The members of the boundary are pixels $n_b$ that belong to the foreground object but which share a 4-connected neighbourhood with one or more background pixels. They are considered to be vectors whose normals point from the object to the background. Let $d_{\perp}(n_b)$ be the component of the displacement vector $d(n_b)$ perpendicular to $B(n_b)$.

For each pixel $n_b$ belonging to the boundary, the extension of a search area is indicated
Chapter 4. Feature Extraction

Figure 4.22: Definition of areas of search (occlusion and uncovering areas) (a) and rule for inclusion and exclusion of connected components of outliers (b).

by $d_{\perp}(n_k) + \epsilon$, where $\epsilon$ is a constant that allows for a small error in the displacement vector. The area will be an occluding area if $d_{\perp}(n_k) \cdot n_k > 0$ and an uncovering area otherwise, where the operator $\cdot$ indicates the inner product of two vectors.

Figures 4.20 (c) and 4.20 (d) show the improvement in the object/background mask obtained with the rejection of outliers, using the procedure described above. In Figure 4.20 (c), the segmentation of the moving object is presented with the boundaries of the motion mask superimposed on the original frame. In Figure 4.20 (d), the same segmentation is presented, taking into account the outliers that are added or subtracted to the object segmentation, according to the procedure explained above. It is possible to see the improvement in the accuracy of the definition of the boundaries that lie on the direction perpendicular to the motion of the object. Here, areas that have been uncovered by the object motion have been eliminated from the segmentation, while areas that have been newly occluded by motion have been recovered and added to the segmentation, which is now much more refined overall. The same pattern of influence of outliers has been noticed in other test sequences.

All the possibilities of outlier occurrences and their location in relation to a moving object are shown in Figure 4.22. Outliers can be considered as connected components and only the ones that overlap with the object boundary are taken into account. There are four positions that the outliers can occupy in relation to the motion of the object,
4.2. Motion Segmentation

represented in the Figure 4.22 (b) with labels A, B, C and D. The case of outliers at positions A and D have already been discussed. This represents over ninety percent of outliers. In the case of the outlier at location B, these are locations that have been newly uncovered by the motion of the object and they do not belong to the segmentation of the object. They are therefore not taken into account. The case of location C represents an error in the segmentation, where the area has been included in the segmentation where it should not have been. Therefore, these outliers, i.e. the connected components that touch on the boundaries of the object and lie for more than a given percentage of their area inside the object segmentation, are to be subtracted from the segmentation. These cases represent less than 10 percent of the occurrences of outliers, but they are mentioned for generality.

4.2.9 Conclusions on Motion

In this section, a method for motion estimation, segmentation and labelling has been presented. It has been shown how the use of robust estimation can be more advantageous than the classic least-squares formulation for obtaining more accurately defined moving object boundaries. A method to exploit further information obtained from motion outliers has also been presented. Moreover, the information provided by the motion segmentation is enriched with some labelling, which allows for checking the consistency of the segmentation itself.

One interesting element emerged from the experimental work carried out on the analysis of motion. The use of motion not only provides the location of objects, but also reliable motion boundaries in the case where the region of support is, in qualitative terms, textured. This is due to the fact the motion estimation used here is a gradient-based method, so it cannot extract information from homogeneous regions, where the gradient is null. This fact has to be taken into account when quantitatively analysing the textural content of the sequence, which is the subject of the next section.
4.3 Texture Segmentation

Texture is a collective property of an image which has occasionally been modelled as a periodic repetition of elementary structural elements in the 2D lattice.

In the case of moving scene segmentation, in Section 4.2.1 it has been shown how motion estimation produces more accurate results on a textured region of support. In fact, textural measurements can be used to evaluate the reliability of motion estimates [88]. In the case of still picture segmentation, the presence of texture can cause over-segmentation of the pictures. This problem can be seen in various spatio-temporal segmentation approaches which do not rely on textural segmentation. The results are often inconsistent from a perceptual point of view. In fact, the human viewer perceives a textured region as a meaningful and homogeneous entity [113]. According to psychological studies and their application in the field of computer vision [114], there are two kinds of meaningful perceptual boundaries located in the scene, if only pre-attentive mechanisms are taken into account. These are the boundaries between perceived uniform and perceived textured regions, and boundaries between perceived different textures.

4.3.1 Role of Texture

This section introduces some concepts relating to textural feature extraction.

Texture is composed of elements which can be regarded as texture primitives (texels). A texel is viewed as a visual primitive with certain invariant properties. The main work related to texture analysis is to extract the relationship between primitives in order to recognise and classify texture. There are two broad classes of techniques dealing with these requirements. There is the structuring approach, which sees primitives as forming a regular pattern and therefore describing the pattern in terms of rules used to generate it. Alternatively, there is the statistical approach that describes texture with statistical rules governing the distribution and the relationship between grey-levels. The first approach works well with texture that exhibits much regularity and it is mostly used for synthesis in computer graphic. The second works better for natural scenes.
4.3. Texture Segmentation

Many textures in nature do not have a defined regularity, so they can be better described in statistical terms. Statistical pattern recognition is a paradigm to classify statistical variations in patterns. The pattern recognition approach is to classify instances of texture into a set of classes. Commonly used features for this purpose are: energy, entropy, correlation, inertia and local homogeneity. In [131], the use of fourteen different textural features is discussed.

The accurate location of textured areas of an image plays a fundamental role in supporting object-based segmentation for two main reasons. In the temporal part of the segmentation, the reliability of the motion information is estimated on the basis of textural content of the region of support [90]. In the spatial part of the segmentation, the extraction of areas presenting the same spatial pattern as a whole avoids oversegmentation and improves the effectiveness of the segmentation from a perceptual point of view.

Texture-based segmentation for video processing has specifications that set it apart from the conventional texture analysis for still image segmentation. Due to the volume of the data to be analysed in a relatively short amount of time, the technique must be fast. Therefore, classification-based techniques [86] present a computational disadvantage. Second, the accuracy of the location of the spatial region of support is more important than the classification of different spatial patterns.

State-of-the-art techniques for texture analysis use edge density information, co-occurrence matrices or wavelet filters to extract textural features. The features are estimated using a given window which causes spectral mixing of the textural patterns [6]. Alternatively, an approach based on mathematical morphology [132] is presented in this work. This approach offers the advantage to locate extremely accurately textured area boundaries, while being extremely fast and easy to implement.
4.3.2 Statistical Texture Description

Texture Feature Extraction

Textural features are computed using a window of observation in an image. Statistical approaches calculate statistical information about the distribution of grey-levels in the image in order to calculate a feature vector that can be processed with pattern recognition methods.

Co-occurrence matrices [28] describe the frequency of appearance of a pattern. They work well for a large variety of textures and describe spatial relationships between grey-levels. Co-occurrence matrices provide a quantitative description of the frequency of occurrence $P_{\phi,d}(a,b)$ of a certain pattern $(a,b)$ in the direction of the angle $\phi$ and at a distance $d$ [6]. Let us suppose that the texture has to be analysed in a rectangular portion of dimensions $N \times M$. Let $(k,l)$ and $(m,n)$ be two sets of coordinates belonging to the rectangular portion of the image. Let $| \cdot |$ denote a distance. Non-normalised frequencies of co-occurrence, as a function of angle and distance, are described in [6].

\[ P_{0,d} = \{|((k,l),(m,n)) : k - m = 0, |l - n| = d, f(k,l) = a, f(m,n) = b|\} \quad (4.34) \]

\[ P_{45,d} = \{|((k,l),(m,n)) : (k - m = d, l - n = -d) \text{or} \]
\[ (k - m = -d, l - n = d), f(k,l) = a, f(m,n) = b|\} \]
\[ (4.35) \]

\[ P_{90,d} = \{|((k,l),(m,n)) : |k - m| = d, l - n = 0, f(k,l) = a, f(m,n) = b|\} \quad (4.36) \]

\[ P_{135,d} = \{|((k,l),(m,n)) : (k - m = d, l - n = d) \text{ or} \]
\[ (k - m = -d, l - n = -d), f(k,l) = a, f(m,n) = b|\} \]
\[ (4.37) \]

Among the features derived in [28] from the co-occurrence matrix are:
4.3. Texture Segmentation

- Energy (or angular second moment), a measure of homogeneity:

\[
\sum_{a,b} P_{\phi,d}^2(a,b)
\]  

(4.38)

- Contrast, a measure of local variation:

\[
\sum_{a,b} |a - b|^2 P_{\phi,d}^\lambda(a,b)
\]  

(4.39)

- Correlation, a measure of linearity. Linear directional structures in the direction \( \phi \) result in a large value

\[
\frac{\sum_{a,b}(a - \mu_x)(b - \mu_y) P_{\phi,d}(a,b)}{\sigma_x \sigma_y},
\]  

(4.40)

where \( \mu_x \) and \( \mu_y \) are means and \( \sigma_x \) and \( \sigma_y \) are standard deviations.

\[
\mu_x = \sum_a a \sum_b P_{\phi,d}(a,b)
\]  

(4.41)

\[
\mu_y = \sum_b b \sum_a P_{\phi,d}(a,b)
\]  

(4.42)

\[
\sigma_x = \sum_a (a - \mu_x)^2 \sum_b P_{\phi,d}(a,b)
\]  

(4.43)

\[
\sigma_y = \sum_b (b - \mu_y)^2 \sum_a P_{\phi,d}(a,b)
\]  

(4.44)

Alternatively to spatial frequencies, edge frequencies can also be compared. In order to compute edges, standard edge detection masks or measures of gradient can be used. In [6], the following gradient measure is suggested for a distance \( d \) in a neighbourhood of an image \( f \):

\[
\frac{\sum_{a,b}(a - \mu_x)(b - \mu_y) P_{\phi,d}(a,b)}{\sigma_x \sigma_y}
\]  

where \( \mu_x \) and \( \mu_y \) are means and \( \sigma_x \) and \( \sigma_y \) are standard deviations.
Several texture features can be computed from the first and second order statistics of the edge distributions. Examples are coarseness (measured by the edge density), contrast and linearity. For a thorough description, the interested reader is referred to [6].

Texture Segmentation using the RSST

In order to exploit textural features for the application of texture segmentation with the RSST, it is necessary to define a feature vector and a cost function, or similarity measure, between features. The generic feature vector corresponding to a graph vertex \( k \) at the iteration \( t \) is described by Equations 3.3, 3.7 and 3.11. Let \( f_{i,k}(t) \), \( i = 1, \ldots, N \) be one of the \( N \) textural features constituting the description of vertex \( k \) at iteration \( t \). The textural features are calculated using a window of observation centred on a given pixel of coordinates \((x, y)\) and then assigning the value of the textural feature to that pixel. In this way the mapping of the image onto a dense graph is achieved. Each vertex of the graph corresponds to a set of textural features.

The literature of pattern recognition is particularly rich in distance measures to evaluate similarity between feature vectors and clusters. A full investigation into the different similarity figures is beyond the scope of this dissertation. The Minkowsky distance measure has been used to build the cost function between different links.

Let the cost function expressing the link weight between vertex \( k \) and vertex \( l \) at time \( t \) depend only on the textural features \( f_{i,k} \) and \( f_{i,l} \):

\[
L_{k,l}^T(t) = \left[ \sum_i |f_{i,k}(t) - f_{i,l}(t)|^r \right]^{1/r}
\]  

(4.46)

The experiments have used the Minkowski distance with \( r = 1 \), so that the same cost function used for texture could be easily compared or combined with those used in
4.3. Texture Segmentation

Figure 4.23: Segmentation with the RSST of test sequence *Renata*. In (a) using variance as the only textural feature and in (b) combining variance and grey-level intensity.

Section 4.2.3 on motion. The features $f_{i,k}$ representing different textural measurements of the vertex $k$ have been normalised to the same dynamic range of excursion.

The first series of experiments has exploited grey-level variance and a combination between the variance and grey-level value itself. Variance is a widely used measure of texture, most commonly used in video segmentation research [88]. The results of segmentation using the RSST framework with variance as the only feature are shown in Figure 4.23 (a). The results of the segmentation using the RSST framework with the combination of grey-level intensity and variance are shown in Figure 4.23 (b).

It is possible to see that the segmentation in Figure 4.23 (b) is better defined than that in (a). Therefore, the combination of intensity and variance is to be preferred. However, the boundaries are still rather inaccurate, due to the fact that the textural feature of variance is computed in a window. At the boundaries this causes blurring of the edges.

Another series of experiments has used co-occurrence features such as homogeneity, contrast and correlation. In Figure 4.24 (a), the co-occurrence features map for contrast is presented. In Figure 4.24 (b) the segmentation obtained with the use of contrast as only textural feature is given. Other experiments have been performed using homogeneity, contrast and correlation separately. For each of these features, four different directions at a distance $d$ are considered. The example in Figure 4.24 refers to a segmentation that
uses the features of contrast in the four different directions for a distance $d = 1$. Other segmentations, obtained with the use of intensity and correlation, yield similar results. The segmentation is qualitatively quite similar to the one obtained with the exclusive use of variance, for the same number of regions. Experiments using a combination of more textural features (variance, contrast, homogeneity, correlation) and grey-level intensity yield also similar results. In all attempted segmentations, blurring at the edges occurs. Additionally, a computational problem arises from having to consider, for the co-occurrence matrices, numerous features calculated for different distances and angles. This can make the cost function very complex and the amount of computation required for each iteration considerable.

Another series of experiments regarded the use of features related to the frequency of edges as opposed to the frequency of spatial patterns. A measure of the gradient was calculated using Equation 4.45 and used as a textural feature in the cost function 4.46 to provide an adaptive way of calculating edge density in a region. At the beginning, vertices which exhibit the same edge strength will be favoured to merge. However, as regions grow bigger, the value of the edge strength is averaged in the region, giving a measure of the density or local strength of the edges. The results obtained with such a cost function were not satisfactory. The results obtained by calculating the average of the feature in Equation 4.45 over a fixed size rectangular window centred in the current
Figure 4.25: Use of edge density. In (a) example edge gradient feature map and in (b) segmentation obtained with the RSST and using edge gradient as textural feature. In (c) edge density feature map and in (d) segmentation obtained with RSST using edge density as textural feature.

The use of edge density proved unsuccessful. However, edges and their density give an immediate qualitative assessment of the amount of texture. Let us consider the edge map in Figure 4.27, obtained with the used of the Sobel filter. Here, the density of edges clearly indicates the presence of texture. Since more computationally expensive methods do not dramatically improve the definition of region boundaries, a quick way of localising the textured areas of the image would be to compute the density of these
edges. This is simply the sum of edge pixels as highlighted in black in the map in Figure 4.27 over a window of a certain size. Obviously, the problem of the blurring at the edges of textured regions remains as well as the problem posed by areas of high contrast between plain homogeneous areas. One solution to these problems would be to define a second filter that locates and eliminates from the edge map the edges between high contrast homogeneous areas and between textured and plain areas. This can be done by defining a mask, as in Figure 4.26, which will be placed in the direction perpendicular to the edge and centred on the edge pixel. If the number of edge pixels inside the mask is lower than a certain threshold $T_1$ on both sides of the mask, than the edge pixel is part of an edge between two homogeneous regions and has to be eliminated. If the number of edge pixels inside the mask is lower than a certain threshold $T_1$ on only one of the sides of the mask and higher than another threshold $T_2$ on the other side, than the edge pixel is part of an edge between a homogeneous and a textured region and has to be eliminated. The edges located as high contrast edges or boundaries between textured and untextured areas are shown in Figure 4.27 (a) with a lighter shade of grey and the resulting edge map obtained eliminating such edge pixels is shown in Figure 4.27 (b).

Considering an edge map like the one visualised in Figure 4.27 (b), it is possible to localise regions of higher textural activity as the ones having a larger number of edge pixels than a certain threshold $T_3$ in a window of a given shape and size. This results in a binary mask, with the pixels belonging to a textured area highlighted in black in Figure 4.28 (a) and the boundaries of such regions corresponding to more highly textured areas shown superimposed in black in Figure 4.28 (b). The boundaries of such areas are not accurately defined. However, such a method localises the areas characterised by higher textural activity. This method is very simple and therefore is a
4.3. Texture Segmentation

Figure 4.27: Map of edges calculated using the Sobel filter: in (a) with the edges representing high contrast between homogeneous regions and between homogeneous and textured regions (in lighter grey) and in (b) the same mask where the edges representing higher contrast have been omitted.

A reasonable alternative to more complicated schemes that have been presented before.

4.3.3 Conclusions on Statistical Texture Segmentation

Texture in video processing is used as an indicator of the reliability of the motion estimate within a certain region of support. Therefore, the main objective that has to be met by the texture analysis, in the case of video processing, is to separate accurately textured and homogeneous areas of the image.

Statistical textural features, like the ones obtained using co-occurrence matrices, are needed in high numbers in order to provide a good characterisation of texture. This makes the task of combining textural and grey-level intensity features in the RSST framework computationally expensive.

Statistical features do not seem to provide a particularly accurate characterisation of the regions in terms of definition of boundaries. This is a well-known problem, due to the fact that the features are calculated in a window and mixing occurs in the windows at the boundaries between different regions.
In this case, an easier solution to the problem of localisation of textured areas is given by the calculation of the density of edges given by an edge detector. It has been shown that good localisation of textured regions can be achieved, albeit the accuracy of such boundaries is quite poor.

### 4.3.4 Morphological Texture Description

Mathematical morphology (MM) has been used for texture description in the case of binary images, while extensions of the method to grey-level images exist. The structuring elements can be a single pixel or more complex shapes like squares or lines. The application of morphological operations to images emphasises correlations between repetitive elements [6].

MM has been successfully employed for image simplification on the basis of the size of regions and local contrast [133], by using filters that preserve the geometry of perceptually important spatial features. It has also been advocated for region-based video coding [134]. Morphological transforms like watershed are indeed widely used for image and video segmentation [75, 74]. For a wider look at possible applications of MM, the interested reader is referred to [135].
4.3. Texture Segmentation

Here, a totally new application of MM is proposed, namely the use of MM operators on binary images for the purpose of texture segmentation, which accurately preserves the boundaries of the textured regions. This approach is inspired by granulometry and sieving analysis. Granulometry was first developed for the analysis of porous material, whose size characteristics were calculated using a sequence of opening operations [132]. It is still used as a tool to classify objects by size, especially in fields like materials engineering [136] and geology [137]. As observed in [6], MM-based approaches (that in the past were considered computationally expensive) can now be computed quickly and effectively, representing an implementation advantage for video applications.

Motivation

The approach proposed here takes into account an application of texture segmentation which tries to locate boundaries between textured and homogeneous regions in the image. From the psycho-visual point of view, two categories of boundaries are of importance in the pre-attentive stage of the segmentation, namely boundaries between textured and homogeneous regions and boundaries between different textures [113, 114]. The distinction between homogeneous and textured regions is sufficient for the video applications. The main purposes of texture in this analysis are:

- to locate the areas in which the motion estimates are reliable.
- to locate the areas that present higher textural activity.

The regions, which present higher textural activity, are the ones that will be problematic to segment using conventional grey-level segmentation techniques.

From this analysis, it is clear that texture analysis should produce a binary labelling into two classes of regions, namely the ones that show uniformity and the ones which show high textural activity. However, an essential requirement is to detect precisely the boundaries between these two classes of regions.
Morphological Operations

This section illustrates some concepts of MM useful in the progression of the explanation of the method [6].

A morphological transformation $\Psi$ is given by the relationship between an image (set $X$) and smaller set $B$, called the *structuring element*. $B$ is expressed with respect to another point $O$ called the representative point. When the transformation $\Psi(X)$ is applied to $X$, the structuring element $B$ is moved exhaustively along the image. The position of the origin of the structuring element into the image is called the current pixel. The result of the application of the relation to the current pixel can be either zero or one.

The morphological transformation dilation $\oplus$ combines two sets using vector addition (e.g. $(a, b) + (c, d) = (a + c, b + d)$). The dilation $X \oplus B$ is the set of all possible vector additions of pairs of elements, one from each of the sets, namely:

$$X \oplus B = \{ p : p = x + b, x \in X, b \in B \} \quad (4.47)$$

In the case where the structuring element behaves the same way in all directions, dilation is equivalent to an isotropic expansion.

Erosion $\ominus$ combines two sets by vector subtraction of set elements:

$$X \ominus B = \{ p : p + b \in X \text{ for every } b \in B \} \quad (4.48)$$

Erosion followed by dilation creates a transformation called opening, namely:

$$X \circ B = (X \ominus B) \oplus B \quad (4.49)$$

Dilation followed by erosion creates a transformation called closing:

$$X \bullet B = (X \oplus B) \ominus B \quad (4.50)$$
4.3. Texture Segmentation

Opening and closing with the same structuring element are used to eliminate smaller insignificant details in the image while the global shape of the object remains unchanged. The closing operation, in particular, connects objects that are close together, filling holes and gaps and smoothing the object outlines.

Texture Segmentation by Mathematical Morphology

In order to use MM, the image needs to be binarised. In Section 4.3.2, the importance of edges in the characterisation of textural activities has been discussed. In this approach, edge pixels are set to 1 and all the other pixels to 0, thus forming a binary representation of the edge mask. Canny edge detection is particularly robust to noise and is therefore a good choice for this application [138]. Moreover, thanks to the mechanism of hysteresis thresholding [6], weak edges connected with strong ones are selected, improving the localisation of continuous and closed object boundaries.

The textural analysis starts with the extraction of edges in an image frame, using the Canny edge detector [6], using the same parameters for all test sequences and experiments. The parameters used have been $\sigma = 1$ for the Gaussian filter and size of filter $S = 7$. In Figure 4.29, the edges extracted using this method are pictured.

The regions that would be intuitively defined as textured, as opposed to homogeneous or uniform or flat, are the areas of greatest edge density. The edge density can be defined as the number of edges in a defined area of the image. However, exploiting this definition to locate textured areas would require setting up a suitable window in size and shape and, more importantly, a threshold representing a divide between textured and homogeneous regions. This would re-create the problems already highlighted in Section 4.3.2.

An alternative method, in comparison to these classical approaches to the calculation of textural features, is needed. A high density of edges implies that the distance between edges is relatively small. Therefore, a dilation operation would extend the support of isolated edges until they merge with adjacent edges. This would produce a region of support for locating a textured area. At the same time, two effects should be avoided. First, the transformation of isolated edges into textured regions and second
Figure 4.29: Results of Canny edge detection for sequences: (a) Renata, (b) Mobile and Calendar and (c) Garden.
the extension of the support of the textured regions beyond their external boundaries. This is achieved by following the dilation operation with an erosion operation using the same structuring element. In MM, the sequence of dilation-erosion using the same structuring element is known as closing.

In Figure 4.30, the result of the morphological closing, using a disk of 3 pixels radius as a structuring element, is shown. In Figures 4.30 (a), (c) and (e), the result of the morphological operation corresponds accurately to the more texturally active areas of the images. Some holes inside such textured areas remain. Therefore, connected components, contained inside textured areas and not exceeding a maximum size fixed by a given threshold T (in the case of these experiments T = 100 pixels), are labelled as textured and incorporated in the corresponding textured connected component. This operation has been indicated as filling in Figure 4.30 and is shown in Figures 4.30 (b), (d) and (f). In Figure 4.31, the boundaries of the textured areas obtained as a results of the morphological analysis are superimposed to the original frame to show the accuracy of textured area boundaries. The definition is extremely precise, as it is possible to see in Figures 4.31 (a), (c) and (e). In Figures 4.31 (b), (d) and (f), the textured areas have been masked to show the resulting homogeneous connected components. These areas correspond to intuitive homogeneous areas and are precisely defined.

Multi-scale Morphological Texture Analysis

In the Section 4.3.4, MM and edge detection have been used for effective and simple location of textured regions. This is suitable for the purpose of moving object segmentation.

The results shown have been obtained using a structuring element constant in shape and size. Indeed, in the experiments made, for a given resolution of images, a fixed shape and size of the structuring element has yielded perceptually plausible results in the location of textured regions.

However, it is possible to vary the shape and size of the structuring element and obtain a description of the texture present in the scene. Let us first take into account a structuring element of a given shape, e.g. a disk, whose radius varies. Let $0 = r_0 <$
Figure 4.30: Results of morphological closing before and after the filling of the holes (the holes are indicated in light grey) respectively for (a) and (b) Renata, (c) and (d) Mobile and Calendar and (e) and (f) Garden.
Figure 4.31: Outlines and spatial support for textured areas of the image frames respectively for: (a) and (b) *Renata*, (c) and (d) *Mobile and Calendar* and (e) and (f) *Garden*. Outlines and spatial support are indicated either in black or white for visualisation purposes.
\( r_1 < r_2 < \ldots < r_n \) be a set of radius dimensions. The corresponding set of structuring elements is \( S_0 < S_1 < S_2 < \ldots < S_n \). From the application of each structuring element, it is possible to obtain \( n \) binary masks. Each binary mask describes a texture at a different scale, obtaining a multi-scale description, from the finest granularity, to the coarsest one, using structuring elements of increasing radius. The multi-scale description can be summarised by a set of \( n + 1 \) different labels.

It is possible to use different shapes as well as different sizes of the structuring element. The following set of experiments has been performed using a set of different shapes, namely disks and lines at different orientations. A set of disks of different radii \( r_i \) for \( i = 1, \ldots, 12 \) and a set of lines of different lengths \( l_i \) for \( i = 1, \ldots, 12 \) and different orientations \( \phi_i \) for \( i = 1, \ldots, 4 \), (\( \phi_1 = 0 \), \( \phi_1 = 45 \), \( \phi_1 = 90 \) and \( \phi_1 = 135 \)) have been used in the experiments. The idea is to extract the structuring elements composing the texture in a multi-scale form. The experiments have been performed on texture samples from the Brodatz collection [139] and on synthetic pictures obtained by combining samples from the Brodatz collection.

This technique, despite its simplicity, is particularly effective in the segmentation of the repetitive elements, highlighting the structure of the texture.

The findings of the experiments are summarised below.

- **Multi-scale analysis:** In the case of textures composed of ensembles of elements characterised by similar shapes but different scales, an effective multi-scale segmentation can be obtained by using a bank of structuring elements. The shape of the filters is kept constant, while the size varies. An example is given in Figure 4.32.

- **Directional analysis:** As well as the use of MM for multi-scale analysis, directional analysis has also been investigated. Directional analysis can be performed by using structuring elements which share the same shape (e.g. a line of a given length), but have differing orientations in the 2D space. Directional analysis can be combined with multi-scale analysis. Examples of this application are shown in Figure 4.33.
• **Analysis of regular textures**: Segmentation of textures composed of a regular repetition of elements characterised by a simple geometric shape is one of the applications that have been pursued more often in the past. MM analysis is particularly suited for this purpose, as is shown by the results reported in Figure 4.34.

• **Analysis of irregular textures**: Segmentation of textures composed of irregular elements is more challenging. Therefore, this application has not been explored in much detail in the present work. However, it is also possible to separate elements which exhibit roughly same shape and size, but are characterised by an irregular distribution of different textures, as shown in Figure 4.35.

• **Application: Video Processing**: The main purpose of the use of texture analysis for video object segmentation is to discern between textured and homogeneous areas. This is an application in which MM excels, as amply discussed earlier in this Section. Another such example is given in Figure 4.36.

• **Application: Texture Classification**: The main application of textural analysis for segmentation is regarding the separation of textures belonging to different textural classes. Efficient segmentation of multiple textures is required. In order to test the performance of the proposed method, the MM analysis has been applied to synthetic pictures resulting from the composition of differently textured samples. The results shown in Figure 4.37 demonstrate the possibility of applying MM to this problem with some success.

**Utilisation of Morphological Signatures with the RSST**

The possibility of associating different signatures to different textures by a combination of morphological masks and of accurately defining texture boundaries, suggests a modification of the merging strategy in the RSST framework. The region growing techniques have often been criticised because they do not offer flexibility in the merging criteria. In the presence of texture, one would like to segment the pixels using grey-level similarity in the regions that have low textural content, while grouping pixels that belong to the
Figure 4.32: Examples of the use of morphological filters in order to segment structuring textural elements of same shape but different size.
4.3. Texture Segmentation

Figure 4.33: Examples of the use of morphological filters to segment directional structuring elements.
Figure 4.34: Examples of the use of morphological filters to segment structuring textural elements in regular textures.
4.3. Texture Segmentation

Figure 4.35: Examples of the use of morphological filters to segment structuring textural elements in irregular textures.

Figure 4.36: Examples of the use of morphological filters to separate homogeneous and textured regions in an image.
Figure 4.37: Examples of the use of morphological filters to segment different texture samples.
4.3. Texture Segmentation

same texture classes. Moreover, one wants to avoid over-segmentation due to textural content and leakage due to similarity in grey-levels between small regions belonging to different texture classes.

In order to fulfil these requirements, the criteria for merging according to the texture signature associated with each pixel have been modified.

Pixels characterised by a different texture signature are kept separated, such that merging between members of the different textures takes place according to different criteria. The similarity between grey-level values of pixels is not as important in highly textured regions as it is in homogeneous ones. It is necessary, then, to decrease the importance of grey-level similarity in highly textured regions and to favour the merging of pixels belonging to regions characterised by the same textural signature.

First, to keep the pixels belonging to different textural classes separated, an identifier is associated with each different class. Merging is allowed only between pixels or regions belonging to the same textural classes or to regions and pixels belonging to classes that have neighbouring signatures, if no other merging inside the same classes is possible.

Two merging strategies have been considered.

Strategy 1

Since similarity between grey-level is important only among homogeneous regions, the first strategy implemented is to use a weighted cost function between regions. The grey-level dissimilarity is diminished by a quantity which is dependent on the activity described by the textural signature.

Every node in a textural class described by the signature $s_i$ is weighted by a quantity $W(s_i)$, so that the cost function between vertices $k$ and $l$ characterised by the same signature $s_i$ is:

$$L_{k,l} = \frac{|I_k - I_l|}{W(s_i)} \quad (4.51)$$

$W(s_i)$ increases with the increase of the textural signature, diminishing the importance of contrast in highly textured regions.
For vertices \( k \) and \( l \) belonging to adjacent textural classes:

\[
L_{k,l} = \frac{|W(s_l)I_k - W(s_k)I_l|}{W(s_l)W(s_k)}
\]  

(4.52)

Let us suppose that the regions \( l \) and \( k \) are characterised by the same grey-level average \( I_l \). Using the conventional cost function, the two regions would have the same average and would be merged immediately. Using the new weighting system, the cost function is proportional to the difference in the textural weighting.

Strategy 2

Another consideration to take into account is that textural regions are made up of a repetition of regions of constant size that can have large differences in grey-level, hence the high density of edges. This means that the size of the regions is an increasingly important criterion for merging as the textural content of the region increases. Another strategy (strategy 2) is that of merging according to grey-level similarity until a certain level of similarity \( \Delta I \), then merging according to the size difference of the regions, i.e. favouring the creation of regions that have similar sizes. The threshold \( \Delta I \) varies according to the textural signature. It is set to 255 for homogeneous regions, so that homogeneous regions are merged according to the conventional criterion. However, making \( \Delta I = \Delta I(s_i) \), it will decrease as the signature \( s_i \) increases, so that for highly textured regions the constituent elements can be very small and size is a more important criterion for merging:

\[
L_{k,l} = \begin{cases} 
|I_k - I_l| & \text{if } |I_k - I_l| \leq \Delta I(s_i) \\ 
\min_{k,l}(A) & \text{otherwise} 
\end{cases}
\]  

(4.53)

Where \( A \) is the area of a region and \( \min_{k,l}(A) \) is the minimum of the two areas corresponding to regions \( k \) and \( l \).
4.3. Texture Segmentation

Experiments with the RSST and Strategies 1 and 2

The results of the application of the two strategies presented above are summarised in Figures 4.38 and 4.39. In Figure 4.38, the diagrams of the performance of the two alternative strategies are compared with the conventional cost function in terms of Peak Signal-to-Noise Ratio (PSNR, defined in Appendix B). It must be emphasised that the PSNR is not an ideal measure to evaluate performance in this case, since the conventional cost function should always produce a better result in terms of PSNR. The alternative strategies proposed are effectively a violation of the criterion of minimisation of SNR. However, the second strategy performs better in two of the three cases proposed. In other series of experiments, the analysis of the performance shows that the higher the textural activity of the scene, the better the performance of the second strategy proposed. This is because this strategy avoids the leakage phenomenon, which is a well known problem of the region growing methods. If a link between two regions is noisy or erroneous, i.e. there is a small but very similar bridge between two regions which are otherwise different, the two regions are merged regardless. This results in a large error in performance. This is particularly evident in the sequence Garden, where the tree is easily merged with the flower-bed because of small elements in the texture of the flower-bed itself. The merging of these two big regions creates a large overall error in the performance. This error is avoided by using the proposed strategies.

Figure 4.39 shows that the employment of the two alternative strategies avoids the formation of small insignificant regions which exhibit high contrast. The two strategies allow the conservation of meaningful entities in the image. Examples are: the flower-bed, now separated from the tree, and the tree, separated from the sky, in Garden, the arm, separated from the calendar, in Renata, the calendar, separated from the background, and the train, separated from the calendar, in Mobile and Calendar. Additionally, the segmented images obtained by these strategies are more pleasing to a human observer.
Figure 4.38: Comparison of performance between the conventional RSST segmentation and the modified segmentation using strategies 1 and 2. Test sequences: (a) Renata, (b) Mobile and Calendar and (c) Garden.
Figure 4.39: Comparison of performance between (a) the conventional RSST segmentation and the modified segmentation using (b) strategy 1 and (c) strategy 2. All the segmentations contain 4000 segments.
4.3.5 Conclusions on Morphological Texture Segmentation

This section has explored the possibility of characterising the textural content of the image in a simple way which has the advantage of preserving the boundary information. This is a novel method which makes use of mathematical morphology. The application of the morphological closure of an edge map is a powerful and simple technique to obtain a binary separation between perceptually plausible textured and homogeneous regions of the image. The result is a region of support which is particularly accurate. This region can be used for further elaboration as a region of support for adaptive extraction of features of interest, like variance.

In the experimental section, the application of masks of different shapes at different scales allows the association of textural signatures with texture classes. The use of these textural signatures associated has been incorporated in flexible and adaptive merging criteria to use within the RSST framework. These criteria have proved effective in avoiding the leakage phenomenon, a drawback of region merging techniques.

4.4 Colour Segmentation

The original formulation of the RSST has been expressed with regard to grey-level segmentation of still images. In [117], the application of the RSST had been extended to the realm of colour images. In this section, extension of RSST to colour segmentation is discussed, with emphasis on colour spaces useful for digital video technology.

Let us summarise some useful notions about colour theory [8]. When addressing the segmentation using grey-level intensity and colour attributes, it is necessary to relate them to luminance. Luminance is often referred to as grey-level intensity in the 8-bit representation of images. Luminance (Y) is directly proportional to the intensity, but weighted by the spectral sensitivity of human lightness perception. Intensity refers to the power flow in a specified region of the electromagnetic spectrum. The magnitude of luminance is proportional to physical power, therefore it is like intensity, but the spectral composition of the luminance is related to the lightness sensitivity of the human
vision. Luminance can be computed as a properly weighted sum of red, green and blue linear tristimulus components.

The Commission Internationale de l’Eclairage (CIE) standardised $L^*$ to approximate the lightness response of the human visual system. $L^*$ is a power of luminance with a linear segment close to black. $L^*$ is defined as one component of a uniform colour space. Perceptually uniform colour spaces indicate the fact that, if a small perturbation to a component of the colour value occurs, this is approximately perceptible across the dynamic range of that value in the same way. Perceptually uniform colour spaces have not been taken into account in this work. However, the lightness sensation of vision is to be taken into account, if an image has to be coded so as to maximise the perceptual quality of the displayed image. The non-linearity of a cathode ray tube (CRT) display is very similar to the inverse of the lightness sensitivity of the HVS. Therefore, applying the gamma correction to the intensity signal is a good way of taking into account perceptual characteristics of the signal (refer also to Appendix A). For a CRT display the following relationship holds:

$$I = V^\gamma$$  \hfill (4.54)

where $I$ is the intensity and $V$ is the voltage and $\gamma$ is a constant. In practical terms:

$$I = V^{2.5}.$$  \hfill (4.55)

In order to recover from this effect of non-linearity, which is incidentally consistent with the human visual system sensitivity to lightness contrast, the following correction has to be applied:

$$S = I^{1/\gamma},$$  \hfill (4.56)

where $S$ is the corrected signal.

Colour is the perceptual result of light having wavelengths from 400 nm to 700 nm being incident upon the retina. The human retina has three kinds of colour photoreceptor, called cones. Therefore, three numerical components are enough to describe colour, provided that appropriate spectral weighting are applied. The CIE has defined several systems to map the spectral power distribution (SPD) to a triplet of numerical components that are the mathematical coordinates of a colour space.
The systems useful for colour specification are all based on CIE XYZ. Y indicates luminance and X and Z are chrominance components. The XYZ tristimulus is calculated from the integral of the spectral power distribution of a source, therefore their spectral distribution corresponds to colour matching by HVS [8]. The useful systems, from the image coding point of view, are linear RGB and non linear (gamma corrected) R'G'B'. For the definition of R'G'B', the interested reader is referred to Appendix A.

In the following sections, the term luma and the symbol Y' will refer to the gamma corrected (non-linear) luminance information, according to human contrast sensitivity.

4.4.1 Colour Segmentation with the RSST

When choosing the colour space to code the information, the distance metric that is going to be used in order to express the link weight between different vertices must also be considered. The vertex information can be expressed in the form of a vector containing a colour triplet.

As link weight, the Euclidean distance (a special case of the Minkowsky distance) between colour vectors is widely used in works on colour segmentation [117]. Therefore indicating two adjacent vertices in a graph as k and l and the colour triplet associated to each vertex at the time (iteration) t as \( \mathbf{c}(t) = (c_1(t), c_2(t), c_3(t)) \), the generic link weight used to apply the RSST framework is expressed as:

\[
L_{k,l}(t) = \sqrt{(c_{1,k}(t) - c_{1,l}(t))^2 + (c_{2,k}(t) - c_{2,l}(t))^2 + (c_{3,k}(t) - c_{3,l}(t))^2}
\]  

4.4.2 Colour Spaces

Colour spaces that are useful for colour coding according to the definition given in [8] will be taken into account.

In Section 4.57, the Euclidean distance has been indicated as a useful objective colour metric. In this work, the Euclidean distance has been used with two colour spaces: the RGB and the Y' Cb Cr.
4.4. Colour Segmentation

The RGB colour space is very useful and used in computing [27].

The \( Y' C_B C_R \) space is also of practical interest in the context of this work since it is used for digital television [8]. This coding space is motivated by the observation that most luminance \( Y \) information consists of green stimuli (between 60 and 70 percent [8]). Therefore, it is possible to base the chrominance coding on the other two primaries, blue (B) and red (R). One wants to remove the B and R information from the Y and this is done simply by subtraction. The \( C_B \) and \( C_R \) components are related to the differences \( (B' - Y') \) and \( (R' - Y') \) respectively. The reason for the use of gamma corrected \( (B' - Y') \) and \( (R' - Y') \) is that \( Y'C_B C_R \) is a coding space for digital compression and television. For television applications, gamma correction is applied automatically at the camera end [8] (while at the display end, the gamma correction should be set up and performed by the user).

In Figure 4.40, the excursions of the chrominances of \( Y'C_B C_R \) colour space are represented in relation to the corresponding values of R, G, B and cyan (Cy), Magenta (Mg) and Yellow (Yl). Only the chrominances are represented. According to Rec. 601 (see Appendix A), the excursion for \( C_B \) and \( C_R \) is from 16 to 240, for the differences \( (B' - Y') \) and \( (R' - Y') \) in the range \( [0, \ldots, 1] \). For the reference white, where \( R', B' \) and \( Y' \) are equal to 1, \( C_B \) and \( C_R \) are equal to 128. For the reference black, where \( R', B' \) and \( Y' \) are equal to 0, \( C_B \) and \( C_R \) are equal to 128. If \( B' = 1 \), then \( C_B = 240 \). If
Table 4.5: PSNR for different colour spaces using the Euclidean distance on RSST framework. The segmentations consist of 4000 segments.

\[
\begin{array}{|c|c|c|c|}
\hline
\text{Colour space} & \text{Renata} & \text{Mobile and Calendar} & \text{Garden} \\
\hline
Y' C_B C_R & 25.198 \text{ dB} & 26.311 \text{ dB} & 22.873 \text{ dB} \\
R' G' B' & 21.983 \text{ dB} & 18.930 \text{ dB} & 22.577 \text{ dB} \\
\hline
\end{array}
\]

The idea of texture characterisation based on morphological operations using structuring elements of differing sizes has been further extended to the analysis of colour textures.

In order to make the analysis as presented in Section 4.3.4 feasible, it is necessary to extract colour edges. This has been done applying the Canny edge detector to the three colour components constituting a colour image.
4.4. Colour Segmentation

<table>
<thead>
<tr>
<th>Colour space</th>
<th>Renata</th>
<th>Mobile and Calendar</th>
<th>Garden</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Y'_{CB}C_R)</td>
<td>25.8616 dB</td>
<td>23.0647 dB</td>
<td>24.674 dB</td>
</tr>
<tr>
<td>(R'G'B')</td>
<td>22.9502 dB</td>
<td>19.2116 dB</td>
<td>23.1807 dB</td>
</tr>
</tbody>
</table>

Table 4.6: PSNR using colour texture characterisation on RSST framework. The segmentations consist of 4000 segments.

It is possible to apply \(n\) different structuring elements to the three colour edge maps, indicated as \(A\), \(B\) and \(C\). From the application of each structuring element, a binary mask is obtained. Each mask contains two labels, which can assume values \((0,1)\). It is possible to obtain a set of \(3 \times 2^n\) colour edge labels, that are indicated as \(A_i\), \(B_i\) and \(C_i\) maps for \(i = 1, \ldots, n\). Indicating with \(L\) the number of labels obtained from the application of the different structuring elements to each map, these maps can be combined to obtain \((L/2)^3\) texture labels, each label indicating a particular mixture of colour textures and therefore a particular class of colour textures.

In Figure 4.41, the labelled maps obtained with the combination of one binary mask (and therefore two labels) for each colour component are shown. The results are presented for two colour spaces, the \(R'G'B'\) and \(Y'_{CB}C_R\). In fact, different edge maps are obtained for different colour spaces. Some edges are more relevant in one colour space than the other and therefore different types of characterisation are possible. From visual inspection of the colour texture labels for the two spaces, colour textures are more differentiated in the \(Y'_{CB}C_R\) than in the \(R'G'B'\) space. For the interpretation of the colour codes used in Figure 4.41, a summary of the colour labels, combinations and corresponding codes is presented in Table 4.7.

This characterisation of the colour textures is used for segmentation purposes. Let us take into consideration the simplest case of analysis, which is the one of the combination of labels shown in Figure 4.41. In total, \(2^3\) different colour texture classes are present. The Euclidean distance is used, but with one rule. To avoid pixels or regions belonging to different colour texture classes from merging, despite the individual colour similarity, every vertex is assigned a label identifying the colour texture class to which it belongs. The segmentation is carried out with the normal framework, except that
Table 4.7: Colour codes for texture description using $n = 1$ maps for each colour component, like in Figure 4.41. For experiments with $R'G'B'$, $C_1$ corresponds to $R$, $C_2$ corresponds to $G$ and $C_3$ corresponds to $B$. For experiments with $Y'CBR$, $C_1$ corresponds to $C_R$, $C_2$ corresponds to $Y'$ and $C_3$ corresponds to $C_B$.

<table>
<thead>
<tr>
<th>Label</th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$C_3$</th>
<th>Code in Figure 4.41</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Black</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Blue</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Green</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Yellow</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Red</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Magenta</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Cyan</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>White</td>
</tr>
</tbody>
</table>

Only elements belonging to the same colour texture are allowed to merge. Using this rule, the segmentation results obtained using the Euclidean distance for the $Y'CBR$ and $R'G'B'$ colour spaces are summarised on Table 4.6.

The comparative evaluation of the use of the RSST framework for colour texture segmentation is presented in Figure 4.42. The use of colour texture labels, obtained with the use of MM, ensures that the segmentation is particularly accurate in the boundary definition, without being over-segmented. The visual inspection of the results obtained with the use of $R'G'B'$ and $Y'CBR$ shows once again that the latter colour space allows perceptually meaningful colour segments to be differentiated successfully. For this reason, in this work, experiments regarding colour segmentation have been performed using $Y'CBR$.

4.4.4 Conclusions on Colour

From the results of the experiments presented in Section 4.4.3, the use of the $Y'CBR$ colour space yields better performance both with the use the conventional RSST framework and with the use of colour texture rules within the RSST framework.
Figure 4.41: Colour texture maps for test sequences Renata, Mobile and Calendar and Garden using (a) $R'G'B'$ and (b) $Y'CBC_R$ colour spaces. Refer to Table 4.7 for interpretation of the colour codes corresponding to colour texture mixtures.
Figure 4.42: Results of colour segmentation using colour texture description for test sequences, *Mobile and Calendar* and *Garden* using (a) \( R'G'B' \) and (b) \( Y'CbCr \) colour spaces. The segmentations contain 4000 segments.
The extension of the texture analysis method proposed in Section 4.4.3 has been demonstrated. The method of generating a classification into a number of texture classes by morphological operators is simple yet very effective. The analysis can be extended to a great number of texture classes generated for each sequence. However, even when restricting the colour texture classes to the minimum for three colour components, the comparative evaluation with the classical RSST segmentation formulation shows a remarkable improvement.

In the continuation of the work presented here, the colour segmentation is performed on the $Y'CB\!C_R$ colour space with gamma correction, as it yields better results.

4.5 Conclusions

In this chapter, the effective extraction and segmentation of image features useful for video object segmentation have been investigated. Moreover, the performance of the RSST framework regarding single-feature segmentation with a variety of features has been explored. In particular, it has been shown how this framework can be adapted for segmentation purposes using motion, texture and colour.

Regarding the use of motion, advantages arising from the inclusion of novel motion features like curl, divergence and acceleration in the RSST framework have been demonstrated. The motion segmentation procedure has been complemented by a consistency check procedure, the aim of which is to ensure temporal consistency of the motion segmentation and rectify erroneous or unreliable segmentation with the use of a memory of the segmentation derived from previous frame segmentations. The outliers resulting from a robust motion estimation procedure have also been harnessed for the purpose of improving motion segmentation accuracy at the motion boundaries.

Regarding the use of texture, a novel texture signature approach based on the application of morphological filters has been formulated. The results of the segmentation using RSST and mathematical morphology have been compared with textural measurements obtained by conventional statistical methods. The important advantage of the proposed approach is the capability of preserving the boundaries and shape of tex-
tured objects. The characterisation of textures using a bank of morphological filters has been used to differentiate and render more flexible the merging criterion of RSST in the presence of texture. Improvements in the segmentation of meaningful regions have been demonstrated as well as an effective treatment for the problem of leakage.

The use of textural signatures has been extended to take into account colour textures and utilise edge information obtained from the three different colour channels for further differentiation and characterisation of the texture. Improved segmentation results using the RSST and the colour textures have been obtained both in $R'G'B'$ and $Y'CBC_R$ colour spaces.
Chapter 5

Feature Integration

5.1 Introduction

In Chapter 4, key features for object-based segmentation have been identified and the RSST framework has been employed to obtain segmentation using single features appropriately.

The next step towards scene segmentation into meaningful objects is the investigation of possible feature combinations in such a way that the segmentation highlights important characteristics of the object.

In video segmentation, motion is the primary source of information. However, since motion boundaries are not accurate, so in order to extract them, it is necessary to resort to spatial information, such as intensity or colour, to extract them. Texture provides an indication of the reliability of the motion information, but at the same time renders the intensity or colour segmentation more difficult since intensity or colour homogeneity properties are violated in textured areas belonging to the same object.

In order to extract accurate boundaries, motion, intensity (and/or colour) and texture have to be integrated in the segmentation. There are broadly two ways of achieving this.

The first way is the so-called hierarchical feature integration, where all the features are taken into consideration simultaneously, using a cost function which contains all
the features of interest. This cost function has to be minimised and each step of the minimisation corresponds to a description of the scene. All the representations form a hierarchy of possible segmentations for the considered scene.

The second way is the so called parallel feature integration, where single features of interest are segmented separately and in a parallel fashion. Only when all the single-feature segmentations are available, can they be fused.

The RSST framework is well adapted to use a complex cost function to perform a hierarchical segmentation. As the number of iterations increases, the description of the scene evolves from very detailed to more generalised. The problem is how to combine features which are very different in nature. Motion provides a global object description, while colour and intensity are local properties and texture is a collective property evaluated over large areas. This is the kind of investigation described in the next section.

5.2 Hierarchical Multiple-feature Segmentation

In order to obtain a hierarchical, multiple-feature segmentation with the RSST framework, each vertex in the graph must be associated with a feature vector (containing attributes related to multiple features). This differentiates the multiple-feature segmentation from the single feature segmentation, where the vertices represent only one feature.

The crucial decision to take is the definition of the cost function, which establishes a homogeneity criterion between regions and therefore the hierarchical representation of the scene itself. The cost function is expressed as the link between vertices. At each recursion of the algorithm, the smallest link is removed and the two most similar regions are merged. However the definition of homogeneity is not trivial to determine for multiple-feature segmentation, since an object consists of inhomogeneous regions, each homogeneous with respect to at least one feature.

The problem of segmentation becomes one of determining how different features relate to and influence each other. It is desirable that this relationship is expressed by a single
5.2. Hierarchical Multiple-feature Segmentation

cost function.

The way the problem has been approached is to attach different scalar weights to inter-frame (optic flow) and intra-frame (colour and/or intensity) features, according to priority, in [14, 9, 88, 89]. The novelty of the proposed approach lies in the fact that while the above methods consist of numerous segmentation algorithms, here, an integrated approach on the sole basis of the RSST framework is attempted.

5.2.1 Link Weight Options

In the experiments presented in Chapter 4, the link weight was often expressed in terms of the absolute value of the difference of features. Each link $L_{k,l}$ between two adjacent vertices $k$ and $l$ is therefore defined as:

$$L_{k,l}(t) = |I_k(t) - I_l(t)|$$

(5.1)

However, other expressions of the cost reflecting the error made by merging two vertices (regions) $k$ and $l$ have been investigated. The findings are reported briefly below.

In Figure 5.1, the different distances used for experiments on the grey-level intensity have been summarised. The variable $x$ is equal to the expression of the error $|I_k(t) - I_l(t)|$. The results of the use of the RSST framework with such cost functions are summarised in Figure 5.2. The performance of the segmentation does not differ very much, as shown in Figure 5.3. Experiments with optic flow and colour triplets have provided similar results.

5.2.2 Multiple-feature Combination

In Chapter 4, single feature segmentation using a variety of features was addressed. In order to perform multiple-feature segmentation, the obvious first step is to investigate the fusion of two features at a time. Potentially, the most interesting combination of two cues is that of intensity and motion, not least because they capture fundamentally different properties of a scene. As previously stated, motion estimates were obtained using the algorithm in [12].
Chapter 5. Feature Integration

Figure 5.1: Distance metrics used as cost function (associated to links in the graph) for preliminary experiments with grey-level intensity.

Figure 5.2: Results obtained for different distance metrics used as the cost function (associated to links in the graph) for experiments with grey-level intensity.
Figure 5.3: Qualitative comparison of performance using distance metrics in (a) $|x|$, in (b) $x^2$, in (c) $|x|^{1/2}$ and in (d) $\exp|x|$. The variable $x$ is a measure of error.
In this case, the features of interest are, therefore, the grey-level intensity at each pixel position and the horizontal and vertical components of the dense optic flow at the same position. The feature vector associated with a vertex (region) \( I \) is expressed as:

\[
\bar{I}(I) = [I_0(I)I_1(I)I_2(I)] = [I_0(I)u^N(I)v^N(I)]
\]  

(5.2)

where \( u^N(I) \) and \( v^N(I) \) are the horizontal and vertical elements of the optical flow for each pixel appropriately normalised so that their dynamic range spans the same interval of values (i.e. 0 — 255). Therefore, each link \( L_{k,i} \) can be defined as [101]:

\[
L_{k,i}(t) = |I_k(t) - I_l(t)| + |u_k^N(t) - u_l^N(t)| + |v_k^N(t) - v_l^N(t)|
\]  

(5.3)

in a linear fashion or:

\[
L_{k,i}(t) = \left( |u_k^N(t) - u_l^N(t)| + |v_k^N(t) - v_l^N(t)| \right) e^{I_k(t) - I_l(t)}
\]  

(5.4)

in a non-linear fashion. The cost function of Equation 5.4 is better suited to reflect the fact that perceived contrast sensitivity of the HVS is proportional to the logarithm of the intensity of the physical luminance signal [8]. In Equation 5.4, the motion component is modulated by the intensity of the spatial signal, as perceived by the human viewer.

Linear and non-linear cost function surfaces are shown in Figure 5.4.

In Figure 5.5, the results of the use of linear (in (a)) and non-linear (in (b)) cost functions are shown. The difference in performance is not relevant. However, the boundaries of the moving objects are preserved better by the linear cost function than by the non-linear combination.

Moreover, when the results of the application of multiple-feature homogeneity criteria were compared with the ones obtained by the use of grey-level intensity alone (using PSNR as the evaluation metric) the introduction of multiple features did not improve the performance of the segmentation, as shown in Figure 5.6. In this figure, the diagram represents the difference in performance between the use of intensity alone and the use of the linear combination of intensity and motion.
5.2. Hierarchical Multiple-feature Segmentation

Figure 5.4: Combination of intensity and motion features in (a) linear and (b) non-linear (b) fashion.

Figure 5.5: Qualitative comparison of performance using (a) linear and and (b) non-linear combination of intensity and motion.
Figure 5.6: The introduction of multiple features does not improve the PSNR of the segmentation. Here, the comparison is between using intensity only and using intensity and motion.

This is possibly due to the fact that PSNR is not an adequate performance metric for object-based segmentation. However, it can also be seen from a qualitative inspection that the segmentation is suffering from inaccurate motion boundaries and leakage.

A final note about normalisation is necessary. In this set of experiments, in the experiments in Chapter 4 and in the following experiments in Chapter 5, normalisation of features with regards to range is used. This is not the only normalisation employed in this work. Another popular normalisation technique, used in [88, 89] to name but a few, is done with regards to variance. It consists of the ratio of the quadratic difference of a certain feature and the corresponding variance calculated for the whole image. Experiments conducted with this normalisation technique have generally yielded very similar results from a qualitative point of view. From a quantitative point of while yielding numerically close, but slightly worse results compared with the range normalisation. An example is given in Figure 5.7, where segmentations employing intensity and optic flow components are compared.
5.2. Hierarchical Multiple-feature Segmentation

5.2.3 Linear Weighting of Intensity and Motion

A crucial issue to be investigated is the one of the relative importance to attach to different features. Limiting the influence of a feature in relation to another can improve the results of the segmentation. This aspect has been investigated with the use of appropriate weights.

In previous publications on the subject the use of weights to modulate the influence of different features has been expressed as a linear homogeneity criterion, e.g. in [14]. Following this example, the investigation in this work has also concentrated on linear criteria for the experiments using different weights.

The incorporation of the use of weights in the RSST framework has been done using the linear criterion, suggested in [14] and expressed as:

\[ L_{k,l} = \alpha D(I_l, I_k) + (1 - \alpha) D(v_l, v_k) \]  

(5.5)

where vector \( v \) is the velocity associated with a generic vertex (or region) and \( D(\cdot) \) indicates a generic distance between image features. In order to make the specification of the distance \( D(\cdot) \) explicit, the norm \( L_1 \) has been used in this implementation.

Therefore, the homogeneity criterion has been expressed as:
Chapter 5. Feature Integration

Figure 5.8: Comparison of performance between the use of cost function 5.6 and of the set of parameters \((0.5, 0.5, 0.5)\) and set of parameters \((0.5, 0.25, 0.25)\). Test sequences: (a) Renata, (b) Mobile and Calendar and (c) Garden.

\[
L_{k,l} = \alpha|I_l - I_k| + \beta|u_l - u_k| + \gamma|v_l - v_k|
\]  

(5.6)

where \(u\) and \(v\) are respectively the horizontal and vertical components of the velocity vector \(\mathbf{v}\) and \(\alpha\), \(\beta\), and \(\gamma\) are constants representing feature weights. All the features have been normalised to the same dynamic range, i.e. \(0 - 255\).

The cost function used in Equation 5.3, can be assumed as having parameters \((1, 1, 1)\). However the problem with this weighting is that the sum of the weights associated with the motion features is double the weight associated with the intensity. This can be rectified by using this set of parameters \((0.5, 0.25, 0.25)\). Results of the experiments using these two sets of parameters are shown in Figure 5.8, to verify how much this imbalance between features influences the final result. Following this idea and the work in [14], where the sum of the weights is taken as 1, the experiments have been performed with the use of the sets of parameters shown in the following corresponding figures.

In Figure 5.8, the performance of the segmentation using the sum of all weights equal to 1 is compared with the case where the influence of the motion features is doubled.

In Figure 5.9, a general comparison between segmentation performances is presented, taking the sum of weights as 1.
5.2. Hierarchical Multiple-feature Segmentation

In Figures 5.10 and 5.11, it is possible to evaluate qualitatively the results of the segmentations obtained using different weights for the same number of regions. No combination of weights provided satisfactory results.

Using Outlier Maps with a Linear Combination of Features

A seen in Figures 5.10 and 5.11, if the influence of the motion feature is high, then the erroneous motion boundaries are projected onto the final segmentation, defying the purpose of the use of multiple-features to improve the accuracy of the motion segmentation. It is reasonable to try to counteract this by using a measure of reliability of the motion estimates used as motion features. The influence of the motion feature must diminish, where the features do not carry reliable information.

The way this has been implemented is with the introduction of a weight \( w \) related to the reliability of motion estimates. The output of the robust motion estimation algorithm also provides the location of the motion outliers, as a binary map [12]. If the motion estimate is an outlier, the value of the weight \( w \) is set to a suitable scalar \( w > 1 \). Otherwise it is set \( w = 1 \). Considering a set of weights \( (\alpha(w), \beta(w), \gamma(w)) \) which are dependent on the reliability of the motion estimate, the modified cost function is expressed as:
Figure 5.10: Qualitative comparison of performance with different weights for intensity and motion features. Settings: (1,1,1) (in (a), (b) and (c)), (0.8,0.1,0.1) (in (d), (e) and (f)), (0.6,0.2,0.2) (in (g), (h) and (i)), (0.4,0.3,0.3) (in (j), (k) and (l)) and (0.2,0.4,0.4) (m, n and o). The number of regions is 1000.
Figure 5.11: Qualitative comparison of performance with different weights for intensity and motion features. Settings: (0.5, 0.25, 0.25) (in (a), (b) and (c)), (0.8, 0.2, 0.2) (in (d), (e) and (f)), (0.6, 0.4, 0.4) (in (g), (h) and (i)), (0.4, 0.6, 0.6) (in (j), (k) and (l)) and (0.2, 0.8, 0.8) (in (m), (n) and (o)). The number of regions is 1000.
\[ L_{k,l} = \alpha(w)|I_l - J_k| + \beta(w)\frac{|u^l - u^k|}{w} + \gamma(w)\frac{|v^l - v^k|}{w} \] (5.7)

The weights associated with the intensity and motion are not fixed but depend on the reliability of the motion estimate. In order to calculate this influence of motion vector reliability, a set of weights \((\alpha(w), \beta(w), \gamma(w))\) is associated with any vertex \(l\) and initialised as follows:

\[
(\alpha(w), \beta(w), \gamma(w)) = \begin{cases} 
(\alpha, \beta, \gamma) & \text{if } l \text{ is not an outlier} \\
(1 - 2\frac{\beta}{w}, \frac{\beta}{w}, \frac{\beta}{w}) & \text{otherwise}
\end{cases}
\] (5.8)

Experiments have been performed using different parameters \((\alpha, \beta, \gamma)\) and varying the parameter of influence of the outlier \(w\). The values of parameters used are presented in Figure 5.12.

From the performance evaluation with PSNR (see Figure 5.12) it appears that the introduction of the outlier weight improves the results for large numbers of regions and where the weight attached to the intensity is larger than the weight attached to the motion. Better results are achieved for higher values of weight \(w\).

The opposite effect is observed for weights attached to intensity lower than the weights attached to motion. It is necessary to take into account that the PSNR evaluates the fidelity of the segmentation to the original image in terms of intensity/colour. However, by visual inspection of Figure 5.13, it can be appreciated that the definition of the boundaries of the moving objects is more accurate and leakage is reduced, for higher values of motion weights.

Overall this kind of measure encourages a good object-based segmentation. On the other hand, the experimental results show that the performance of the segmentation degrades very quickly and it is not possible to obtain meaningful segmentations which contain few regions. In this case, the introduction of the weight in terms of a binary value linked to the positions of the outliers renders the segmentation very noisy.

However, as seen in Chapter 4, location of motion outliers provides some information in the case of textured regions, but not in the case of homogeneous regions. Therefore
Figure 5.12: Comparison of performance for different values of the parameter of influence of outliers $w$ (it varies from 2 to 8) and different settings of parameters $(\alpha, \beta, \gamma)$. The test sequence is Renata.
Figure 5.13: Qualitative comparison of performance for $(\alpha, \beta, \gamma) = (1, 1, 1)$ (in all four examples) and $w = 2$ (in (b) and (d)). In (b), an effective improvement in the accuracy of the motion object boundaries is achieved. The opposite is true for case (d). The segmentations consist of 4000 segments. Images have been equalised for better visibility of region boundaries.
5.2. Hierarchical Multiple-feature Segmentation

the use of such a reliability weight is of limited help in correcting the erroneous motion boundaries.

Using Acceleration with a Linear Combination of Features

Another eventuality is where in areas of low intensity contrast and low textural activity, both grey-level intensity and motion do not provide sufficient information to discriminate boundaries between moving objects. The experiments in Chapter 4 show that acceleration features generally give a good indication of the localisation of the motion boundary. The acceleration features defined there could then be used to avoid leakage, acting as a reinforcement of the motion feature in the determination of the boundaries. This is especially important if another frequent phenomenon of multiple-feature segmentation is observed. It is possible that even with the weighting scheme, motion and intensity features are combined in such a way that the sum of a high motion contrast and a low intensity contrast is lower than that of a medium motion contrast and a medium intensity contrast. If this happens, then leakage and the merging of two different moving objects occur. In order to prevent this, a weight linked to the acceleration can be associated with the intensity difference, to enhance a low or medium contrast in the presence of a motion boundary. Of course, this could damage the accurate localisation of the boundaries, but also prevent a much larger error in the global localisation of the object.

This correction can be expressed as:

\[ L_{k,l} = \alpha |A_i \tilde{I}_i - A_k \tilde{I}_k| + \beta |\tilde{u}_i - u_k| + \gamma |\tilde{v}_i - v_k| \]  

where \( A_i \) for a generic vertex \( l \) is defined as:

\[ A_i = \begin{cases} 1 & \text{if } a_{i,u}^n - \bar{a}_u < T_u \text{ or } a_{i,v}^n - \bar{a}_v < T_v \\ \frac{|a_{i,u}^n - \bar{a}_u| + |a_{i,v}^n - \bar{a}_v|}{2} & \text{otherwise} \end{cases} \]  

where \( a_{i,u}^n \) and \( a_{i,v}^n \) are the acceleration components associated with the vertex \( l \) and normalised to have the same dynamic range, and \( \bar{a}_u \) and \( \bar{a}_v \) are the averages of the respective acceleration components.
Chapter 5. Feature Integration

Figure 5.14: Performance evaluation of weighting when taking acceleration into account, compared with the worst case settings for the parameters \((\alpha, \beta, \gamma)\) in Equation 5.6. Test sequence: Garden.

The experiments have been performed using the same set of parameters \((\alpha, \beta, \gamma)\) used in the case the evaluation of cost function of Equation 5.6.

The evaluation with the PSNR (see Figure 5.14) does not compare positively with the worst possible combination of parameters in Equation 5.6 taken as a reference. Inspection of Figure 5.15 offers the possibility of evaluating the results qualitatively.

The use of this multiplicative weight, in addition to the one attached to the intensity, forces the objects that belong to the background to merge. The resulting segmentation is comparable to one archived with motion features alone. However, there is a buffer zone where the boundary between moving object and background is highlighted and preserved. However, the segmentation is very noisy and the boundaries are located but not segmented correctly.

5.2.4 Using Texture as a Reliability Measure

In Section 5.2.3, the importance of attaching a reliability measure to a motion feature has been discussed. For this purpose, Equation 5.7 is not satisfactory for two reasons.
Figure 5.15: Qualitative comparison of performance for \((\alpha, \beta, \gamma) = (0.8, 0.2, 0.2)\), using outlier information (left column) and using acceleration (right column). In (b), an effective improvement in the accuracy of the motion object boundaries is achieved. The opposite is true for case (d). The number of regions is 500 in (a) and (b) and 16000 in (c) and (d).
Chapter 5. Feature Integration

The first is that it uses a binary map of outlier locations to guide the weighting. A binary system of weights produces abrupt transitions in the segmentation. It is desirable to have a smooth transition between motion boundaries which are characterised by different reliability. The second reason is that the location of the outliers gives an incomplete picture of the reliability of the estimates themselves. Outliers tend to be localised in textured areas. In more homogeneous areas, there is no indication of the reliability of the motion features. Indeed, it is in homogeneous areas that boundary errors are more likely to occur. For this reason, the approaches that aim to estimate the reliability of the motion features, achieved this by linking it to the quantity of textural activity involved, on the basis that motion measures are more reliable on textured areas than on the homogeneous ones. Therefore, colour or intensity should be the primary sources of information in the homogeneous regions.

One of the most interesting criteria for changing the reciprocal influence of intensity and motion on the basis of the textural content of the support, is the one discussed in [88]. Here, the criterion is to merge the regions according to colour similarity if the region of support is homogeneous in colour, or grey-level or merging according to motion if otherwise. The weight associated with motion and grey-level is not static as in Equation 5.5 but has a variable profile calculated on the basis of the variance of the support and modelled as the profiles shown in Figure 5.16 for the purpose of fuzzy clustering.

In this work, this profile has been used as a cost function and incorporated into the

Figure 5.16: Qualitative representation of the kind of profile used by [9].
5.2. Hierarchical Multiple-feature Segmentation

RSST framework. However in the work reported in [88] and the following approaches, the measure used to identify textured regions was statistical, i.e. the variance. This approach has all the drawbacks derived from feature mixing which blurs the boundaries between textures. In the decision about the level of textural content, a novel criterion based on the use of multiple scale morphological masks was developed.

The idea is to use two different masks obtained by the same structuring element at two different sizes $s_1 > s_2$. The corresponding masks are $M_1$ and $M_2$. The areas selected as homogeneous by the mask $M_1$ are the most homogeneous. The areas selected as textured by the mask $M_2$ are the most highly textured. The areas in between the two masks are the ones for which the decision about the textural content is more difficult. For a graphical visualisation, refer to Figure 5.17.

The areas that are selected as textured by the wider mask $M_1$ but not by $M_2$ are the areas of gradual transition in textural content. We define $d_1$ as the Euclidean distance of a point on the image plane from the homogeneous area labelled in $M_1$ and $d_2$ as the Euclidean distance of a point from the textured area labelled in $M_2$, as shown in Figure 5.20. The maps of the distances $d_1$ and $d_2$ are shown in Figure 5.18. The two distances are combined in a distance $D = \frac{d_1}{d_1 + d_2}$. The plot of this is shown in Figure 5.19.

Between the region selected as textured by $M_2$ and as homogeneous by $M_1$ there will be
Figure 5.18: Map of distances $d_1$ (in (a), (b) and (c)) and $d_2$ (in (d), (e) and (f)). The mask $M_1$ is obtained using a disk structuring element of radius $r_1 = 7$ pixels and the mask $M_2$ is obtained using a disk structuring element of radius $r_2 = 3$ pixels. Test sequences: in (a) and (d) Renata, in (b) and (e) Mobile and Calendar, in (c) and (f) Garden.
Figure 5.19: Diagram of the weighting profile $D = \frac{d_1}{d_1 + d_2}$. $D = 1$ where the mask $M_2$ shows the textured areas of support and $D = 0$ where the mask $M_1$ shows the homogeneous areas of support. This function is used as a weight to modulate the influence of motion and intensity features.

Figure 5.20: Definition of distances $d_1$ and $d_2$. Points $d_1 = d_2$. 

$d_2 = 0$

$d_1 = 0$
Chapter 5. Feature Integration

an uncertainty region, which will typically have a configuration like the one in Figure 5.17.

In the region that is maximally textured, the profile $D$ will have value of 1, while, in the region which is maximally homogeneous, the profile $D$ will have value 0. The profile degrades more or less smoothly from 1 to 0 according to the width of the region of uncertainty, while it assumes an intermediate value in the centre of the uncertainty region, as shown in Figure 5.20.

The advantage of defining the profile with the use of morphological filters lies in the fact that the morphological filters preserve the shape of the textured regions, as discussed in Section 4.3.4. Therefore, the use of two filters based on different filter apertures $s_1$ and $s_2$ will discriminate between the density of the texture, but avoid blurring of the boundary between textured areas and homogeneous ones. Transition areas will be areas where the textural content is diminishing. Moreover, the profile $D$ adapts to the width of the transition areas, ensuring that its shape follows closely the distribution of the texture. A short distance between textured and homogeneous areas will be accompanied by a more rapid transition in the profile $D$, while a larger distance will be characterised by a smoother profile.

Let us define $C = 1 - D$ as the complementary profile to $D$. Let us also indicate as $D_i$ and $C_i$ the value of the texture profiles corresponding to the generic vertex $i$. The cost function used for experiments is derived from the linear weighting in Equation 5.6 and it is expressed as:

$$L_{k,l} = \alpha|C_l I_i - C_k I_k| + \beta|D_l u_l - D_k v_k| + \gamma|D_l v_l - D_k u_k|$$

(5.11)

where $\alpha$, $\beta$ and $\gamma$ are constants. Profile $C$ modulates spatial features and $D$ temporal features. As in [88], the influence of the spatial features decreases with an increase in textural content and vice versa for the temporal features. A weighting of the temporal and spatial features is also applied using parameters $\alpha$, $\beta$ and $\gamma$.

A performance evaluation of this weighting approach is shown in Figures 5.21 and 5.22. In Figure 5.21, the performance obtained by the baseline linear weighting function (parameters $\alpha = 1$, $\beta = 1$ and $\gamma = 1$) in Equation 5.6 is compared with the approach
based on textural profiles $D$ and $C$, for various settings of parameters $\alpha$, $\beta$ and $\gamma$. The performance (see Figure 5.21) obtained with the use of textural information is worse than the one obtained using the simple linear combination, which does not take into consideration texture, when the number of regions is large. This is because, for highly textured regions, the vertices are merged using only motion information and this causes loss of detail. For lower numbers of regions however, the performance is much better. When there are fewer regions in the segmentation, the creation of temporally homogeneous segments, using reliable motion information weighted by texture, improves the accuracy of the segmentation. To highlight this fact, the experiment was repeated with a reference method obtained with the use of textural profiles $C$ and $D$ (weighted by the parameters $\alpha = 0.8$, $\beta = 1$ and $\gamma = 1$, which produced the highest PSNR for a lower numbers of regions). This has been compared with a range of parameter settings for Equation 5.6 that have yielded better results in the case of segmentations consisting of a low number of regions. The results of this experiment are shown in Figure 5.22. The use of textural profiles still provides better results than any of the simpler linear combinations. This segmentation procedure produces a better overall segmentation by avoiding over-segmentation and preventing leakage. This is clearly shown in Figure 5.23.

However, even with this cost function, it has been impossible to obtain a truly satisfactory moving object segmentation.

5.2.5 Comments on Hierarchical Multiple-feature Segmentation

In this section, various methods for combining different features into a single cost function have been implemented using the RSST framework. It has been shown how RSST can be flexibly adapted for this task, providing hierarchical segmentation that is perceptually plausible and comparing favourably with other more complex and costly state-of-the art methods, like the ones presented in [14, 9]. Improvements on the cost functions used in [14] have been presented with the consideration of motion outliers and acceleration measures. Improvements on the method in [9] have been proposed with regard to texture characterisation and the texture related weighting strategy, as
Figure 5.21: Comparison between the cost function of Equation 5.6 ($\alpha = 1$, $\beta = 1$ and $\gamma = 1$) and different sets of parameters for the profile of motion $D$ and intensity $C$. Test sequences: in (a) Renata, in (b) Mobile and Calendar and in (c) Garden.
5.2. Hierarchical Multiple-feature Segmentation

Figure 5.22: Comparison between the use of the cost function of Equation 5.11 ($\alpha = 0.8$, $\beta = 1$ and $\gamma = 1$) and that of Equation 5.6. Test sequences: in (a) Renata, in (b) Mobile and Calendar and in (c) Garden.
Figure 5.23: Effect of introducing a textural profile function with \((\alpha, \beta, \gamma) = (1, 1, 1)\). Comparison between cost functions expressed by Equations 5.6 (left column) and 5.11 (right column). The use of texture yields an improvement in the case of (a) and (b), but the results are not always acceptable, as shown in (c) and (d). The number of regions is 350 for (a) and (b) and 500 for (c) and (d).
5.3 Hybrid Hierarchical Architecture for Moving Object Segmentation

well as the incorporation of a corresponding cost function into the iterative structure of RSST.

However, the experiments have shown that the hierarchical approach which incorporates all the features into a unique cost function is not of practical use for object-based segmentation. The use of multiple features does not usually return the moving object for a small number of regions. This is also the reason why methods using this kind of approach must often resort to tracking or additional motion segmentation to identify a moving object. The parameters are difficult to set. There is no set of parameters that can provide both a robust motion segmentation and at the same time accurate motion boundaries, like the ones provided by colour and/or intensity. But improvement on the accuracy of the motion boundaries is the reason why multiple features are employed in the first place. A modification of the merging criteria seems more appropriate, as shown in the experiments. However, it would be very difficult to derive the exact point at which the criteria have to change.

While experiments have only been carried out with linearly weighted combination of features, from the limited experience gained from the non-linear cost functions, it seems that a non-linear cost function would be more difficult to appraise in relation to the contribution of each constituent feature.

Given all these difficulties, it was decided not to try to further incorporate colour, as it would have made the weighting scheme more complicated. On the other hand, most studies have concluded that, colour does not give much of an advantage compared to the intensity alone, from the perceptual point of view.

5.3 Hybrid Hierarchical Architecture for Moving Object Segmentation

In order to obtain an accurate definition of meaningful object boundaries, spatio-temporal segmentation techniques operate in the spatial (inter-frame) and temporal (intra-frame) domain, using intensity/colour and motion information. In Chapter 2, it has been shown how these methods can be classified into two different categories accord-
ing to the strategy employed to fuse spatial and temporal information, namely hierarchical spatio-temporal segmentation methods and parallel spatio-temporal segmentation methods. Hierarchical methods rely on the definition of joint spatio-temporal similarity (or homogeneity) criteria. On the basis of these criteria, they proceed in a region merging fashion. The definition of a joint spatio-temporal similarity function presents numerous difficulties due to the necessity of defining weights which are generally sequence dependent and based on heuristics. They often result in over-segmentation and they are computationally expensive. On the other hand, parallel methods perform single-feature spatial and temporal segmentations separately and successively integrate the information by employing an appropriate set of rules. These methods manage to overcome the over-segmentation problem shown by the hierarchical techniques, achieve good accuracy in the location of object boundaries and are simpler to implement. However, intensity/colour and motion alone do not allow for the complete extraction of meaningful objects. In particular, this class of methods is unable to cope with textured sequences, which represent most of natural scenes.

In Section 5.2, an extension of the RSST framework applied to the problem of performing multiple-feature segmentation has been presented. This is a hierarchical approach and it suffers from the same drawbacks of other state-of-the-art approaches of the same kind. In particular, the use of a local multiple-feature homogeneity criterion for merging regions is disadvantageous since unreliable local estimates of the features can produce global effects. Moving objects are global entities of the scene, while the spatial information is necessarily local. The presence of a weak boundary can lead to the erroneous merging of two or more regions that are globally very different and this problem is not easily solved by weighting schemes, as it has been possible to see in the experiments in Section 5.2 as well as in the current literature. Keeping temporal (global) segmentation and spatial (local) segmentation separated seems to lead to a better localisation of the object.

The hierarchical approach to the processing of multiple features is not supported by psycho-visual evidence. As discussed in Chapter 3, processing of motion information occurs in parallel to the processing of the spatial information. A functional separation of visual tasks is present in the form of distinct visual areas devoted predominantly to
the processing of single features. There is, however, an element of hierarchical organisation inside visual areas and visual pathways: the detected features are progressively organised in a more complex structure, contributing to the perception of more complex objects.

The parallel structure described in the primal visual pathways theory is not a simple one either. With reference to Chapter 3, there is extensive feedback from higher visual areas (like V3 and V4 or V5) to lower visual areas (like V1). This mechanism helps refine the outputs of the lower visual areas [107]. There is also communication between visual areas situated in separated pathways.

The idea of a hierarchical segmentation on the basis of a unique and static cost function is not perceptually motivated and it does not provide a reasonable improvement in the segmentation, unless it is coupled with tracking. However, tracking is inherently a temporal processing step. There is obviously a scale of importance of features: motion is the most important feature in object segmentation. It is then possible to think then of a hierarchy of features that descends from the more global one, motion, to a more localised one, texture, which still requires a wide area of support to be evaluated, to a local one, colour or intensity. Parallelism, separation and feedback are useful concepts provided by the biological model. The reasonable hypothesis on which to ground the segmentation, is that the moving objects are globally correct, and need refinement only locally on the motion boundaries, where motion estimators are more likely to be erroneous. The spatial information contained in each moving object detected by motion alone can be processed in parallel and the grouping of textural and colour features inside these objects can be kept separated (single-feature segmentation). A refinement (multiple-feature segmentation) is possible once more global evidence of regions has been acquired. This is performed once again hierarchically, first by comparing segments obtained by spatial features and corresponding to the same motion segment and then comparing spatial segments only on the boundaries between different motion segments.

In this section, we present a novel architecture for object-based segmentation of moving sequences, based on the concepts explained above [140]. This architecture comprises two stages. In the first stage, objects of interest are located using a top-down approach
that uses single features and exploits global information to avoid over-segmentation. The features used are motion, intensity and texture. In the second stage, a bottom-up (region-merging) rule-based approach is used, employing multiple features and exploiting local information across object boundaries. The proposed architecture does not depend on any particular method for feature extraction or segmentation, while most of the required stages can be performed in parallel.

This architecture can be considered a hybrid hierarchical architecture, where the hierarchy is built according to the order in which the single feature segmentation is performed. The baseline segmentation technique used here is RSST. Each step requiring a different cost function and feature (or combination of features) is indicated as layer, to stress the resemblance to the layered structure of the visual cortex. The results obtained from a previous layer of segmentation are used in the subsequent one and there is extensive feedback from the multiple-feature segmentation to the outputs of the single feature segmentation layers. In this way, a pyramid of different feature segmentation is constructed. At the vertex of the pyramid, in a top down fashion, is the motion segmentation, because this can avoid damaging the moving object boundaries because of leakage or weak intensity/colour boundaries. Once this distinction is made, the scene is further decomposed.

In a second stage of elaboration, a proper multiple-feature stage, the different features are fused together, in two different steps, first with the use of RSST and second with a rule-based approach for the final definition of the object.

In comparison with a purely hierarchical bottom-up strategy, this architecture does not present the difficulties and ambiguities linked to the use of a multiple-feature cost function. In comparison with a purely parallel moving object segmentation, where colour/intensity and motion segmentation are performed in parallel and then fused by the use of rules, this approach is more robust to erroneous segmentation in the colour/intensity domain, while preserving the ability to refine the motion boundaries with the colour/intensity information.
5.3. Hybrid Hierarchical Architecture for Moving Object Segmentation

5.3.1 Proposed Architecture

The proposed architecture, which is shown in Figure 5.24, comprises two stages. In the first stage the image is segmented in a top-down fashion, using a hierarchy of features, namely motion, texture and grey-level intensity. Global motion information is used to locate moving objects approximately, followed by texture analysis and intensity-based segmentation.

In the second stage, the segments are refined using a bottom-up strategy that employs a combination of features and a set of appropriate rules. As a result, local errors which previously occurred in the description of boundaries are rectified. The proposed architecture combines the best features from the top-down approach (avoidance of over-segmentation and use of global information) and the bottom-up approach (accurate location of boundaries).

5.3.2 Building a Feature Hierarchy

The first stage comprises three layers described in detail below.

Motion segmentation layer (M-layer): Image $F_t$ (a frame of the moving sequence at time $t$) is segmented into several non-overlapping and collectively exhaustive motion segments $M_{i,t}$ so that $F_t = \bigcup_{i=1}^{N} M_{i,t}$ and $M_{i,t} \cap M_{i,t'} = \emptyset$, when $i \neq i'$, for $i = 1, \ldots, N$, $N$ being the total number of motion segments. Motion segmentation is obtained by using the RSST, as in Chapter 4.

In this case of motion segmentation, the feature vector initially assigned to a vertex consists of the horizontal and vertical components $[u(p)v(p)]$ of a previously computed dense optic flow field [12].

The cost function between two regions $p_i$ and $p_j$ used in this case is:

$$C_m(p_i, p_j) = |\tilde{V}^k(p_i) - \tilde{V}^k(p_j)|$$

(5.12)

where $|\cdot|$ is the $L_1$ norm. The output of this layer is a picture segmented in a top-down fashion into regions each corresponding to a dominant motion.
Figure 5.24: Diagram of proposed architecture and outputs of layers of processing of a dynamic scene.
5.3. Hybrid Hierarchical Architecture for Moving Object Segmentation

Texture segmentation layer (T-layer): Each motion segment $M_{t,i}$ is further segmented using textural activity criteria. Segments of high textural activity corresponding to motion segment $M_{t,i}$ will be indicated as $T_{t,i}^l$, where $T_{t,i}^l \subseteq M_{t,i}$ for $l = 1, \ldots, L$, while segments of low textural activity will be indicated as $H_{t,i}^m$ where $H_{t,i}^m \subseteq M_{t,i}$ for $m = 1, \ldots, M$ and $\{\bigcup_{l=1}^L T_{t,i}^l\} \cup \{\bigcup_{m=1}^M H_{t,i}^m\} = M_{t,i}$. The textured segments have been obtained from the connected components of the labelled areas obtained by morphological filtering (see Chapter 4) on the edges computed on the grey-level image using Canny edge detector. Each textured segment can be more finely segmented using a cost function such as the one in Equation 5.13. This choice does not affect the effectiveness of the method in its generality. In this layer, each motion segment $M_{t,i}$ can be processed independently, thus facilitating parallel processing.

Intensity segmentation layer (I-layer): Each low-texture segment $H_{t,i}^m$ is subject to grey-level segmentation, using the RSST algorithm. In this case, the feature vector after $k$ iterations is the mean grey-level intensity $\bar{\mu}(p)$ computed over a region $p$. The cost function used is:

$$C_i(p_i, p_j) = |\bar{\mu}(p_i) - \bar{\mu}(p_j)| \quad (5.13)$$

The set of segments produced by this segmentation is indicated as $I_{t,i}^{mj}$, so it holds that $\bigcup_{j=1}^J I_{t,i}^{mj} = H_{t,i}^m$ and $\{\bigcup_{l=1}^L T_{t,i}^l\} \cup \{\bigcup_{m=1}^M \bigcup_{j=1}^J I_{t,i}^{mj}\} = M_{t,i}$.

5.3.3 Boundary Refinement

As has already been stated, the first stage produces rather inaccurate boundaries which need improvement. This is accomplished at this stage, where processing is restricted to the neighbourhood of the boundaries, thereby reducing the amount of information to be processed. The second stage comprises two layers.

Intra-M-frame boundary refinement

Intensity and Texture fusion (IT-layer): Within the boundaries of the same motion segment $M_{t,i}$, segments $I_{t,i}^{mj} \subseteq M_{t,i}$ and $T_{t,i}^l \subseteq M_{t,i}$ are merged using the RSST algo-
Figure 5.25: Schematic description of the notation for the definition of opposite segment.

Inter-M-frame boundary refinement

Grey-level intensity, texture and motion fusion (ITM-layer): We define $B_{ij}$ the common boundary between two motion segments $M_{ti,i}$ and $M_{ti,j}$, as shown in Figure 5.25. We define $(S_{ti,i}^r, S_{ti,j}^q)$ as a pair of segments that touch a common boundary and lie on different motion segments so that $S_{ti,i}^r \subseteq M_{ti,i}$ and $S_{ti,j}^q \subseteq M_{ti,j}$. This is illustrated in Figure 5.25. According to this topology, $M_{ti,j}$ is the opposite motion segment for $S_{ti,i}^r$ and $M_{ti,j}$ is the opposite motion segment for $S_{ti,j}^q$. Each segment $S_{ti,i}^r$ is now specified by four features: average optic flow in the horizontal $U_{ti,i}$ and vertical direction $V_{ti,i}$, average gradient of intensity $G_{ti,i}$ (obtained by a Sobel operator) and average intensity $I_{ti,i}$. We denote the average horizontal and vertical optic flow of each motion segment $M_{ti,i}$ as $U_{ti,i}$ and $V_{ti,i}$ respectively. If the area of a segment $S_{ti,i}^r$ is $A_{ti,i}$, we define $\min(S_{ti,i}^r, S_{ti,i}^q) = \min(A_{ti,i}, A_{ti,i}^q)$. We also define the operation of merging a segment $S_{ti,i}^r$ with its opposite motion segment $M_{ti,j}$ as $\text{merge}(S_{ti,i}^r)$. The refinement of common boundaries between two motion segments is carried out using a set of rules.
The motivation for this rule-based approach is that optic flow is more reliable if the regions have a high textural activity, otherwise it is more reliable to use motion information only. Moreover, small segments are merged with larger segments if the difference in intensity is not very high. The application of the above rules requires thresholds as follows: $T_i$, against which the intensity difference between two segments is compared, $T_s$, against which region size is compared and $T_v$, against which the average gradient value is compared. The above rules can be described by the following pseudo-code:

```plaintext
if ($|I_{t,i}^m - I_{t,j}^m| \leq T_i$)
  • if ($\left( (A_{t,i}^m \leq T_s) \lor (A_{t,j}^m \leq T_s) \right)$)
    - merge(min($S_{t,i}^m$, $S_{t,j}^m$))
  • else
    if ($\left( (C_{t,i}^m \leq T_s) \land (C_{t,j}^m \leq T_s) \right)$)
      * merge(min($S_{t,i}^m$, $S_{t,j}^m$))
    else
      * if ($\left( |U_{t,i}^m - U_{t,i}^s| + |V_{t,i}^m - V_{t,i}^s| < |U_{t,j}^m - U_{t,j}^s| + |V_{t,j}^m - V_{t,j}^s| \right)$)
        • merge($S_{t,i}^m$)
      * if ($\left( |U_{t,j}^m - U_{t,j}^s| + |V_{t,j}^m - V_{t,j}^s| < |U_{t,i}^m - U_{t,i}^s| + |V_{t,i}^m - V_{t,i}^s| \right)$)
        • merge($S_{t,j}^m$)
```

If segment $S_{t,i}^m$ has neighbours belonging to more than one motion segment $M_{t,i}$, it is always possible to reduce this case to the case of two neighbouring objects, setting all the objects other than the one taken into consideration as background. This means considering the static background and other differently moving objects as the same object.
5.3.4 Experiments with Hybrid Hierarchical Architecture

Experimental results using the test sequences *Renata*, *Mobile and Calendar* and *Garden* are presented. Three thresholds need to be specified. $T_e$ is expressed as a percentage of the largest segment in the image. $T_i$ is a measure of the intensity difference or contrast, between different segments. It is chosen in such a way that $T_i \leq \bar{C}_i$, where $\bar{C}_i$ is the average intensity difference according to Equation 5.13 among all adjacent segments composing the ITM-layer. Finally, $T_t$ is a measure of the intensity difference, or contrast, between pixels belonging to the same segment. Therefore, following [39], we define a segment as *non-textured* if $\bar{C}_{S,t} \leq T_t < T_i \leq \bar{C}_i$, i.e. if the average contrast between elements of the same segment $S$ is smaller than the average contrast between different segments. The parameters used to produce the results in this section are summarised in Table 5.1. $T_e$ is insensitive to the sequence content, while $T_i$ depends on the intensity distribution of the scene and $T_t$ is linked to $T_i$.

In Figure 5.24, outputs of each layer of the proposed architecture are given with reference to test sequence *Renata*, indicating the boundaries of constituting segments in red. In the first row of Figure 5.24, the output of the M-layer is shown, with region boundaries superimposed on the original image. The second, third, fourth and fifth rows show the outputs of the T-layer, I-layer, IT-layer and ITM-layer respectively. From the comparison between the outputs of the first and the fifth rows, it is possible to see the improvement obtained in the definition of the boundaries with respect to the motion segmentation, especially in regions with low texture (like the wall or the lower part of the calendar) where the motion information is not very reliable. The algorithm avoids over-segmentation of the image. This is particularly evident in the segmentation of highly textured objects (like the upper part of the calendar and the tapestry). The progressive refinement achieved at different layers leads to a hierarchical description of the input scene, which can be very useful for progressive coding applications. The outputs obtained from different layers of the architecture with the use of test sequences *Mobile and Calendar* and *Garden* are presented in Figure 5.26.

The performance of the segmentation achieved for a longer sequence is presented with reference to the test sequences *Mobile and Calendar* and *Garden* in Figure 5.27. Object
boundaries are located consistently from frame to frame, without explicit need for segment tracking.

Overall, the results demonstrate that the proposed algorithm yields intuitively correct segmentations corresponding to actual scene objects whose boundaries have been extracted with a substantial degree of accuracy [141].

5.3.5 Conclusions on the Hybrid Hierarchical Architecture

In summary, the proposed architecture comprises two stages of processing. The first produces a hierarchy of segments that are extracted using single features, namely motion, texture and intensity. The second stage fuses multiple features to reinforce effectively the spatio-temporal coherence of the segments obtained in the first stage. We have adopted an effective morphological texture segmentation technique which allows for the accurate definition of region boundaries and requires low-complexity.

The proposed technique represents a significant improvement in the field of object-oriented segmentation as it avoids over-segmentation caused by bottom-up approaches, whilst improving the accuracy of the definition of object boundaries [141, 140, 142]. A further useful feature is the capability of this architecture to achieve a hierarchical decomposition of a scene into perceptually significant objects. Not only does it locate moving objects accurately, but it also provides a separation between textured and untextured regions. This hierarchical representation of the moving scene can be exploited for object-based analysis and coding of video.

The technique is attractive from an implementation point of view, in the sense that it avoids computationally expensive homogeneity criteria and weighting functions that
Figure 5.26: Outputs of each layer of the hybrid hierarchical architecture, using test sequences Garden and Mobile and Calendar. The segments are indicated by the corresponding highlighted boundaries.
Figure 5.27: Results obtained with the hybrid hierarchical architecture are consistent through time. Here are examples of Garden and Mobile and Calendar at frames 10, 20, 30, 40 and 49.
competing methods require, while most processing stages can be carried out in parallel. The architecture is generic, in the sense that no particular restrictions are imposed in the choice of segmentation tools.

5.4 Incorporating Further Aspects of Psycho-visual Perception

In Section 5.3, it has been seen how paradigms derived from psycho-visual perception of the HVS, such as parallelism, functional separation and feedback, can be successfully implemented into a hybrid hierarchical architecture for performing multiple-feature segmentation.

Experimental results show the effectiveness of the approach and the accuracy of the segmentation obtained. In particular, the parallel processing of the spatial information from different motion segments has proven particularly advantageous in avoiding gross segmentation errors. In fact, this reduces the influence of noisy or ambiguous estimates by delaying the decision about the merging of two regions. This responsibility is delegated to the final rule-based layer of the processing, where the properties of the segments which are candidates for merging have acquired more global consensus from the previous spatial layers. This separation of spatial and temporal processing must be kept in the model. However, the hybrid method presented above suffers from a number of drawbacks.

The flow of processing does not correspond to the one performed by human visual perception, where the processing of colour and texture are kept separated by the processing of motion and only at a later stage is the information fused together. The number of stages involved in the overall process also increases notably the complexity of the segmentation.

A number of perceptually important features have been omitted. Colour information has not been taken into account and the same applies to colour texture and edge information. The latter is a particularly critical omission due to the fact that, while pure region-based segmentation techniques do not rely heavily on edges, the perception
5.4. Incorporating Further Aspects of Psycho-visual Perception

of edges is of fundamental importance in the HVS and is the basis of all other visual processing in V1.

From the computational point of view and with regard to multimedia applications, it requires a number of user-defined parameters, such as the number of parameters relating to the minimum size of the regions, the indicator of homogeneity of the region and texture activity of the regions. These parameters are also sequence-dependent. Ideally, for automatic generation of a semantic description of a sequence, one would like to have a tool that can generate automatically the parameters required with the possibility of intervention as necessary.

The notion and localisation of opposite segments presents difficulties. In general, one segment will have more than one opposite segment. In this case, the choice of opposite segment must be done on the basis of similarity in intensity but this could lead to erroneous identification of the segments, for weak intensity boundaries. This is the same problem previously referred to as leakage. It is also possible to take into consideration the length of the common boundary, but this would introduce another user-defined parameter and the need to balance arbitrarily the influence of intensity similarity with boundary length.

Experiments performed using luminance and texture definition based on the luminance gradient showed that, as far as object-based segmentation is concerned, the fourth layer of segmentation does not improve the definition of the object boundaries. The same segmentation can be achieved by classifying the regions into homogeneous and textured on the basis of the morphological signature, and then applying the rules of the rule-based processor directly to textured regions.

It is necessary to address these drawbacks. Since the parallelism is perceptually motivated, more robust to noise and computationally advantageous, it is reasonable to exploit it, furthering the similarities of the architecture to the HVS biological model. We want to keep the rule-based approach which allows one to fuse the information when all the evidence has been gathered and it is geared to being extended when needed. We want to keep the functional subdivision of tasks, lowering the number of layers necessary to obtain a segmentation.
The following is a proposed model that resembles more closely the functional structure of the visual cortex and integrates the positive elements of the hybrid hierarchical architecture in a simpler way with regard to the necessary processing layers, while using a more complete set of features.

5.4.1 Perceptual Parallel Paradigm

Details of the parallel architecture for moving object segmentation are presented in Figure 5.28. The most important characteristic of the architecture is the parallel processing of temporal and spatial information to mimic the two visual pathways in the visual cortex. Therefore, motion segmentation is kept separate from the colour and texture segmentation and performed in parallel.

Temporal processing is enriched by new stages of processing presented in Sections 4.2.4, 4.2.7, 4.2.8. In particular, motion segmentation using multiple motion features is used, while outliers and consistency checks refine the segmentation. Labelling of motion segments follows. These segments will be refined at the rule-processor stage with spatial information. The output of the motion pathway is a set of motion segments $M_{t,i} = M_i$ so that $F_t = \bigcup_{i=1}^{N} M_{t,i}$ and $M_{t,i} \cap M_{t,j} = \emptyset$, when $i \neq j$, for $i = 1, \ldots, N$, $N$ being the total number of motion segments.

Colour and texture segmentation is performed in parallel with motion segmentation. The discrimination between homogeneous and textured areas is done with the morphological segmentation technique explained in Section 4.4.3. This step allows one to discriminate textured and homogeneous areas without the use of the parameter $T_t$, and avoiding texture segmentation as presented in the T-layer.

Edges are calculated separately on three colour components using the Canny edge detector ($\sigma = 1$, filter size $s = 7$). Morphological closing is performed on each of the three colour edge components, obtained using the $Y'G_BG_R$ colour space. The result of the morphological operation is a binary mask, indicating the presence of texture in a given colour space. The possible combinations of labels are $2^3$. Homogeneous areas of support will be defined as those that do not contain texture in any colour plane, indicated globally as $H$. In total the textured regions are defined as connected
Temporal processing

$F_{t+1}$ $F_{t-1}$

- Motion segmentation
- Moving object labeling
- Outlier refinement
- Consistency check

Spatial processing

$F_t$

- Colour edges
- Colour texture blobs
- Texture segments
- Homogeneous support area
- Edge continuity
- Homogeneous segments

Rule processor

Video Objects

Figure 5.28: Flowchart of parallel architecture for object-based segmentation which mimics the human perceptual visual pathways.
components whose individual labels can be any one of the remaining \((2^3 - 1)\) possible labels. Therefore, the only parameter needed to obtain the colour texture segmentation is the size of the structural element used to define the textured regions. The number of textured regions is determined automatically by the connected components obtained by morphological operations. In this work, such textured connected components are also referred to as colour blobs to emphasise the similarity with the input received from V2 to V3. From the texture segmentation part, it is possible to obtain a set of textured segments \(T_{t,i} = T_i\) so that \(F_t = \bigcup_{i=1}^{L} T_{t,i} + H\) and \(T = \bigcup T_{t,i}\), so that \(T + H = F_t\) and \(T \cap H = \emptyset\), for \(i = 1, \ldots, L\), where \(L\) is the total number of textured segments.

From this step, which mimics the functionality of the V3, it is possible to obtain the areas of support of the homogeneous regions. The homogeneous segmentation is done with the use of a fourth source of information, namely the edges. It is useful to remember that in the hybrid approach, catastrophic errors in the luminance or colour segmentation were avoided by the first layer of the segmentation. There, motion segmentation was performed and kept in place as a separation that would avoid erroneous merging from the motion point of view. The separation was kept until all the information available would have been gathered. This was the main advantage arising from the use of motion segmentation at the top of the processing hierarchy. On the other hand, the number of homogeneous regions would have to be determined by the user and the use of a user-defined parameter is necessary to specify the level of homogeneity for the rule-based approach. In the current parallel paradigm, the information provided by edges is used.

Edges are identified in each of the three colour planes. Then, a dilating operation is performed with the same structural element used for the textural segmentation. The morphological processing helps the definition of a continuous boundary, as suggested in [143] and by the Gestalt law of good continuity [107]. Edges are defined by the union of all areas belonging to the three colour components that have been dilated. Homogeneous segments \(H_{t,m} = H_m\) are defined as connected components which define close areas and are separated from each other by closed edges (boundaries). Therefore \(H = \bigcup H_{t,m}\), so that \(T + H = F_t\) and \(T \cap H = \emptyset\), for \(i = 1, \ldots, M\), \(M\) being the total number of homogeneous segments. This represents the termination criterion so there is no need for the user to define a final number of regions. There is also no need for
5.4. Incorporating Further Aspects of Psycho-visual Perception

the user to specify the parameter $T_i$, that indicates the level of homogeneity.

Once the colour texture and colour homogeneous segmentations are obtained, the rules explained in Section 5.3 with reference to the hybrid method are applied to textured or homogeneous regions whose area of support belongs to more than one motion segment. This renders the segments available for immediate rule processing. All motion segments can compete simultaneously for the assignment of a textured or homogeneous region.

The fusion of the information is performed by a rule-based processor. The rules are applied only to segments which are either textured or homogeneous, as explained in Figure 5.29 and 5.30. Following the findings of the experiments reported in Section 5.3.4, parts of a textured segment that belong to different objects are analysed separately, i.e. a segment, pixels of which belong to different motion segments, is separated into a number of sub-segments. Each sub-segment contains pixels belonging to only one motion segment. Every textured sub-segment is assigned an average horizontal and vertical flow, and these are compared with all the average horizontal and vertical flows of the motion segment, as seen in the rule-based approach explained in Section 5.3.3. For the computation of the average flows of a sub-segment of textured segment, the outlier pixels are excluded. If a textured sub-segment is made up of more than $T\%$ of outlier pixels, it is not taken into consideration and it will be assigned to the motion segment later. Here, the majority of the sub-segments belonging to the same textured segment have been assigned. Homogeneous segments, which overlap a motion boundary, are entirely assigned to the moving object with which they have maximum overlap.

It is also useful to notice that the user can always influence the segmentation and tune it for specific purposes or applications. A number of textured and homogeneous segments arises automatically as connected components of the morphological filtering. Each of these segments (connected components) can be further divided, obtaining a finer description of the spatial information. In fact, the RSST can still be used in a parallel fashion on textured and homogeneous areas of support, which are processed as separate forests of vertices. The pixels or vertices corresponding to edges are subtracted from the initial forest in the case of colour segmentation to force the merging of those connected
components first. The subtraction of edge pixels from the RSST graphs assures that continuous boundaries of objects will not suffer from leakage. The segmentation can proceed in each separate connected component of textured or homogeneous regions with the preferred criteria, e.g. colour similarity. In this way, it is possible to segment into more regions than the ones set automatically for the object based segmentation. For fewer regions, it is always possible to continue segmenting with the RSST, where each connected component identified in the previous automatic procedure will be merged according to a similarity criterion, e.g. using colour.

5.4.2 Experimental Results and Comparison

Experimental results for the latter refined and simplified version of the segmentation algorithm are presented using the standard test sequences Renata, Mobile and Calendar and Garden. A comparison with corresponding results obtained using the previous version of the algorithm (Section 5.3) is also presented in Figure 5.31.

Motion segmentation is obtained for results obtained from the hierarchical architecture using the RSST with only translational motion features. For results obtained with the parallel architecture, the motion segmentation has been obtained with multiple motion...
5.4. Incorporating Further Aspects of Psycho-visual Perception

Figure 5.30: Simplified schematic of textured segment overlapping multiple motion object boundaries. Average motion features for each part of the segment belonging to different moving objects are calculated without taking outliers into consideration.

features, refined by the use of motion outliers and consistency checks. This results in a more complete labelling of the moving objects as is clear in the example of Figure 5.32 (c). In this case, labels obtained for the objects in the foreground correspond to a number of objects: the tree trunk, the tree branches and two layers of the flowerbed which have different depths. In the case of the motion segmentation using only translational features the object labelled is only the tree trunk.

For the three colour planes, the edges were calculated using the Canny edge detector ($\sigma = 1$ and $\alpha = 7$) before a closing operation was performed. The choice of structural element was motivated from the work of in [108] where it is shown how the luminance texture is perceptually more relevant than the red-green and blue-yellow textures. The structural element is a $3 \times 3$ square mask. In the case of luminance $Y'$ the textured areas are further dilated in order to increase the emphasis on this texture. Again according to [108] the textures in the blue-yellow plane are less relevant than those in the red-green plane. Therefore the textured areas obtained from the blue-yellow-ish signatures $C_B$ are eroded, always using the same structural element. This specifies the texture signature and the number of textured regions is determined automatically as the result of the definition of the connected components that share the same textural signatures.
The segmentation into homogeneous segments is obtained by taking the connected components, whose area of support is not textured, into consideration. The colour texture and colour homogeneous segmentation provide a set of segments and a classification into textured and homogeneous. The segments, which overlap on a motion boundary, are then processed according to the rules set in Section 5.4.1. These rules require that textured segments be split according to motion (as shown in Figure 5.30) and assigned to the motion object most similar to them using the horizontal and vertical flow for similarity purposes. With reference to Figure 5.30, this rule means that sub-segments $T_{1,i}$, $T_{2,i}$, and $T_{3,i}$ obtained from textured segment $T_j$ are processed separately for the evaluation of their motion similarity against neighbouring moving objects $M_1$, $M_2$, and $M_3$. The assignment of homogeneous regions is performed on the basis of the largest area belonging to a single motion segment, as shown in Figure 5.29. In this figure, the area $A_{1,m} > A_{2,m} > A_{3,m}$, therefore the whole segment $H_m$ is assigned to the moving object $M_1$.

Results of the segmentation obtained with this version and the previous version (in Section 5.3) are presented in Figure 5.31. For an objective comparison, the segmentation obtained in the I-layer was performed using colour triplets instead of intensity only, so it was the segmentation obtained in the IT-layer. Textured areas were determined using binary masks obtained by morphological segmentation, instead of using the parameter $T_t$ [142].

In test sequence Renata (see Figure 5.31), it is possible to see the improved accuracy obtained in both a homogeneous area which has a smooth gradient in intensity (the shirt) and in a moving textured segment (the scarf) which has a very similar average colour to the background. The improvement in the segmentation of the textured segments is a common characteristic throughout this whole sequence. It can be attributed to the inclusion of multiple-motion features, the use of motion outliers in the presence of a textured area of support and the better discrimination of textured segments due to the use of colour edges. In the case of the homogeneous segment, the presence of a smooth gradient can lead to ambiguous segmentation with region-merging techniques and the result depends on the user-defined number of regions where the segmentation has been stopped. In the case of the parallel architecture, the number of regions is defined by
the strongest edge identified by the edge detector and consolidated by morphological filtering.

In test sequence *Mobile and Calendar* (see Figure 5.31), the key aspects of successful segmentation are colour texture discrimination and continuity of edges. In the case of segmentation using the hierarchical architecture, the segmentation has some leakage in the region between the train tracks and the surface. The edges are weak in places and the texture of the surface is quite smooth. The hierarchical segmentation using colour is prone to failure and the accuracy is obtained only by over-segmentation because the contrast in that region is quite small. On the contrary, the continuity of the edge of the tracks is ensured by using the parallel architecture, regardless of the number of regions chosen. If the reliability of the motion were given only by the density of edges, such as in the case of the hierarchical architecture, then the motion segmentation would hardly be improved, because most of the support for the frame is textured. With the use of colour textures, however, there is better discrimination and the principle of motion similarity can be applied to segments of identically coloured texture. The advantages are clear in Figure 5.31.

In the case of test sequence *Garden* (see Figure 5.31), a better segmentation of similarly textured areas was achieved. Two contributing factors are the better definition of the edges that separate the homogeneous regions and the reliability of the texture characterisation. Using the Canny edge detector followed by dilation, the continuity of edges is strengthened, even in the cases of lower contrast, because the dilation with a $3 \times 3$ mask allows holes in the boundaries that cause leakage in the case of the segmentation with the RSST to be bridged. While a similar criterion was proposed in [143], similar performance is achieved in the current work using a much simpler method. Texture characterisation on the basis of separate colour edges contributes to the detection of differences that would have been missed with the method used in Section 5.3.4, which was based on luminance contrast alone, even under the influence of user intervention.
Hierarchical hybrid architecture  Parallel architecture

Figure 5.31: Comparison of the foreground-background object-based segmentation obtained with the use of the hybrid hierarchical architecture and the parallel architecture. For multiple moving object segmentation, the interested reader is referred to Figures 6.7, 6.8 and 6.9.
Figure 5.32: Labels corresponding to the foreground objects shown in Figure 5.31. Boundaries between different labelled regions are superimposed to facilitate visual inspection. For the corresponding multiple moving object segmentation, the interested reader is referred to Figures 6.7, 6.8 and 6.9.
5.5 Conclusions on Multiple-feature Segmentation

In this chapter, two different approaches to feature combination for the purpose of object-based segmentation have been investigated.

In the hierarchical approach, achievements have been the extension of the use of the RSST technique for a combination of different features simultaneously, while on previous works, the emphasis has been placed on single-feature segmentation. The hierarchical segmentation technique offers flexibility in the choice of segmentation criteria, but does not correspond well with actual objects.

In the hybrid hierarchical approach, a novel single-feature architecture has been proposed. This architecture shows increased accuracy in the determination of object boundaries and robustness to single feature segmentation errors. This can be attributed to the feed-back path that progressively refines the segmentation.

This model still requires the determination of numerous parameters. A less parameter-dependent scheme employing parallel processing has been developed. In this scheme, motion segmentation is performed in parallel with colour and texture segmentation. It requires the use of a single parameter such as the size of a morphological mask. The number of regions is automatically chosen and regions are automatically divided into textured and homogeneous and fed to the rule-based processor.

This method requires no manual intervention and produces meaningful and consistent segmentations. Moreover, it can produce any number of regions. The segmentation can be performed with RSST onto forests of vertices based on the classification into textured and homogeneous areas obtained by morphological filtering and incorporating spatio-temporal merging criteria, such as the ones seen for hierarchical segmentation.
Chapter 6

Performance Evaluation

6.1 Introduction

This chapter addresses an open issue in image and video processing: the performance evaluation of segmentation algorithms.

If still image segmentation is a well-researched field, which has found no satisfactory general purpose solution yet, the same is true for the problem of objective performance evaluation of still image segmentation. Despite numerous attempts both in the case of grey-level [144] and colour [145] image segmentation, a unique procedure for performance evaluation has not emerged. Researchers have turned to the analysis of the Human Visual System in order to find a quantitative metric able to represent the subjective perception of image distortions [146], especially in the field of image compression [147]. But this research is not in an advanced stage, and so far no satisfactory correlation exists between Human Visual System perception and such objective performance metrics.

On the other hand, it has been shown [148] that none of the more complicated distance metrics show a clear advantage in comparison to the Mean Squared Error (MSE) (and therefore the Peak Signal-to-Noise Ratio (PSNR), which is related to it). For this reason such evaluation metrics have been widely used in this work for the purpose of evaluating single-feature segmentation and multiple-feature segmentation. However
the straightforward application of these metrics to the evaluation of multiple-feature segmentation is problematic, as previously commented in Section 5.2. Such metrics are in fact well-apt to describe similarity or dissimilarity between homogeneous quantities, while multiple-feature segmentation is aimed at combining inhomogeneous features. This is one characteristic which renders the performance evaluation of video object segmentation even more difficult than the one of still image segmentation.

Video objects are inhomogeneous entities as they are the result of a combination of multiple features and/or the final output in a multi-stage process that handles single features separately. The main criterion to judge the effectiveness of the object-segmentation depends on the meaningfulness of the object by the human user point of view. This is still out of grasp.

Common methods for performance evaluation of object-based segmentation rely on the use of ground truth moving objects of interest, segmented manually by a human operator [149]. This procedure is cumbersome and time consuming for the human operator, therefore only a limited amount of frames are usually available for comparison and relatively few sequences are manually segmented and available. Moreover, this process can yield unpredictable mistakes due to the tiredness of the human operator, his/her experience and attention granted to the task.

It is clear that an objective and automatic (which does not depend on human interaction) spatio-temporal segmentation measure would be a very useful basis for comparison of algorithmic performance. It could moreover lead to a useful assessment for practical applications, such as video coding or editing. Literature is not vast with regard to the problem of object-based segmentation evaluation. However, an interesting approach is the one of proceeding with the definition of the characteristics possessed by a meaningful object [11] and tuning the metrics to reflect such characteristics.

In this work, object-based segmentation performance is evaluated by using both ground truth and without using ground truth sequences. A procedure to improve the sensitivity of the objective (without the use of ground truth) performance metrics based on the work in [11] is also presented. The results of the segmentation approach presented in Section 5.4 are compared with three state-of-the-art spatio-temporal segmentation
Choice of Benchmark Algorithms

To date there is no universally accepted benchmark algorithm, therefore in this work a number of different approaches have been implemented for comparison. It is interesting in the first instance to compare the segmentation obtained using multiple-features with a segmentation based on motion alone.

The first method considered is a top-down approach to motion-only segmentation [12]. A dense optic flow field is evaluated using a robust multi-resolution motion estimation method. Then the global motion components are extracted in order to obtain a change detection mask by manual thresholding, which is applied separately to each frame of interest. This method of motion estimation seems effective towards locating semantically meaningful objects and the results are consistent during a long sequence. However, the inaccuracy in the determination of real boundaries, makes it less suitable for content-driven analysis.

Then, spatio-temporal methods have been implemented for comparison purposes. Multiple-feature methods are not frequently encountered in the literature [2] and usually depend heavily on a number of user-tuned parameters or require user interaction [88]. Therefore, methods based on colour and motion features alone, that depend on a relative low number of parameters, have been considered here.

The first of these methods is a hierarchical spatio-temporal technique where temporal and spatial information is integrated into a single similarity function [14]. Motion and intensity similarity contribute to this function using a weighting scheme presented already by Equation 5.5. Iterative region merging is achieved by evaluating the joint similarity of neighbouring regions using the watershed segmentation technique. This method suffers from a degree of spatial over-segmentation. Moreover, due to the sensitivity of the watershed transform to noise, the tracked boundaries are often not consistent throughout a long sequence.
The second of these methods is a parallel spatio-temporal segmentation technique which uses single-feature temporal and spatial segmentation in order to obtain spatial and temporal masks and then combines the results obtained using a set of appropriate rules [13].

The motion segmentation part of all the above methods was based on the method in [12], as in the motion-only segmentation technique described at the beginning of this Section, in order to make the comparison between different techniques fair.

6.3 Evaluation using Ground Truth

Performance evaluation metrics using ground truth have been proposed in [149]. They define the ground truth for an image sequence as the support of the object(s) in the subsequent frames of the sequence. When this ground truth is available, they propose different distance measures between the ground truth and the segmented video object planes. They assume that the frames contain \( N \) pixels and \( M \) objects.

In this work three evaluation metrics which use ground truth are used: the misclassification penalty, the shape penalty and the motion penalty. The misclassification penalty quantifies the number of pixels that have been misclassified, weighting them by their distance from the boundary of the ground truth. The shape penalty quantifies the difference in shape between the ground truth and the segmented object, taking into account the difference in a descriptor of curvature at the boundaries. The motion penalty quantifies the difference in the value of the motion (parameters) between the ground truth and the segmented object.

Let us review in detail the performance metrics proposed by [149].

6.3.1 The Misclassification Penalty (DP)

Misclassified pixels between ground truth objects and segmented objects can be penalised according to their distance from the ground truth boundary.

Let the object collections from the ground truth and segmented frames be denoted as \( G = \{g_i, i = 1, \ldots, M\} \) and \( S = \{s_i, i = 1, \ldots, M\} \) respectively, where \( g_i \) is a generic
6.3. Evaluation using Ground Truth

object corresponding to the ground truth and \( s_i \) is a generic object of the segmented frame.

Let the label functions \( L_G^t(k,l) \in \{1, \ldots, M\} \) and \( L_S^t(k,l) \in \{1, \ldots, M\} \) denote the objects to which the pixel at location \((k,l)\) belongs at frame \( t \), in the ground truth and in the segmented frames, respectively. They are related via the indicator function \( I^t(k,l) \) at pixel location \((k,l)\) at frame \( t \) as:

\[
I^t(k,l) = \begin{cases} 
1 & \text{if } L_G^t(k,l) \neq L_S^t(k,l) \\
0 & \text{if } L_G^t(k,l) = L_S^t(k,l)
\end{cases}
\] (6.1)

The discrepancy between two objects \( g_i \) and \( s_i \) due to misclassified pixels can be calculated as shown by Equation 6.2.

\[
0 \leq D_i^t = \frac{\sum_{(k,l) \in \{s_i \cup g_i\}} I^t(k,l) w_{g_i}(k,l)}{\sum_{(k,l) \in \{g_i \cup s_i\}} w_{g_i}(k,l)} < 1
\] (6.2)

Where \( w_{g_i}(k,l) \) is the Euclidean distance of the pixel \((k,l)\) from the boundary of object \( g_i \) [149].

The segmentation error \( D^t \), averaged over all objects, is given by Equation 6.3:

\[
D^t = \frac{1}{M} \sum_{i=1}^{M} D_i^t
\] (6.3)

The spatio-temporal distortion reflecting the misclassification penalty (DP) over \( T \) frames can be calculated as shown in Equation 6.4:

\[
DP = \frac{1}{T} \sum_{i=1}^{T} f(D_i^t)
\] (6.4)

Where \( f(D_i^t) \) is some error function, e.g. the square error, as expressed by Equation 6.5.

\[
f(D_i^t) = (D_i^t)^2
\] (6.5)
Chapter 6. Performance Evaluation

6.3.2 Shape Penalty (DS)

A discriminative shape metric is the turning angle function (TAF) of the object boundaries [10]. The distance between the $M$ object shapes at time $t$ can be calculated using Equation 6.6:

$$0 \leq DS^t = \frac{1}{M} \sum_{i=1}^{M} \left( \frac{\sum_{s=1}^{P} |\Theta^i_{g}(s) - \Theta^i_{s}(s)|}{P \times 2\pi} \right) \leq 1 \quad (6.6)$$

Where $\Theta^i_{g}(\cdot)$ and $\Theta^i_{s}(\cdot)$ denote the turning angle function (TAF) of the ground truth object $i$ at time $t$ and the corresponding segmented video object plane (VOP) respectively. $P$ denotes the number of contour elements used for the calculation of the TAF. The TAF is defined for a given polygon, as shown in Figure 6.1. Let us consider a reference point $O$ on the polygon boundary, and a reference axis through $O$. Let us indicate by $s$ the generic position of a point on the polygon boundary in reference to $O$. The TAF $\nu$ of the point is a function of its position, i.e. $\nu \equiv \nu(s)$ the (counter clockwise) angle between the tangent to the boundary in $s$ and the reference axis.

The definition of TAF can be extended for an object of arbitrary shape as suggested in [149]. Let us suppose that the shapes of the objects examined are convex and the boundary pixels of the objects are assigned a signature as a pair of coordinates in the $(r, \theta)$ domain. The signatures of the shapes to be compared are re-sampled so that equal numbers of samples are obtained at uniformly spaced and identical $\theta$ values, as shown in Figure 6.2, for $\theta$ sampled every 45 degrees. The pointing vectors which point from one boundary pixel to the next pixel are calculated. These vectors approximate...
6.3. Evaluation using Ground Truth

Figure 6.2: Definition of boundary coordinates \((r, \theta)\) and regular sub-sampling of ground truth \(g_i\) object boundary and segmented \(s_i\) object boundary.

Figure 6.3: Definition of rotation angle between two successive pointing vectors on the newly sub-sampled object boundary.
the tangents to the object boundary, in the sampling points considered. The rotation angle between each successive pointing vector is added up to find the TAF of the object shape, as shown in Figure 6.3.

If the shapes are not convex, the re-sampling of the boundary is done by fitting a continuous curve to the boundary pixels and taking samples at equal intervals on this curve.

The discrepancy between two objects can be weighted by the object Area, before averaging. The shape distortion of a sequence can be computed as a weighted average over interval $T$ (where $T$ is the length of the sequence examined), as shown in Equation 6.7:

$$DS = \frac{1}{T} \sum_{i=1}^{T} f(DS_i)$$

(6.7)

### 6.3.3 Motion Penalty (DM)

Let $M_{gt}^t$ and $M_{si}^t$ denote the motion information at time $t$ for the ground truth and the segmented object. $M$ can be parametric or it may represent the motion field at each pixel or neighbourhood of pixels. For video objects, it is useful to compare the trajectories of the segmented objects and the ground truth objects. Therefore it is necessary to define some distance $D$ between motion planes represented by $M_{gt}^t$ and $M_{si}^t$:

$$D_i = D(M_{gt}^t, M_{si}^t)$$

(6.8)

The following $D_i$ distance measure is proposed in [149], as shown by Equation 6.9:

$$0 \leq D_i = \frac{1}{T} \sum_{i=1}^{T} \frac{\|M_{gt}^t - M_{si}^t\|}{\|M_{gt}^t\| + \|M_{si}^t\|} \leq 1$$

(6.9)

The discrepancy $D_i$ is averaged over all objects, giving the final motion penalty $DM$, as expressed by Equation 6.10:

$$DM = \frac{1}{M} \sum_{i=1}^{M} f(D_i)$$

(6.10)
Finally, it is possible to define a combined penalty that sums the different measures of penalty with some appropriate weight.

### 6.4 Evaluation without Ground Truth

The performance evaluation metrics described in Section 6.3 are useful when ground truth maps are available. However, ground truth maps are of limited availability for arbitrary scenes, therefore some objective criteria to evaluate the performance of the object-based segmentation are of interest. In [11] intra-frame and inter-frame colour and motion metrics are presented with the objective of evaluation of segmented video planes without the use of ground truth.

The idea of these metrics is that, not having any ground truth information, boundaries that present high contrast in the spatial and temporal domain are more likely to be the correct ones. Therefore, taking into account one frame, the colour and motion contrast is calculated along the boundaries of a segmented object. A high value of contrast in colour or in motion (parameters) indicates a good segmentation. This idea is exploited in the intra-frame colour metric and in the motion metric used in this work. Additionally, a measure of consistency of the segmentation is also defined by the inter-frame colour metrics. Let us consider a global descriptor of a segmented object at time $t$, such as the colour histogram of the object. If a segmentation is consistent, the difference between the histogram of an object at time $t$ and an averaged histogram calculated on the basis of the temporal neighbours of the same object is small. This temporal difference is calculated in this work using four metrics to compare histograms. A small difference in histograms indicates a good segmentation.

Let us now see with more detail the metrics proposed in [11].

#### 6.4.1 Intra-frame Colour Metrics

The performance metrics presented in [11] are based on the following assumption:

- Object boundaries coincide with colour boundaries.
In order to evaluate the performance on the basis of the above assumption, the colour of the pixels just inside and just outside of the estimated object boundary can be compared. In order to define the just outside and just inside, normal lines of length $L$ are drawn from the object boundary at equal intervals towards the outside and the inside of the object as shown in Figure 6.4 (a). The end points are marked as $p_O^i$ and $p_I^j$. The colour difference metric $d_{CB}(t)$ is defined by Equations 6.11 and 6.12:

$$0 \leq d_{CB}(t) = \frac{1}{K_t} \sum_{i=1}^{K_t} d_{CB}(t, i) \leq 1$$  \hspace{1cm} (6.11)$$

$$d_{CB}(t, i) = \frac{||C_O^i(t) - C_I^j(t)||}{\sqrt{3 \times 255^2}}$$  \hspace{1cm} (6.12)$$

Where $K_t$ is the total number of normal lines drawn to the boundary of the object in frame $t$ and $C_O^i(t)$ is the average colour calculated in the $M \times M$ neighbourhood of the pixel $p_O^i(x, y, t)$. $C_I^j(t)$ is defined similarly.

The colour metric for the whole sequence is:

$$0 \leq D_{CB} = f(d_{CB}(t), t = 1, \ldots, T)$$  \hspace{1cm} (6.13)$$
6.4. Evaluation without Ground Truth

Figure 6.5: Diagram of the weighting function \( f(r) \) used to compute contrast on the object boundary which depends on the distance \( r \) from the boundary.

Following the philosophy of this evaluation metric, a different method is developed to evaluate the average in colour in the areas just inside and just outside the object boundaries. For each pixel belonging to the boundary a circular area is defined. In this area, the region of support that belongs to the object \( A_I \) and the region of support that belongs to the foreground \( A_O \) are considered, as shown in Figure 6.4 (b). A weighted average is calculated using function \( f(r) \):

\[
f(r) = \begin{cases} 
  r^{-\alpha} \cos^\beta \left( \frac{\pi r}{2r_{\text{max}}} \right) & \text{if } r < r_{\text{max}} \\
  0 & \text{otherwise}
\end{cases}
\]  

(6.14)

Where \( r = \sqrt{x^2 + y^2} \) denotes the distance from the neighbourhood centre and \( \alpha \) and \( \beta \) are positive integers. A graphical representation of one of these filters is given in Figure 6.5, for \( \alpha = 3 \) and \( \beta = 0 \), which are the parameter values used for experimentation in this work. The influence of function \( f(r) \) decreases with the distance to the boundary pixel that is the centre of the area (as shown in Figure 6.5) and is then normalised by the area of the region of support.

The average is calculated according to Equation 6.15:
\[ C(r) = \frac{\sum_{i \in A} f(r_i) c(r_i)}{\sum_{i \in A} f(r_i)} \]  

Where \( c \) indicates any of the three colour components taken into account, in this case using the \( Y'CbCr \) colour space, and \( r_i \) is the Euclidean distance between pixel \( p_i \) and the object boundary. The average of the three colour components is taken into account. The contrast and the relative metric are calculated as above. This modification in the calculation of the average allows the metric to be more sensitive to errors committed along the boundaries.

The difference in the approach followed by [11] and the approach proposed in this work is shown in Figure 6.4. It is possible to see that in (a) there is an error \( E_b \) introduced by the choice of the distance \( L \) of the centroid of the square windows used to calculate the averages of the colour components and the dimensions of the windows themselves. In the case presented in (b), the areas \( A_I \) and \( A_O \) used to calculate the averages of the colour components follow accurately the boundaries of the object.

An example of performance of this metric is shown in Figure 6.6. Given the incorrect segmentation in Figure 6.6 (b), the evaluation of the contrast as given by metric 6.15 is shown in (c), where lighter values of the image correspond to higher contrast. It is possible to notice that in points where the segmentation is incorrect, the metric value is close to zero. This fact can be used not just as an overall indication of the performance of the segmentation, but also to determine locally which regions of the boundary need refinement.

### 6.4.2 Inter-frame Colour Metrics

The inter-frame colour metrics are intended to verify the consistency of the segmentation through the sequence. The way to achieve this purpose is to verify the constancy of the colour histogram corresponding to the same moving objects in different frames [11]. This method can be therefore used as a metric for tracking performance.

For a given frame, this metric is based on a number of comparisons between the colour histogram of the current frame at time \( t \) and a comparative averaged frame. In this
6.4. Evaluation without Ground Truth

Figure 6.6: Original frame of test sequence (a), possible segmented object (b) and evaluation of contrast at the boundaries of the segmented object (c).

work, the comparative averaged frame is obtained from the average of the contributions of frames at time \((t - 1)\) and at time \((t + 1)\). The histograms are calculated in the \(Y'CbCr\) space. Let us define \(H_t(j)\) as the histogram corresponding to the current frame at time \(t\), for \(j = 1, \ldots, B\), where \(B\) is the number of bins used for the histogram. Let us also indicate by \(H_{t,av}\) the averaged comparative frame, as given by Equation 6.16:

\[
H_{t,av}(j) = \frac{H_{t-1}(j) + H_t(j) + H_{t+1}(j)}{3} \quad \text{for} \quad j = 1, \ldots, B
\]  

(6.16)

For convenience, a single one dimensional histogram has been calculated for each frame, appending all three colour component contributions. For each colour plane 256 bins have been used, therefore each histogram is constituted by \(B = 3 \times 256\) bins.

Let us consider the following definitions.

\[
R_1 = \sqrt{\frac{N_{H_t}}{N_{H_{t,av}}}} \quad R_2 = \frac{1}{R_1}
\]  

(6.17)

\[
N_{H_t} = \sum_{j=1}^{B} H_t(j)
\]  

(6.18)

\[
N_{H_{t,av}} = \sum_{j=1}^{B} H_{t,av}(j)
\]  

(6.19)
Four metrics have been taken into account.

$L_1$ metric: The $L_1$ metric is defined according to Equation 6.22:

$$0 \leq d_{L_1}(H_t, H_{t,av}) = \frac{\sum_{j=1}^{B} |R_1 H_t(j) - R_2 H_{t,av}(j)|}{2\sqrt{N_{H_t}N_{H_{t,av}}}} \leq 1$$ (6.22)

$L_2$ metric: The $L_2$ metric is defined according to Equation 6.23:

$$0 \leq d_{L_2}(H_t, H_{t,av}) = \frac{\sum_{j=1}^{B} |R_1 H_t(j) - R_2 H_{t,av}(j)|^2}{NS_{H_t} + NS_{H_{t,av}}} \leq 1$$ (6.23)

$\chi^2$ metric: The $\chi^2$ metric is defined according to Equation 6.24:

$$0 \leq \chi^2(H_t, H_{t,av}) = \frac{\sum_{j=1}^{B} \frac{(R_1 H_t(j) - R_2 H_{t,av}(j))^2}{H_t(j) + H_{t,av}(j)}}{N_{H_t} + N_{H_{t,av}}} \leq 1$$ (6.24)

Histogram intersection (HI) metric: The histogram intersection metric is defined according to Equation 6.25 and 6.26:

$$0 \leq d_{HI}(H_t, H_{t,av}) = 1 - HI(H_t, H_{t,av}) \leq 1$$ (6.25)

Where

$$0 \leq HI(H_t, H_{t,av}) = \frac{\sum_{j=1}^{B} \min[H_t(j), H_{t,av}(j)]}{\min(N_{H_t}, N_{H_{t,av}})}$$ (6.26)

and $HI$ is the histogram intersection.

The $d_{L_1}$ and $d_{L_2}$ are the city block and the Euclidean distances respectively, applied to the measure of distances between corresponding bins and normalised to be in the range between $[0,1]$. 
6.4. Evaluation without Ground Truth

The $\chi^2$ metric measures the magnitude of difference in sample percentage. In statistics, it is used to compare if two bins are drawn from the same probability function [124]. This metric is normalised.

Measuring the intersection between two histograms means measuring the minimum error probability (in Bayes' sense), which is obtained as the overlap between two probability distributions [150]. To use it as a distance metric, it has to be converted to a non-intersection measure. This has been done defining $d_{HI} = 1 - HI$. This metric is normalised.

6.4.3 Motion Metric

The motion metric used in [11] is conceptually and practically similar to the colour intra-frame colour metric presented in Section 6.4.1. The assumption about the motion of the segmented object is the following:

- Motion vectors of the object just inside and just outside of the object boundary are different.

According to [11], the motion metric can be estimated according to Equations 6.27, 6.28 and 6.29:

$$0 \leq d_M(t) = 1 - \frac{\sum_{i=1}^{K_t} d_M(t, i)}{\sum_{i=1}^{K_t} w_i} \leq 1 \quad (6.27)$$

$$d_M(t, i) = d(v_D^0(t), v_I^j(t)) \cdot w_i \quad (6.28)$$

$$0 \leq w_i = R(v_D^0(t)) \cdot R(v_I^j(t)) \leq 1 \quad (6.29)$$

Where $v_D^0(t)$ and $v_I^j(t)$ denote the average motion vectors calculated in the $M \times M$ neigbourhood of the pixel $p_D^0(x, y, t)$ and $p_I^j(x, y, t)$, in an analogous way to the calculation of the colour averages as seen in Section 6.4.1. The distance $d(v_D^0(t), v_I^j(t))$
denotes the distance between the two average motion vectors. It is calculated according to Equation 6.30:

$$0 \leq d(v_H^i(t), v_I^j(t)) = \frac{\|v_H^i(t) - v_I^j(t)\|}{\|v_H^i(t)\| + \|v_I^j(t)\|}$$

(6.30)

$R(.)$ denotes the reliability of the motion vectors. Therefore the overall motion metric $d_M(t)$ is obtained as the sum of the differences in corresponding motion vectors just inside and just outside the motion boundary (a sort of motion contrast) weighted by the reliability of the same motion vectors and normalised by the sum of all the weights.

In this work, two modifications to the calculation of the motion metric have been made. The first difference is represented by the different way of calculating the averages of the motion vectors just inside and just outside as it has been presented in Section 6.4.1 with regard to colour averages (Equations 6.14 and 6.15).

The second modification regards the computation of the reliability of the motion vectors. In the work presented in [11], motion vector reliability is calculated using the backwards projection calculated on the basis of the available motion vectors of the current frame $t$ onto frame $t - 1$ and then calculating the difference in colour components between corresponding pixels of the frame $t$ (projected onto $t - 1$) and frame $t - 1$. This method is unable to provide any indication about the reliability of the vectors in case the error is committed in a relatively homogeneous region of the image where the motion boundary falls. On the other hand, in this work and in the literature available, texture measurements have been usefully used as reliability measures [88]. Therefore, in this approach the amount of texture contained in the region of support is used as a reliability criterion and the associated weight is the one described in Chapter 4 in Section 5.2.4. Moreover, the vectors classified as outliers do not take part in the calculation of the average, as anyway they would not provide useful information.

### 6.5 Quantitative Evaluation

In this section, quantitative evaluation of object-based segmentations obtained with the use of the method of Section 6.3 and Section 6.4 is carried out and compared with
6.5. Quantitative Evaluation

the state-of-the-art techniques discussed in Section 6.2. Results with and without the use of ground truth are given.

For test sequence *Renata*, only one object is present. Therefore the results are related to the quality of background-foreground segmentation, with the object being the human moving figure.

For test sequence *Mobile and Calendar*, three main objects are moving and have been evaluated separately. The train has a mainly translational horizontal motion, the ball has a translational and 2D rotational motion, with deceleration and acceleration, and the calendar has a mainly translational vertical motion.

For test sequence *Garden*, given that the motion is apparent and depends on the relative depth of the objects composing the scene, the object tree which exhibits the main motion has been evaluated.

The ground truth information has been obtained by manual segmentation of single frames of the sequence by non-expert individuals. A group of 3 people, with no background in image and video processing (or to the problem of segmentation), have been shown the test sequences for a number of times. They have been given the individual frames of each test sequence as images and with the support of a picture editing program (Corel Draw 8) instructed to cut out the shapes of the single objects moving in different ways in the scene, paying particular attention to the boundaries of the moving objects and filling in different objects with different colours (labels). In the case of test sequence garden, where the motion is apparent, they were instructed to cut out the tree. The masks used for the final evaluation have been obtained by majority voting. If a pixel has been deemed as belonging to the object by the majority of the 3 individuals, the pixel was accepted as belonging to the object mask. Examples of ground truth are presented in Appendix C.

Whenever in the evaluation tracking of a given object over a number of frames is required (that is to say, the correspondence between an object $O_t$ in frame at time $t$ and the same object $O_{t+i}$ at time $t+i$ has to be established), this has been done by forward or backward (in the case of $t-i$) projection, using the motion vectors associated with the pixels belonging to object $O_t$. 
6.5.1 Evaluation with Ground Truth

Results of the comparative evaluation with the use of ground truth are presented in Table 6.1 for test sequence Renata, Table 6.2 for test sequence Mobile and Calendar and Table 6.3 for test sequence Garden.

The misclassification penalty $D_P$ indicates the area of an object that has been incorrectly segmented in comparison with the ground truth. Either the object has been over-segmented or under-segmented. In this case, the lower the number the closer the segmentation to the actual object. The proposed method outperforms all the other methods, with the exception of the object ball in the Mobile and Calendar sequence. Overall it yields more consistent results for all objects and all sequences tested.

The shape penalty $D_S$ can be considered a refinement of the misclassification penalty. It reflects the accuracy of the segmentation in the reproduction of the overall shape of the object. Here, the lower the figure, the better the segmentation in representing the overall shape. The proposed method outperforms all the other methods and overall, it yields more consistent results for all objects and all sequences tested.

The motion penalty $D_M$ reflects the accuracy of the segmentation in the reproduction of the motion of the object. In this case the motion features taken into account are the horizontal and vertical components of the optical flow, so this measure reflects the accuracy of the representation of the 2D translational motion of the objects. Also in this case, the lower the figure, the better the segmentation. The performance of the proposed method is better than the others, including motion-based segmentation. This is because the motion-only segmentation can split objects that do not move rigidly. Therefore the average of the optic flow calculated inside the object can contain contributions from different motions. The use of more features increases the accuracy of the segmentation. However, the proposed method is more accurate than the other two spatio-temporal methods. This is because the use of information about edges and colour-textures renders it more robust to leakage and to the presence of boundaries of weak contrast. Additionally, the consistency check of the motion in adjacent frames renders it more robust to errors in the estimation of motion features. This method is overall more spatio-temporally consistent than the other spatio-temporal methods.
6.5. Quantitative Evaluation

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>DP</td>
<td>2.4973e-4</td>
<td>0.0015</td>
<td>0.0030</td>
<td>0.0284</td>
</tr>
<tr>
<td>DS</td>
<td>0.0130</td>
<td>0.0201</td>
<td>0.0248</td>
<td>0.0364</td>
</tr>
<tr>
<td>DM</td>
<td>0.0170</td>
<td>0.0393</td>
<td>0.0310</td>
<td>0.0793</td>
</tr>
</tbody>
</table>

Table 6.1: Results of evaluation with the use of ground truth for test sequence Renata.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>train</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DP</td>
<td>4.2002e-4</td>
<td>0.0029</td>
<td>0.0206</td>
<td>0.0035</td>
</tr>
<tr>
<td>DS</td>
<td>0.0398</td>
<td>0.0532</td>
<td>0.2389</td>
<td>0.0649</td>
</tr>
<tr>
<td>DM</td>
<td>0.0389</td>
<td>0.0543</td>
<td>0.2471</td>
<td>0.1034</td>
</tr>
<tr>
<td>ball</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DP</td>
<td>0.0034</td>
<td>2.8949e-4</td>
<td>0.8000</td>
<td>6.1496e-4</td>
</tr>
<tr>
<td>DS</td>
<td>0.0110</td>
<td>0.0113</td>
<td>0.8032</td>
<td>0.0240</td>
</tr>
<tr>
<td>DM</td>
<td>0.0624</td>
<td>0.1062</td>
<td>0.8479</td>
<td>0.2085</td>
</tr>
<tr>
<td>calendar</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DP</td>
<td>3.1500e-4</td>
<td>0.0011</td>
<td>3.2870e-4</td>
<td>0.0120</td>
</tr>
<tr>
<td>DS</td>
<td>0.0051</td>
<td>0.0088</td>
<td>0.0052</td>
<td>0.0244</td>
</tr>
<tr>
<td>DM</td>
<td>0.0227</td>
<td>0.0238</td>
<td>0.0065</td>
<td>0.1616</td>
</tr>
</tbody>
</table>

Table 6.2: Results of evaluation with the use of ground truth for test sequence Mobile and Calendar.

here compared. This element can be especially appreciated in the segmentation of the Mobile and Calendar sequence. In this sequence, some objects are segmented better and some objects worse than the other methods, but the performance obtained over all the objects is consistent. This is an important advantage for general purpose object segmentation.

6.5.2 Evaluation without Ground Truth

The intra-frame colour metric $d_CB$ represents the strength of colour contrast along the object boundaries. Therefore, the higher the figure, the better the segmentation.
Table 6.3: Results of evaluation with the use of ground truth for test sequence Garden.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( DP )</td>
<td>( 4.7138 \times 10^{-4} )</td>
<td>0.0327</td>
<td>0.0309</td>
<td>0.0015</td>
</tr>
<tr>
<td>( DS )</td>
<td>0.0324</td>
<td>0.0464</td>
<td>0.0653</td>
<td>0.0382</td>
</tr>
<tr>
<td>( DM )</td>
<td>0.0211</td>
<td>0.1793</td>
<td>0.1363</td>
<td>0.0573</td>
</tr>
</tbody>
</table>

This relies on the assumption that the object boundaries correspond to the strongest colour edges locally. This assumption is reasonable in the absence of ground truth but misleading. In fact, the object boundaries which represent the biggest obstacle to accurate segmentation are usually the ones that exhibit lower colour or texture contrast. This is the reason why an element of edge continuity over local weak contrast was introduced in the proposed algorithm. According to the criterion described by \( d_{CB} \), spatio-temporal methods [13, 14] are expected to perform better. Results here presented show that the proposed method does not yield the highest contrast of all. For most of the objects taken into account, the spatio-temporal segmentation methods [13, 14] perform better. However, there is a considerable variation of results obtained by these methods, including the total merging of the object ball for method [13] in most of the frames. Spatio-temporal methods, like the ones in [14, 13], are likely to set the object boundary to the strongest local edge in terms of colour or motion or a combination of the two features. This is because the methods rely on the maximisation of the contrast either in colour or in motion, or colour and motion at the same time. Therefore, a region edge that exhibits a strong contrast will be selected as object edge, because no element of edge continuity or complexity has been considered in the cited methods and in the evaluation metric used. All these considerations apply to the analysis of the first row of data, referring to \( D_{CB} \), in Table 6.5 for test sequence Renata, Table 6.6 for test sequence Mobile and Calendar and Table 6.7 for test sequence Garden.

The inter-frame colour metrics \( d_{l1}, d_{l2}, \chi^2 \) and \( d_{HI} \) (histogram intersection), compare object histograms over a three-frame time interval in order to verify the constancy and coherence of the segmentation over multiple frames. In this case, the lower the figure, the better the segmentation. Higher figures indicate higher dissimilarity between histograms taken at different times in the sequence. The data relative to these four
6.5. Quantitative Evaluation

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Renata</td>
<td>0.3641</td>
<td>0.3770</td>
<td>0.4548</td>
<td>0.3157</td>
</tr>
<tr>
<td>Mobile and Calendar</td>
<td>0.3749</td>
<td>0.4362</td>
<td>0.7312</td>
<td>0.5197</td>
</tr>
<tr>
<td>Garden</td>
<td>0.4141</td>
<td>0.5383</td>
<td>0.3964</td>
<td>0.5305</td>
</tr>
</tbody>
</table>

Table 6.4: Summarised results of performance evaluation gathered from intra-frame colour dissimilarity metrics.

metrics can be found detailed in data columns 2 to 5 of Table 6.5 for test sequence Renata, Table 6.6 for test sequence Mobile and Calendar and Table 6.7 for test sequence Garden.

On Table 6.4, the total score of the four colour metrics is summarised. The data is obtained from the averaged summation of the four metrics $d_{L1}$, $d_{L2}$, $\chi^2$ and $d_{HI}$ over the number of frames and objects in test sequences. The results on Table 6.4 show that the figures associated with the proposed algorithm are generally lower than all the other ones, with the exception of [14] for test sequence Renata and [13] for test sequence Garden. Moreover, the figure shows quite a constant value over the different sequences, while the other techniques show much higher variations. This means that the proposed technique is consistent over time and the results can be consistently reproduced over a number of different sequences. From the inspection of the results referring to the single intra-frame colour dissimilarity metrics in Tables 6.5, 6.6 and 6.7 it is also possible to notice that the figures associated with this method are usually close to, but lower than, the ones associated with the motion-only method considered [12]. This reflects the fact that the consistency of this method is dependent upon the consistency of the motion information, as it has been explained in Section 4.2.7.

The motion metric $d_M$ represents the strength of motion features contrast along the object boundaries. Therefore, the higher the figure, the better the segmentation. This criterion can be quite misleading, since the accurate boundaries of the optic flow are generally difficult to estimate and correspond to both the uncovered and the occluded part of background/foreground in the direction of the motion. This has been partially taken into account by the reliability weighting system. However, this can be a criterion for comparing different approaches, when the ground truth is not available. In this
Table 6.5: Results of evaluation without the use of ground truth for test sequence Renata.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_{CB}$</td>
<td>0.0292</td>
<td>0.0243</td>
<td>0.0442</td>
<td>0.0413</td>
</tr>
<tr>
<td>$d_{L1}$</td>
<td>0.3449</td>
<td>0.3688</td>
<td>0.4476</td>
<td>0.2938</td>
</tr>
<tr>
<td>$d_{L3}$</td>
<td>0.5332</td>
<td>0.5630</td>
<td>0.6544</td>
<td>0.5190</td>
</tr>
<tr>
<td>$\chi^2$</td>
<td>0.2360</td>
<td>0.2509</td>
<td>0.3328</td>
<td>0.1802</td>
</tr>
<tr>
<td>$d_{HI}$</td>
<td>0.3384</td>
<td>0.3254</td>
<td>0.3874</td>
<td>0.2699</td>
</tr>
<tr>
<td>$d_M$</td>
<td>0.0014</td>
<td>0.0028</td>
<td>0.0018</td>
<td>0.0012</td>
</tr>
</tbody>
</table>

case, the higher the figure, the better the segmentation, since the contrast in the motion features is higher. From the inspection of the data reported in the last row of Table 6.5 for test sequence Renata, Table 6.6 for test sequence Mobile and Calendar and Table 6.7 for test sequence Garden, the following observations can be made. Not surprisingly, the method that performs better in this case is the one that exploits motion only as the source of information. The proposed method presents an average performance compared with spatio-temporal methods [13, 14]. There is no indication that any of the other spatio-temporal methods ([13, 14]) performs better, from the perspective of the maximisation of the motion difference along boundaries.

### 6.6 Qualitative Evaluation

Finally, the segmentation results obtained using the proposed method are presented in Figure 6.7 for Renata, Figure 6.8 for Mobile and Calendar and Figure 6.9 for Garden, for the purpose of qualitative evaluation. The proposed method (shown in Figure 6.7 (m), (n), (o) and (p) for test sequence Renata, in Figure 6.8 (m), (n), (o) and (p) for test sequence Mobile and Calendar and in Figure 6.9 (m), (n), (o) and (p) for test sequence Garden) is compared against the results obtained my a motion-based method [12] and two spatio-temporal methods [13] and [14].

Let us first compare the proposed method with the motion-based method in [12], shown by Figure 6.7 (a), (b), (c) and (d) for test sequence Renata, by Figure 6.8 (a), (b), (c)
## 6.6. Qualitative Evaluation

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>train</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$D_{CB}$</td>
<td>0.0753</td>
<td>0.0664</td>
<td>0.0830</td>
<td>0.0667</td>
</tr>
<tr>
<td>$L_1$</td>
<td>0.2079</td>
<td>0.2953</td>
<td>0.8407</td>
<td>0.3121</td>
</tr>
<tr>
<td>$L_2$</td>
<td>0.4702</td>
<td>0.6144</td>
<td>0.9661</td>
<td>0.5733</td>
</tr>
<tr>
<td>$X^2$</td>
<td>0.1105</td>
<td>0.1813</td>
<td>0.7402</td>
<td>0.1770</td>
</tr>
<tr>
<td>$HI$</td>
<td>0.1860</td>
<td>0.2263</td>
<td>0.1508</td>
<td>0.2035</td>
</tr>
<tr>
<td>$d_M$</td>
<td>0.0041</td>
<td>0.0045</td>
<td>0.0014</td>
<td>0.0028</td>
</tr>
<tr>
<td>ball</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$D_{CB}$</td>
<td>0.0738</td>
<td>0.0599</td>
<td>merged</td>
<td>0.0622</td>
</tr>
<tr>
<td>$d_{L_1}$</td>
<td>0.3679</td>
<td>0.4406</td>
<td>merged</td>
<td>0.8012</td>
</tr>
<tr>
<td>$d_{L_2}$</td>
<td>0.5514</td>
<td>0.8066</td>
<td>merged</td>
<td>0.9281</td>
</tr>
<tr>
<td>$X^2$</td>
<td>0.2610</td>
<td>0.3221</td>
<td>merged</td>
<td>0.7585</td>
</tr>
<tr>
<td>$d_{HI}$</td>
<td>0.3586</td>
<td>0.4009</td>
<td>merged</td>
<td>0.2922</td>
</tr>
<tr>
<td>$d_M$</td>
<td>0.0020</td>
<td>0.0014</td>
<td>merged</td>
<td>7.9625e^-4</td>
</tr>
<tr>
<td>calendar</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$D_{CB}$</td>
<td>0.0519</td>
<td>0.0452</td>
<td>0.0786</td>
<td>0.0659</td>
</tr>
<tr>
<td>$d_{L_1}$</td>
<td>0.4282</td>
<td>0.4203</td>
<td>0.4137</td>
<td>0.5286</td>
</tr>
<tr>
<td>$d_{L_2}$</td>
<td>0.8039</td>
<td>0.8127</td>
<td>0.7572</td>
<td>0.9255</td>
</tr>
<tr>
<td>$X^2$</td>
<td>0.3305</td>
<td>0.3223</td>
<td>0.3074</td>
<td>0.3976</td>
</tr>
<tr>
<td>$d_{HI}$</td>
<td>0.4141</td>
<td>0.3919</td>
<td>0.3983</td>
<td>0.2392</td>
</tr>
<tr>
<td>$d_M$</td>
<td>5.5625e^-4</td>
<td>0.0011</td>
<td>0.0014</td>
<td>7.7499e^-4</td>
</tr>
</tbody>
</table>

Table 6.6: Results of evaluation without the use of ground truth for test sequence *Mobile and Calendar*. 
In the proposed method, the boundaries follow more closely the actual contours of the object. This is to be expected since the purpose of using more features is precisely to improve boundary definition using local spatial information provided by colour and texture. However, this improvement is not guaranteed as it is possible to notice in the examples from the other two spatio-temporal segmentations [13] and [14]. In fact, the location of boundaries that fall in low contrast regions are problematic for these methods. In such low contrast areas, motion-based methods, like the one in [12], yield more accurate results.

Let us compare now the proposed method with the spatio-temporal methods in [13] and [14], shown by Figure 6.7 (e), (f), (g) and (h) for [13] and (i), (j), (k) and (l) for [14] for test sequence Renata, by Figure 6.8 (e), (f), (g) and (h) for [13] and (i), (j), (k) and (l) for [14] for test sequence Mobile and Calendar and by Figure 6.9 (e), (f), (g) and (h) for [13] and (i), (j), (k) and (l) for [14] for test sequence Garden.

It is possible to notice that the overall accuracy of the boundaries, obtained by both spatio-temporal methods [13] and [14] considered, is comparable. The proposed method provides a more coherent and perceptually meaningful representation, especially regarding its capability of generating consistent object boundaries from frame to frame. The inconsistencies in the segmentations are particularly evident in the Mobile and Calendar sequence, where consistent errors are present in the two spatio-temporal segmentations as over-merging in case of low contrast or loss of parts of the moving objects.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_{CB}$</td>
<td>0.0495</td>
<td>0.0377</td>
<td>0.0631</td>
<td>0.0444</td>
</tr>
<tr>
<td>$d_{L_1}$</td>
<td>0.3879</td>
<td>0.5566</td>
<td>0.4119</td>
<td>0.4985</td>
</tr>
<tr>
<td>$d_{L_2}$</td>
<td>0.6418</td>
<td>0.7780</td>
<td>0.5945</td>
<td>0.8102</td>
</tr>
<tr>
<td>$\chi^2$</td>
<td>0.2710</td>
<td>0.4327</td>
<td>0.2764</td>
<td>0.3976</td>
</tr>
<tr>
<td>$d_{HI}$</td>
<td>0.3557</td>
<td>0.3858</td>
<td>0.3027</td>
<td>0.4156</td>
</tr>
<tr>
<td>$d_M$</td>
<td>0.0010</td>
<td>0.0018</td>
<td>0.0012</td>
<td>0.0011</td>
</tr>
</tbody>
</table>

Table 6.7: Results of evaluation without the use of ground truth for test sequence Garden.
6.6. Qualitative Evaluation

The key capability of the proposed method seems to lie in the production of accurate boundaries, even in regions of low contrast. This represents a notable improvement with respect to the motion-based method [12] and the two spatio-temporal methods [13] and [14]. Failure to locate low contrast object boundaries in [13] and [14] is particularly evident in test sequences Renata and Mobile and Calendar. In Renata and Mobile and Calendar, the boundaries located are effectively the ones that present the highest local contrasts. However, these boundaries do not belong to the moving object, but to the nearest still object exhibiting high contrast boundaries. The reason for this behaviour is due to the lack of information about the continuity of the edges in the presence of noise or low contrast.

Another important issue related to the performance of every segmentation method is the choice of the number of regions that compose the final segmentation. A similar aspect has been already discussed in Section 4.2.5, with regard to the choice of the number of moving objects composing the sequence. The considerations discussed in Section 4.2.5 apply to the motion-based method [12] presented here for evaluation purposes. The number of objects in this case have been chosen by human interaction, since no automatic selection method was available in that framework. However, in the case of spatio-temporal and multiple-feature video object segmentation are composed by a spatial segmentation stage as well as a motion segmentation step. Therefore in the spatial segmentation stage of the segmentation, the necessity of evaluation of the number of regions composing the segmentation arises again, although differences arise according to the architecture, in case of parallel or hierarchical structure.

In the case of the parallel method in [13] motion and spatial (colour) segmentation are performed separately. There are two parameters involved in the choice of granularity of the segmentation: the number of moving objects, which is taken with regards to the temporal segmentation and the number of colour segments composing the spatial segmentation.

In the case of the hierarchical method in [14], motion and spatial (colour) segmentation are combined through the use of a joint spatio-temporal weight. In this method [14], areas of support of moving objects have to be determined, even if approximately. The
Table 6.8: Number of homogeneous and textured regions automatically selected by the parallel method.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Number of homogeneous regions</th>
<th>Number of textured regions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Renata</td>
<td>126</td>
<td>691</td>
</tr>
<tr>
<td>Mobile and Calendar</td>
<td>126</td>
<td>725</td>
</tr>
<tr>
<td>Garden</td>
<td>60</td>
<td>406</td>
</tr>
</tbody>
</table>

The method is based on the watershed segmentation algorithm, so it requires first the localisation of markers. The number of markers selected for the watershed segmentation influences the number of regions produced by the segmentation. The markers that fall between areas of support of two different moving objects are discarded, since they are not temporally homogeneous. The watershed segmentation is then performed on the basis of a similarity function expressed by Equation 5.5. In general, watershed segmentations consist of numerous segments. In the case of this method, the over-segmentation obtained using the joint spatio-temporal merging criterion is reduced by resorting to the motion information once more. Again, segments belonging to the area of support corresponding to the same moving object are labelled accordingly.

In the case of the proposed method, the number of regions composing the texture and colour segmentation is determined automatically. For the texture segmentation, this is determined by the connected components of pixels sharing the same colour texture label. For the colour segmentation, the number of segments is determined by the connected components created by the partitioning of the homogeneous areas obtained superimposing the morphologically filtered edge pixels. Since the proposed method determines automatically the number of textured and homogeneous regions composing the scene, in Table 6.8 these numbers are summarised for the results shown in Figures 6.7, 6.8 and 6.9. The data relate to number of regions averaged over the length of the sequence and rounded to the next integer towards positive infinity.

In the case of the results shown by Figure 6.7, by Figure 6.8 and by Figure 6.9, the segments related to the spatial part of the segmentation are summarised in Table 6.9. The data relate to number of regions averaged over the length of the sequence, except for [13], where the number of regions can be determined by the user, and rounded to
6.7 Conclusions

In this chapter, a performance evaluation of the overall system for object-based segmentation has been presented in comparison with state-of-the-art segmentation methods that employ single or multiple-features.

The comparison has been based on both objective and subjective assessment. The objective metrics have employed ground truth information using local colour and motion contrast and colour consistency criteria. Performance evaluation without ground truth employed a novel contribution that improves the sensitivity of contrast criteria in both colour and motion.

The evaluation shows that the proposed method compares favourably against state-of-
Figure 6.7: Comparison of the segmentation of frames 10, 20, 30 and 40 in test sequence Renata for the segmentations obtained with the use of [12] in (a), (b), (c) and (d); with the use of [13] in (e), (f), (g) and (h); with the use of [14] in (i), (j), (k) and (l) and with the use of the method here proposed in (m), (n), (o) and (p).
Figure 6.8: Comparison of the segmentation of frames 10, 20, 30 and 40 in test sequence Mobile and Calendar for the segmentations obtained with the use of [12] in (a), (b), (c) and (d); with the use of [13] in (e), (f), (g) and (h); with the use of [14] in (i), (j), (k) and (l) and with the use of the method here proposed in (m), (n), (o) and (p).
Figure 6.9: Comparison of the segmentation of frames 10, 20, 30 and 40 in test sequence Garden for the segmentations obtained with the use of [12] in (a), (b), (c) and (d); with the use of [13] in (e), (f), (g) and (h); with the use of [14] in (i), (j), (k) and (l) and with the use of the method here proposed in (m), (n), (o) and (p).
6.7. Conclusions

The-art methods employing motion and colour as segmentation features. The results are accurate and consistent and, most importantly of all, the segmentation does not depend on user selected parameters, such as the number of regions composing the segmentation.
Chapter 7

Conclusions

This chapter provides a summary of the motivations, main issues addressed and achievements of this thesis.

7.1 Summary

In recent years, the evolution of telecommunications and the increasing expectations of users have led to an unprecedented demand for access to video content. An elementary way of accessing video content is at a low level, for example at a level of visual primitives, by means of video object (also indicated as moving object) segmentation. This accounts for the decomposition of video into meaningful components that are characterised by different motion parameters. A higher-level representation of content can be achieved by enriching the process with semantic descriptors which may support further analysis. This can only be done for specific applications, where the context is known, and/or resorting to user interaction.

The main objective of this work has been to design a strategy to achieve a generic object-based segmentation. Only low-level attributes contributed the segmentation, no user interaction or a priori knowledge was required. However, the defining characteristic of the approach has been the utilisation of a multiplicity of low-level features for the description of a scene. In this work, colour, texture, motion and edges have been
employed as features of interest. The segmentation obtained in such a way is used for the analysis of arbitrary-content visual material. The main issues addressed in the design of object-based segmentation methods for new multimedia services were the following, according to [1]:

- **Flexibility** of segmentation criteria: the framework should be able to provide segmentation using different features.

- **Simplicity** of the processes employed: the complexity has to be kept low in order to allow for real-time or near real-time processing.

- **Generality**: the procedures must be able to provide a reasonable result in an automated way in a variety of real life footage, since the material can be varied and no a priori assumptions can be exploited.

- **User interaction**: the user must be allowed to change the criteria of the segmentation to suit specific needs.

- **Accuracy** and **Meaningfulness**: the objects extracted must respond to perceptually significant criteria, must be reasonably accurate and must have good correspondence with constituent scene components, within a small error limit.

In relation to the above objectives, this thesis had the following structure.

In Chapter 2, a review of state-of-the-art object-based segmentation techniques was presented. Not all low-level features available for description have been employed. Motion grants a global high level description of the object, which can be inhomogeneous in terms of colour and/or texture. The initial approach to object-based segmentation was therefore based on the use of motion information only. In order to increase the accuracy of the segmentation, spatio-temporal techniques were developed. Here, motion, while remaining the principal source of information, has been coupled with spatial intra-frame information like colour, texture and edges. Although numerous approaches exist, which achieve object-based segmentation by combination of motion and colour information (spatio-temporal methods), only a few methods use a higher number of
features. Such multiple-feature methods are usually extremely complex from a computational point of view and they are mostly semi-automatic, requiring user input in defining a (usually high) number of parameters and continued supervision.

In Chapter 3, the main segmentation tools used were described. The RSST segmentation framework has been chosen as the baseline segmentation strategy, as it allows for flexibility in the merging criteria. While being essentially a region growing approach, it requires no seeds or markers to initiate the segmentation. Additionally, it provides a complete and hierarchical description of an input frame, which is stored in a data structure called the Shortest Spanning Tree (SST). In the choice of segmentation strategy, an important consideration has been the relationship between the chosen method and the Human Visual System (HVS). A simplified model of the functional organisation of the visual cortex was considered towards influencing the system architecture presented in Chapter 5.

In Chapter 4, methods of extracting and segmenting features like motion, texture and colour were discussed. The first step was the selection of extractors and descriptors of single features and the study of single-feature segmentation. This chapter was roughly divided into three parts: one concerned with motion segmentation, one concerned with texture segmentation and a third concerned with colour segmentation. The RSST framework was used for the purpose of motion segmentation, first on the basis of the translational optic flow field obtained by a robust estimator and second combining the optic flow field with multiple motion features obtained from the curl and the divergence of the optic flow field. The issue of the temporal consistency of the motion segmentation was also addressed developing a global object descriptor against which the frame to frame segmentation error could be quantified and rectified. Textural feature extraction and texture-based segmentation was addressed in the second part of the chapter. The RSST was used for the purpose of texture segmentation using various features obtained from the co-occurrence matrix. This approach proved unsatisfactory due to mixing effects of different textural classes at region boundaries. An important contribution was the utilisation of mathematical morphology in order to extract textured regions with accurate boundaries and eliminate the problem of mixing between differently textured regions. Further progress was achieved by generating textural feature descriptors that
were incorporated in the RSST. The third part of the chapter discussed colour segmentation. RSST was used for colour segmentation using RGB and $Y'CbCr$ colour primaries. Furthermore, the textural description were extended to colour textures.

In Chapter 5, the problem of multiple-feature segmentation was addressed. This chapter was divided into two parts. In the first part, a hierarchical approach to multiple-feature segmentation was performed with the use of the RSST. In the second part, two approaches, inspired by parallel paradigms to multiple-feature object-based segmentation, were developed.

Exploiting the flexibility of the RSST framework, various methods of combining motion, colour and texture have been investigated, especially in relation to the adaptation of current state-of-the-art weighting strategies [14, 88] to the RSST framework. The results obtained were comparable with those of other methods, with the added advantage of having a complete and flexible description of the scene as provided by the Shortest Spanning Tree. However, the performance of this approach (and others similar to it) proved unsatisfactory. It was observed that output segments did not have good correspondence with the intuitive notion of a scene object. Often, the output needed to be over-segmented in order to be accurate, so the advantage of using a multiplicity of features was not evident.

Therefore, an alternative approach to the problem, shifting from hierarchical to parallel methods, was undertaken. Keeping the RSST as the baseline segmentation strategy, a modular, hybrid hierarchical architecture was developed. In such an architecture, motion maintained a position of dominant importance and the segmentations of texture and colour were performed separately in a single-feature fashion. Separate modules performed single-feature segmentation using motion, texture and colour, and then the combination of sub-sets of features. A feedback mechanism allowed the single-feature segmentations to be refined with the progressive incorporation of other features. The final moving object was obtained by fusion of all available features with the use of a set of rules. The architecture could be defined as hierarchical, since it established a hierarchy of features and segmentations in each module are performed with the use of RSST, which is a hierarchical segmentation method. The architecture could also be defined
as hybrid, since elements of parallelism were present, such as the parallel processing of spatial features relating to different moving objects. This architecture bore a degree of similarity with models approximating information processing by the HVS, especially with regard to modularity and feedback. It yielded accurate and consistent results in the experiments carried out. It was generic and it facilitated user-interaction. However, it still depended considerably on user-defined parameters.

The ultimate aim was to devise an automated way of extracting a generic moving object, with reasonable accuracy and with the possibility of user interaction. This was achieved in the last part of Chapter 5, with a modification of the hybrid hierarchical architecture. The elements of parallelism present in the hybrid hierarchical method evolved into a fully parallel paradigm. A good correspondence between HVS functional units and the proposed architecture consisting of single-feature processing units was achieved. Additionally, edge continuity was exploited as an additional feature. This architecture yielded results comparable with those of the hybrid/parallel architecture in a far simpler and automated way, since it did not employ sequence-dependent parameters.

In Chapter 6, the comparative evaluation of the parallel architecture was carried out taking into consideration three state-of-the-art moving object segmentation methods. Objective and subjective (qualitative) evaluation criteria were employed. The objective performance evaluation criteria were further investigated and modified to ensure more responsiveness of the metrics used along object boundaries. Using these evaluation criteria, the proposed parallel architecture performed favourably in comparison with other motion-based and spatio-temporal segmentations in terms of accuracy and consistency.

### 7.2 Contributions

The purpose of this work was to perform object-based segmentation which is general in the sense that can be applied to a variety of scenes, without relying on a priori information.

A major focus of this work was to formulate strategies achieving meaningful segmentation using different image features in an accurate but relatively simple way, therefore
fulfilling the need for simplicity and generality.

Limiting complexity has been a feature throughout the work, in the choice of motion model, moving object descriptors, textural descriptors and colour spaces.

Obtaining meaningful objects for the human viewer was achieved incorporating elements of the functional modelling of the HVS. A simplified description of functionalities of the visual cortex has been developed in Section 3.5.1.

Aspects of the HVS considered were the hierarchical organisation of functional areas, the presence of feedback from higher to lower levels (so called layers) in the hierarchy and a limited element of parallelism between temporal and spatial processing. This allowed for the formulation of the hybrid hierarchical model in Section 5.3. Given the satisfactory results obtained with this model, the similarity with the perceptual model has been extended to a fully parallel paradigm in Section 5.4.

In both the hybrid hierarchical model and the fully parallel model, single-feature segmentations were performed independently by dedicated processing modules. These were organised in a hierarchy so that outputs of a processing module constituted inputs for a successive module. The multiple-feature integration was performed using a set of appropriate rules. A high degree of modularity and flexibility in the choice of feature descriptors and segmentation strategies were achieved, leaving room for further development and fine-tuning. Inputs and outputs to single-feature processing areas could be substituted by user-dependent parameters or user-selected intermediate segmentations: this allows for much-needed (in the field of multimedia) user interaction and the possibility of introducing higher level semantics without compromising generality. Moreover, in the case of the fully parallel model, the granularity of the segmentation was determined automatically. This functionality is extremely important in the case of post-production applications, where a notable amount of data must be browsed quickly and analysed reproducibly for the purpose of editing, indexing or archiving.

Specific contributions of this thesis were the following:

- The development of a simplified model capturing the functionalities and organisation of the visual cortex, and extending beyond the processing which characterises
7.2. Contributions

- The consideration of higher-level motion features and their incorporation in the RSST framework (Section 4.2.4).

- The development of a strategy to check the temporal consistency of motion segmentation in time. This strategy employed global moving object descriptors and the resulting error as an indicator of temporal consistency (Section 4.2.7).

- The development of a simple and effective method for texture analysis based on edge detection and mathematical morphology which allowed for the detection of textured areas while retaining accurate boundaries (Section 4.3.4).

- The extension of the above method of texture analysis to colour texture (Section 4.4.3).

- The utilisation of the RSST framework for the purpose of multiple-feature segmentation. The RSST has been used in the past for the purpose of single feature segmentation with the use of intensity, colour, motion and texture. The application of RSST to the multiple-feature problem is novel, requiring the design of appropriate cost functions and relative weighting schemes. The performance of such a segmentation method was comparable with the ones presented in the literature and showed a greater level of flexibility (Section 5.2).

- The development of a modular architecture for object-based segmentation based on single-feature and multiple-feature segmentations, using colour, texture and motion. This scheme is effective in the accurate segmentation of objects, producing consistent results. It is general and allows for user interaction (Section 5.3).

- The development of a fully parallel method for object-based segmentation. It follows the perceptual model of organisation presented in Section 3.5.1. It provides an automatic, generic segmentation of moving objects, using fewer parameters than the hybrid hierarchical model, while still allowing for user interaction. It is also more comprehensive than the hybrid hierarchical paradigm, since it employs four features: motion, colour, texture and edges (Section 5.4).
• The improvement of the performance evaluation criterion in [11] in order to increase the sensitivity of objective evaluation metrics to errors made along moving object boundaries (Section 6.4.1).

7.3 Future Work

In this section, possible developments to the thesis described are listed. The list is obviously not exhaustive, the reason being that object-based (multiple-feature) segmentation methods are complex paradigms which involve different algorithms for different features. This is particularly evident in parallel paradigms like the one presented in Section 5.4. Even hierarchical methods, which use a joint spatio-temporal similarity criterion for segmentation, rarely consist of only the region growing part. In fact, moving objects have to be located and roughly segmented to extract motion features. Tracking of homogeneous regions may follow, in order to provide temporal consistency towards object definition.

This work has included topics related to feature extraction and segmentation for motion, colour, texture and edges. In regard to complex object-based segmentation architectures, both hierarchical and parallel architectures have been investigated. All aspects of the vision research this project has been dealing with represent a topic of ongoing research and can be further improved.

Some ideas, that might be interesting to pursue and derive directly from the results of the investigation carried out in this work, are the following:

• Implementation of the perceptual parallel model within the RSST framework: The model presented in this work in Section 5.4 is parallel in terms of the separation of temporal and spatial processing, but can also be parallelised by means of separate processing of textured and homogeneous regions. In this regard, the work presented in [104], where a parallel version of RSST method has been presented, could be used. In [105], fixed size geometrical partitions have been used to parallelise RSST. Using the parallel paradigm, partitions would be created automatically
and would correspond to perceptually meaningful texture classes and homogeneous regions, separated by continuous edges. Using the RSST, the granularity of the segmentation could be controlled by the user, while the parallel segmentation algorithm proposed in Section 5.4 automatically provides parameter settings for a good general-purpose segmentation.

- **Adaptive selection and tuning of hierarchical segmentation parameters:** Hierarchical multiple-feature segmentation was addressed within the RSST framework. The results were not satisfactory, because of the difficulty in finding the right combination of weights for a given frame in a given sequence of arbitrary content. However, the weighting scheme employed was a static one. One could attempt to use a dynamic weighting scheme and minimise the set of weights against a measure of degradation using standard minimization techniques. This strategy was not considered in this work because of the additional complexity that would have introduced in the overall scheme. Therefore, this should be combined with efforts towards a faster implementation of the RSST and the minimisation scheme. In the motion segmentation obtained with the RSST and the use of multiple motion features presented in Section 4.2.4, all the motion features have been attributed the same weight and therefore the same importance towards guiding the correct segmentation. However, not all the motion features have the same importance at the same time, i.e., in terms of segmentation, at the same iteration. It would be useful to devise a method to select features of interest at a particular stage of the segmentation and discard other features that may have been useful towards the definition of the segment for a range of iterations, but outside that particular interval bring no contribution to the segmentation. This could also be of general interest for hierarchical segmentation schemes.

- **Utilisation of perceptual colour space and perceptual colour texture filtering:** The experiments were conducted using the $Y'CbCr$ and RGB colour spaces. A better choice for the purpose of modelling of the HVS would be a perceptually uniform colour space like $La^*b^*$ [108, 8], which would be consistent with colour filtering properties of the HVS [108]. Perceptual colour texture filtering has been pre-
Presented in [86], where a *tower* (as opposed to *pyramid*) of filtered colour textures, representing textures at different scales, was used. In Section 4.3.4, a multi-scale approach to texture characterisation was presented by using banks of structural elements within the framework of mathematical morphology. Perceptual filtering could be used as a pre-processing step, prior to morphological filtering, in order to obtain perceptually meaningful texture labelling. Calculation of edges could be done in perceptual colour components as well.

- **Interactivity in hybrid hierarchical and parallel methods:** User interaction is allowed in principle by the hybrid hierarchical and parallel methods proposed. To realise it to its full potential, the proper interface between each module and the user should be designed and implemented into a system. The system should comprise two modalities: automatic and semi-automatic (user supervised) segmentation. The semi-automatic modality should allow for easy manipulation of parameters such as *granularity* or *number of objects*, the selection of object, areas and/or features of interest and especially the smooth transition from automatic and semi-automatic modes.

- **Definition of additional and/or semantic rules:** The set of rules employed by the rule processor was very simple, since the basic motivation of the project was to obtain a segmentation independent from a priori knowledge. However, it would be desirable to expand the set of rules according to a *definition* of an object which takes into consideration temporal consistency, smoothness of trajectory, size and shape deformation and memory of feature statistics to name but a few. An object might also be defined by the user, incorporating higher level semantics into the rule processor.

- **Incorporation of explicit tracking of object:** In this project segmentation was discussed. Some elements of object tracking have been used in order to obtain temporally consistent objects and labelling of objects present in the same sequence. The motivation for not considering tracking directly was the tendency of certain spatio-temporal approaches, e.g. in [14, 88, 61], to use tracking in order to escape from the difficulties of obtaining an accurate stand-alone segmentation.
However, once a reasonable segmentation is obtained, tracking contributes to the completeness of the description of the scene.

- **Development of object descriptors for tracking and performance evaluation purposes**: In Section 4.2.7, simple global object descriptors were used for the evaluation of motion segmentation. In Section 6.4, object definition criteria were used to implement performance evaluation metrics. In both cases, limited, albeit reasonable, definitions of a video object have been employed. Much research should be devoted to the listing of characteristics of video objects and their mathematical modeling.

As a final note, this segmentation system could be built into numerous applications for the purpose of video editing and browsing, educational resources, traffic monitoring and video surveillance. These are all exciting developments to be investigated.
Appendix A

Test Sequences

Throughout this dissertation, results were obtained using three test sequences: Renata, Mobile and Calendar and Garden. In this appendix, some technical details and generic description of these sequences are given.

The sequence format is a 4 : 2 : 2 format: the chrominance components are half the sampling rate of the luminance Y. The sampling pattern is represented in Figure A.1, where the squares indicate either the luminance or the chrominance components of each pixel. A chrominance component is present every luminance component. The colour space used is $Y' C_B C_R$ according to Recommendation (Rec.) ITU-R BT.601.

Rec. ITU-R BT.601 is the international standard for studio-quality digital video [8]. Luminance is coded in 8 bits and it has an excursion of 219 and an offset of +16, so black is at 16 and white is at 235 [8]. The chrominances $C_B$ and $C_R$ are coded in 8 bits, with excursion of ±112 and offset of +128 resulting in a range from 16 to 240.

When using the $RGB$ colour data are coded at 8 bits, with no head-room and foot-room. This means black is at 0 and white is at 255. According to Rec. ITU-R BT.601 the conversion matrix from $Y' C_B C_R$ to $RGB$ is the following [8].

\[
\begin{pmatrix}
Y_{601}^* \\
C_B \\
C_R
\end{pmatrix} = \begin{pmatrix}
16 \\
128 \\
128
\end{pmatrix} + \frac{1}{256} \begin{pmatrix}
65.738 & 129.057 & 25.064 \\
-37.945 & -74.494 & 112.439 \\
\end{pmatrix} \begin{pmatrix}
R_{255} \\
G_{255} \\
B_{255}
\end{pmatrix}
\]

(A.1)
### Appendix A. Test Sequences

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Source</th>
<th>Resolution</th>
<th>Format</th>
<th>Colour space</th>
<th>Length</th>
<th>Interlacing</th>
<th>Main features</th>
<th>Camera Motion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Renata</td>
<td>video, 25 Hz</td>
<td>720 pixels line, 576 lines</td>
<td>4:2:2</td>
<td>$Y' C_B C_R$</td>
<td>50 frames</td>
<td>yes</td>
<td>sharp edges, artificial texture</td>
<td>pan, zoom</td>
</tr>
</tbody>
</table>

Table A.1: Technical information about test sequence *Renata*.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Source</th>
<th>Resolution</th>
<th>Format</th>
<th>Colour space</th>
<th>Length</th>
<th>Interlacing</th>
<th>Main features</th>
<th>Camera Motion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobile and Calendar</td>
<td>video, 25 Hz</td>
<td>720 pixels line, 576 lines</td>
<td>4:2:2</td>
<td>$Y' C_B C_R$</td>
<td>50 frames</td>
<td>yes</td>
<td>sharp edges, saturated colours</td>
<td>pan</td>
</tr>
</tbody>
</table>

Table A.2: Technical information about test sequence *Mobile and Calendar*.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Source</th>
<th>Resolution</th>
<th>Format</th>
<th>Colour space</th>
<th>Length</th>
<th>Interlacing</th>
<th>Main features</th>
<th>Camera Motion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Garden</td>
<td>video, 25 Hz</td>
<td>720 pixels line, 576 lines</td>
<td>4:2:2</td>
<td>$Y' C_B C_R$</td>
<td>50 frames</td>
<td>yes</td>
<td>weak edges, similar colours, natural texture, details</td>
<td>pan</td>
</tr>
</tbody>
</table>

Table A.3: Technical information about test sequence *Garden*.
Where the use of (•) indicates components which had been gamma corrected. The gamma correction has already been discussed in Section 4.4 with reference to CRTs. Gamma correction is also applied at the acquisition, for example at the video camera end. In a television delivery chain, gamma correction needs to be applied with a value of the exponent of 0.45, as specified in Rec. 709, to achieve near-unity end-to-end gain.

*Interlacing* is a scanning pattern, according to which each video frame is scanned during two successive passes, from left to right and from top to bottom, producing two fields. The lines of the two fields interlace to produce the whole frame, hence the name. In Figure A.2, this scanning system is represented graphically. The scanning lines are not perfectly horizontal in order to facilitate the sweep back to the start of the following line. Each field is composed of either odd or even lines of the total frame. The second
field is temporally delayed from the first field of half the frame time.

In this work, the experiments have been conducted taking into account fields rather than frames. Co-sized fields (e.g. the first field of each frame) have been used. In order to restore the original resolution of the frame, vertical up-sampling has been carried out using bilinear interpolation.

The test sequences used for experiments are briefly described below.

*Renata* is a head-and-shoulders sequence, showing a person moving in front of a complex textured background. The background consists of synthetic textures both in luminance and colour. The motion of the foreground object (person) is rendered more complicated by the fact the the object undergoes sudden deceleration and 3-D rotational movement along an axis parallel to that of the image plane. The sequence has very low contrast and contains very similarly textured areas between background and foreground in some frames. There is also camera panning and zooming. Fields from test sequence *Renata* are shown in Figure A.3.

*Mobile and Calendar* is a synthetic sequence rich in colours and textures. It presents three main objects moving. The ball is rotating along an axis perpendicular to that of the image. At a later stage it stops and then it starts moving again, pushed by the toy train. The toy train is moving in a roughly horizontal direction and uncovers the wall and part of the calendar. The calendar is moving behind the train and in the upper part of the frame, following a roughly vertical direction. There is slight camera panning. Fields from test sequence *Mobile and Calendar* are shown in Figure A.4.

*Garden* (flower garden) is an outdoors scene rich in texture. There is no major object in motion, the movement is apparent and depends on the panning of the camera and scene depth, which is considerable. A tree appears to move from the right to the left at a higher speed than the objects further away from the observer. The lawn behind the tree also appears to move, so that the foreground part of the lawn seems to move faster and it appears as a separate object from the farther layers of the lawn. This sequence does not have a high contrast and has very similar textures in parts of the tree trunk and parts of the wooden fences of the surrounding gardens. The motion of the branches and single leaves of the tree(s) is also quite complex (non-rigid). Fields
Figure A.3: Example of original fields for test sequence *Renata*.

Figure A.4: Example of original fields for test sequence *Mobile and Calendar*.

from test sequence *Garden* are shown in Figure A.5.

Figure A.5: Example of original fields for test sequence *Garden*. 
Appendix B

Performance Evaluation Metrics

Throughout this work, two kinds of commonly used performance evaluation metrics have been used: the Mean Squared Error (MSE) and the Peak Signal-to-Noise Ratio (PSNR). The two metrics are related as it is possible to notice from the definition given in the following. The PSNR is widely used for compression applications.

Let $f(n)$ be an 8 bit image, so that possible values for pixel $n$ range from 0 to 255. Let $N$ be the number of pixels constituting the image $f(n)$. Let $\tilde{f}(n)$ indicate an approximation of the image $f(n)$. The error committed by approximating the function $f(n)$ with $\tilde{f}(n)$ can be measured using the MSE metric as expressed by Equation B.1.

$$MSE = \frac{1}{N} \sum_{n=1}^{N} (f(n) - \tilde{f}(n))^2$$ \hspace{1cm} (B.1)

The Definition B.1 can be extended easily to a sequence of $M$ frames, by adding a factor representing the average over the entire sequence. The same definition can also be extended if function $f$ is a vector.

The PSNR is the inverse of the MSE, measured in dB. The definition of PSNR is given by Equation B.2.

$$PSNR = 10 \log_{10} \frac{255^2}{MSE}$$ \hspace{1cm} (B.2)
These metrics do not correlate with HVS perception as observed by [43] among others. Therefore, it is useful to couple them with a subjective evaluation of the quality of the segmented image. On the other hand, given the lack of a metric representing perceived quality of the segmented image, the MSE (and PSNR as a related metric) are accepted as good numerical references [148] for performance evaluation of still image segmentation and hierarchical spatio-temporal video image segmentation.

They are not however representatives of the properties of video objects, as explained in Chapter 6. Therefore, in the mentioned Chapter 6, specific object-based performance evaluation metrics are discussed.
Appendix C

Ground Truth Sequences

In Figure C.1 examples of the ground truth fields used for objective performance evaluation of test sequences presented in Chapter 6 are given.
Figure C.1: Example of ground truth for test sequences Renata, Mobile and Calendar and Garden.
Bibliography


ternationale de Normalisation ISO/IEC JTC1/SC29/WG11 (Coding of Moving Pictures and audio, 1996.


