



UNIVERSITY OF  
CAMBRIDGE



DEPARTMENT OF  
ENGINEERING

# Edge-based Motion Segmentation

Paul Alexander Smith

Jesus College

August 2001

A dissertation submitted to the University of Cambridge  
for the degree of Doctor of Philosophy.





---

# Abstract

---

Motion segmentation is the process of dividing video frames into regions which have different motions, providing a cut-out of the moving objects. Such a segmentation is a necessary first stage in many video analysis applications, but providing an accurate, efficient motion segmentation still presents a challenge.

This dissertation proposes a novel approach to motion segmentation, using the image *edges* in a frame. Using edges, a motion can be calculated for each object. Edges provide good motion information, and it is shown that a set of edges, labelled according to the object motion that they obey, is sufficient to completely determine the labelling of the whole frame, up to unresolvable ambiguities. The areas of the frame between edges are divided into regions, grouping together pixels of similar colour, and these regions can each be assigned to different motion layers by reference to the edges. The depth ordering of these layers can also be deduced. A Bayesian framework is presented, which determines the most likely region labelling and depth ordering, given edges labelled with their probability of obeying each of the object motions.

An efficient implementation of this framework is presented, initially for segmenting two motions (foreground and background) using two frames. The Expectation-Maximisation algorithm is used to determine the two motions and calculate the label probability for each edge. The frame is then segmented into regions. The best motion labelling for these regions is determined using simulated annealing.

Extensions of this simple implementation are then presented. It is demonstrated how, by tracking the edges into further frames, the statistics may be accumulated to provide an even more accurate and robust segmentation. This also allows a complete sequence to be segmented. It is then demonstrated that the framework can be extended to a larger number of motions. A new hierarchical method of initialising the Expectation-Maximisation algorithm is described, which also determines the best number of motions.

These techniques have been extensively tested on thirty-four real sequences, covering a wide range of genres. The results demonstrate that the proposed edge-based approach is an accurate and efficient method of obtaining a motion segmentation.



---

# Acknowledgements

---

I have to thank my supervisor, Roberto Cipolla, for his guidance and inspiration, and thanks go to all the members of the Vision group at CUED for their friendship and for providing such an enjoyable and stimulating working environment. In particular, I must thank Tom Drummond for countless discussions and periods of brain-bashing. Without those, this work might well have taken a very different form.

The research described in this dissertation was funded by the EPSRC, with a CASE award from AT&T Laboratories, Cambridge. I am indebted to Andy Hopper and AT&T for their timely and generous support, both financial and technical. I would like to thank Ken Wood for his encouragement and for being willing to read drafts of this work from an early stage. Thanks also go to Dave Sinclair for ideas and direction when starting this PhD.

My two colleges, Jesus and Robinson, have both provided additional funding, as well as an inspirational surroundings.



---

# Declaration

---

This dissertation is the result of my own original work and does not include anything done in collaboration with others, apart from where acknowledged in the text. It has neither been submitted in whole nor in part for a degree at any other university. It contains 117 figures and approximately 64,000 words, including appendices, bibliography, footnotes, tables and equations. The following publications were derived from this work:

## Conference presentations

P. Smith, T. Drummond and R. Cipolla. Edge tracking for motion segmentation and depth ordering. In *Proc. 10th British Machine Vision Conference*, volume 2, pages 584–593, Nottingham, September 1999.

P. Smith, T. Drummond and R. Cipolla. Motion segmentation by tracking edge information over multiple frames. In *Proc. 6th European Conference on Computer Vision*, volume 2, pages 396–410, Dublin, Ireland, June/July 2000.

P. Smith, T. Drummond and R. Cipolla. Segmentation of multiple motions by edge tracking between two frames. In *Proc. 11th British Machine Vision Conference*, volume 1, pages 342–351, Bristol, September 2000.



---

# Contents

---

Summary . . . . .	i
Acknowledgements . . . . .	iii
Declaration . . . . .	v
Contents . . . . .	vii
List of Tables . . . . .	xiii
List of Figures . . . . .	xvii
<b>1 Introduction</b>	<b>1</b>
1.1 What is motion segmentation? . . . . .	1
1.2 Why do motion segmentation? . . . . .	2
1.2.1 Video coding and compression . . . . .	2
1.2.2 Video indexing . . . . .	3
1.2.3 Video interpretation and annotation . . . . .	4
1.2.4 Other applications . . . . .	5
1.3 Synopsis . . . . .	6
1.3.1 Contributions . . . . .	6
1.3.2 Thesis outline . . . . .	8
<b>2 A survey of motion estimation and segmentation</b>	<b>11</b>
2.1 Introduction . . . . .	11
2.2 Motion estimation . . . . .	11
2.2.1 The pixel-based approach . . . . .	12
2.2.2 The feature-based approach . . . . .	16

2.2.3	Pixel vs feature-based methods . . . . .	19
2.3	Motion segmentation . . . . .	19
2.3.1	Motion field segmentation . . . . .	20
2.3.2	Layered motion . . . . .	22
2.3.3	Layered motion extraction . . . . .	23
2.3.4	Enforcing spatial coherency . . . . .	26
2.3.5	Using intensity information . . . . .	26
2.3.6	The region merging approach . . . . .	29
2.3.7	The depth of objects . . . . .	31
2.4	Summary . . . . .	32
<b>3</b>	<b>Edge-based motion segmentation</b>	<b>35</b>
3.1	Introduction . . . . .	35
3.2	Edges for motion estimation . . . . .	36
3.2.1	Edges and motion estimation . . . . .	36
3.2.2	Edges and motion segmentation . . . . .	38
3.3	Edges and regions for motion segmentation . . . . .	41
3.3.1	Prior assumptions . . . . .	41
3.3.2	Conditions for a correct segmentation . . . . .	42
3.3.3	Edge labels . . . . .	42
3.3.4	Region labels . . . . .	43
3.3.5	Depth ordering . . . . .	45
3.4	Unsolvable ambiguities . . . . .	46
3.4.1	Missing occluding boundary . . . . .	46
3.4.2	No T-junctions . . . . .	47
3.5	Bayesian formulation . . . . .	48
3.5.1	Parameters and maximum likelihood solution . . . . .	48
3.5.2	Estimating the motions $\Theta$ . . . . .	49
3.5.3	Estimating the labellings $\mathbf{R}$ and $\mathbf{F}$ . . . . .	50
3.6	Summary . . . . .	52
<b>4</b>	<b>Implementation for two motions, two frames</b>	<b>55</b>
4.1	Overview . . . . .	55
4.2	Finding edges . . . . .	57
4.3	Estimating motions from edges . . . . .	59
4.3.1	The aperture problem . . . . .	59
4.3.2	Finding a match . . . . .	61
4.3.3	Motion models . . . . .	62



4.3.4	Lie group formulation . . . . .	64
4.3.5	Solution by re-weighted least squares . . . . .	66
4.4	Multiple motion estimation using EM . . . . .	68
4.4.1	Dominant vs simultaneous multiple motion estimation . . . . .	68
4.4.2	The Expectation-Maximisation algorithm . . . . .	71
4.4.3	Initialisation . . . . .	73
4.4.4	Expectation: Calculating edge probabilities. . . . .	74
4.4.5	Maximisation: Calculating motions . . . . .	79
4.4.6	Convergence . . . . .	80
4.5	Finding regions . . . . .	82
4.5.1	Choice of segmentation scheme . . . . .	82
4.5.2	Voronoi seeded image segmentation . . . . .	83
4.6	Labelling regions and finding the layer order . . . . .	85
4.6.1	Region probabilities from edge data . . . . .	86
4.6.2	Region prior . . . . .	86
4.6.3	Solution by simulated annealing. . . . .	89
4.6.4	A word on probabilistic region labelling . . . . .	92
4.7	Summary . . . . .	93
<b>5</b>	<b>Evaluation</b>	<b>95</b>
5.1	Introduction . . . . .	95
5.2	Test sequences . . . . .	95
5.3	Qualitative and quantitative results . . . . .	96
5.4	Foreman sequence . . . . .	96
5.5	Tennis sequence . . . . .	100
5.6	Coastguard sequence . . . . .	102
5.7	Car sequence . . . . .	104
5.8	Ensemble results . . . . .	106
5.9	Comparative results . . . . .	111
5.9.1	Pixel-based approaches . . . . .	111
5.9.2	Region-based approaches . . . . .	114
5.10	Summary . . . . .	117
<b>6</b>	<b>Extension to multiple frames</b>	<b>119</b>
6.1	Introduction . . . . .	119
6.2	Accumulating evidence: Continued tracking . . . . .	120
6.2.1	Initialisation . . . . .	120
6.2.2	Occlusion . . . . .	121

6.2.3	Combining statistics . . . . .	122
6.3	Using cumulative statistics to segment a frame . . . . .	123
6.4	Templated segmentation of a sequence . . . . .	124
6.5	Deformable segmentation . . . . .	126
6.5.1	Segmenting a new frame: Propagating edges . . . . .	126
6.5.2	Accumulating evidence: Propagating sample points . . . . .	128
6.5.3	Accumulating edge probabilities . . . . .	129
6.5.4	Continued deformable segmentation of a sequence . . . . .	131
6.6	Evaluation . . . . .	131
6.6.1	Foreman sequence . . . . .	132
6.6.2	Tennis sequence . . . . .	136
6.6.3	Coastguard sequence . . . . .	139
6.6.4	Car sequence . . . . .	142
6.6.5	Ensemble results . . . . .	145
6.7	An application: Background mosaicing . . . . .	147
6.7.1	Implementation . . . . .	148
6.7.2	Examples . . . . .	148
6.8	Summary . . . . .	150
<b>7</b>	<b>Extension to multiple motions</b>	<b>151</b>
7.1	Introduction . . . . .	151
7.2	Recursive Splitting EM . . . . .	152
7.2.1	Initialising an extra model . . . . .	153
7.2.2	Determining the best number of models . . . . .	156
7.2.3	Implementation for edge-based motion segmentation . . . . .	157
7.3	Region labelling under multiple motions . . . . .	159
7.4	Global optimisation: EMC . . . . .	160
7.5	‘One region’ constraint . . . . .	163
7.6	Implementation overview . . . . .	164
7.7	Evaluation . . . . .	165
7.7.1	Overview . . . . .	165
7.7.2	Model selection . . . . .	165
7.7.3	Library sequence . . . . .	166
7.7.4	Car & Van sequence . . . . .	168
7.8	Discussion . . . . .	170

<b>8 Conclusion</b>	<b>173</b>
8.1 Summary . . . . .	173
8.2 Discussion . . . . .	174
8.3 Suggestions for further work . . . . .	175
8.4 A final word: Edges vs pixels . . . . .	176
<b>A Parameter estimation</b>	<b>179</b>
A.1 Motion estimation . . . . .	179
A.2 Least squares solution . . . . .	180
A.3 M-estimators . . . . .	181
A.4 Regularisation . . . . .	184
A.5 Normalisation . . . . .	185
<b>B Maximum likelihood estimation via EM</b>	<b>187</b>
B.1 The EM algorithm . . . . .	187
B.2 Estimation of mixture model parameters . . . . .	188
B.2.1 Finding the weights $c_\ell$ . . . . .	190
B.2.2 Finding the model parameters $\theta_\ell$ . . . . .	190
B.3 The M-stage for edge motion parameters . . . . .	191
<b>C The independence of sample points</b>	<b>193</b>
C.1 Introduction: Edges and sample points . . . . .	193
C.2 Errors under different motions . . . . .	193
C.2.1 The effect of assuming independence . . . . .	194
C.3 Errors along an edge . . . . .	195
<b>D Complete multiple-frame results</b>	<b>197</b>
D.1 Introduction . . . . .	197
D.1.1 Image sequences . . . . .	197
D.1.2 Algorithm . . . . .	198
D.1.3 Presentation of results . . . . .	198
D.2 Results . . . . .	198
<b>Bibliography</b>	<b>233</b>
<b>Author Index</b>	<b>249</b>



---

# List of Tables

---

4.1	System overview . . . . .	57
4.2	Parameters used for Canny edge detection . . . . .	59
4.3	Planar transformations . . . . .	63
4.4	The hierarchy of two-dimensional transformations . . . . .	63
4.5	Motion estimation algorithm . . . . .	69
4.6	The EM algorithm for multiple motion estimation using edges . . . .	73
4.7	Multiple motion estimation . . . . .	80
4.8	Optimisation of region labelling and layer ordering . . . . .	89
4.9	Simulated annealing . . . . .	91
5.1	Percentage of pixels correctly segmented using two frames . . . . .	110
6.1	EM initialisation for frames after the first two . . . . .	121
6.2	Deformable segmentation of a new frame . . . . .	131
6.3	Percentage of pixels correctly segmented over multiple frames . . . .	146
7.1	Recursive splitting EM . . . . .	158
7.2	Selecting the best number of motions: Minimum Description Lengths	166
A.1	M-estimators . . . . .	183
A.2	Matrix normalisation . . . . .	186
C.1	Correlation of sample point distances under each motion . . . . .	194



---

# List of Figures

---

1.1	Example of motion segmentation . . . . .	2
1.2	Example edge-based segmentations from this dissertation . . . . .	7
2.1	Pixel-based motion estimation . . . . .	13
2.2	Feature-based motion estimation . . . . .	16
2.3	Layered motion example sequence . . . . .	23
2.4	Region merging example . . . . .	29
3.1	Image intensity and edges in a frame . . . . .	37
3.2	A per-pixel motion labelling . . . . .	39
3.3	Segmented image regions . . . . .	40
3.4	Tracking and labelling edges . . . . .	43
3.5	Labelling regions from edges . . . . .	44
3.6	Labelling a T-junction . . . . .	45
3.7	Unsolvable ambiguity: Missing occluding boundary . . . . .	47
3.8	Unsolvable ambiguity: No T-junction . . . . .	47
4.1	Foreman segmentation from two frames . . . . .	56
4.2	The aperture problem . . . . .	59
4.3	Edge tracking example . . . . .	60
4.4	Sample points in a frame . . . . .	60
4.5	Evaluating edge probabilities . . . . .	74
4.6	Edge statistics . . . . .	76

4.7	Sample point likelihood ratio . . . . .	78
4.8	Probability of a good match . . . . .	78
4.9	EM convergence . . . . .	81
4.10	Edge probabilities as EM converges . . . . .	81
4.11	Example region segmentation . . . . .	84
4.12	Region labelling solution with a flat prior . . . . .	87
4.13	Region statistics . . . . .	88
4.14	Region prior . . . . .	88
4.15	Solutions under different layer orderings . . . . .	90
4.16	Region labelling as simulated annealing converges . . . . .	90
4.17	Probabilistic region labelling . . . . .	92
5.1	Foreman sequence . . . . .	97
5.2	Foreman segmentation from two frames . . . . .	98
5.3	Tennis sequence . . . . .	101
5.4	Tennis segmentation from two frames . . . . .	101
5.5	Coastguard sequence . . . . .	103
5.6	Coastguard segmentation from two frames . . . . .	103
5.7	Car sequence . . . . .	105
5.8	Car segmentation from two frames . . . . .	105
5.9	Examples from the AT&TV sequences . . . . .	107
5.10	Comparison with Wang and Adelson: FlowerGarden sequence . . . . .	112
5.11	Comparison with Ayer and Sawhney: FlowerGarden sequence . . . . .	112
5.12	Comparison with Weiss and Adelson: FlowerGarden sequence . . . . .	112
5.13	Comparison with Ayer and Sawhney: Tennis sequence . . . . .	113
5.14	Comparison with Elias: Tennis sequence . . . . .	113
5.15	Comparison with Elias: Coastguard sequence . . . . .	113
5.16	Comparison with Moscheni and Dufaux: Foreman sequence . . . . .	115
5.17	Comparison with Moscheni and Dufaux: Tennis sequence . . . . .	115
5.18	Comparison with Dufaux et al. : Tennis sequence . . . . .	115
5.19	Comparison with Moscheni and Bhattacharjee: Tennis sequence . . . . .	116
5.20	Comparison with Bergen and Meyer: Foreman sequence . . . . .	116
6.1	Detection of sample point occlusion . . . . .	122
6.2	Foreman sequence: Cumulative statistics . . . . .	123
6.3	Templated segmentation of the Foreman sequence . . . . .	125
6.4	Templated segmentation of the Tennis sequence . . . . .	125
6.5	Templated segmentation of the Car sequence . . . . .	125



---

6.6	Propagation of edges to the next frame . . . . .	127
6.7	Propagation of sample points between frames . . . . .	128
6.8	Cumulative statistics for propagated edges . . . . .	130
6.9	Foreman segmentation of the next frame . . . . .	133
6.10	Segmentation of the <b>Foreman</b> sequence . . . . .	135
6.11	Tennis segmentation of the next frame . . . . .	137
6.12	Segmentation of the <b>Tennis</b> sequence . . . . .	138
6.13	Coastguard segmentation of the next frame . . . . .	140
6.14	Segmentation of the <b>Coastguard</b> sequence . . . . .	141
6.15	Car segmentation of the next frame . . . . .	143
6.16	Occluded sample points in the <b>Car</b> sequence . . . . .	144
6.17	Segmentation of the <b>Car</b> sequence . . . . .	144
6.18	Mosaic of the background to the <b>Car</b> sequence . . . . .	148
6.19	Mosaic of the background to the <b>Simpsons</b> sequence . . . . .	149
7.1	Initialisation by splitting . . . . .	154
7.2	Initialising with too few models . . . . .	154
7.3	Fitting three motions . . . . .	159
7.4	Three-motion edge probabilities and region labels . . . . .	160
7.5	Constrained edge labels . . . . .	162
7.6	Overview of the EMC algorithm . . . . .	162
7.7	Example EMC solution . . . . .	163
7.8	Implementation for multiple motions . . . . .	165
7.9	Library sequence . . . . .	167
7.10	Library segmentation from two frames . . . . .	167
7.11	<b>Car&amp;Van</b> sequence . . . . .	169
7.12	<b>Car&amp;Van</b> segmentation from two frames . . . . .	169
C.1	Markov chain transition probabilities. . . . .	196



# Introduction

---

### 1.1 What is motion segmentation?

Motion is an important cue in vision. Visual motion attracts the attention—it identifies something that is happening, something that is changing. To a moving observer, motion offers additional information since the relative motion between the observer and objects identifies their spatial relationship to each other. In the context of Computer Vision the analysis of the changes between two images of the same scene, or across a sequence of video frames, is a prelude to many important areas of research which try to recreate these human visual processes.

A segmentation of an image is a division into separate areas, usually to some purpose, so that each different segment has a distinct meaning. A *motion segmentation* is the division of a video frame into areas obeying different motions. Each moving object in the scene should exhibit a different motion on the image plane of a camera, and the aim is to cut-out each of these objects in the image. This divides the image into semantically meaningful regions upon which higher-level analysis may be performed.

Such a segmentation is illustrated in Figure 1.1, which shows an ideal segmentation of a sequence used throughout this dissertation. Here the man moves in front of the background and so a motion segmentation would identify the area of the image occupied by him as separate from the rest of the image. Having done this, these two areas of the image can be analysed separately, or compared, in a range of different applications.



Figure 1.1: *Example of motion segmentation.* The man moves against the background, and this relative motion is used to segment the image into two different regions. The motion is determined for each region, and in this case is marked with an arrow (the background is stationary).

## 1.2 Why do motion segmentation?

Motion segmentation is not an end in itself, but is an enabling technology, motivated by a variety of different applications. A survey of the most common applications is presented here. Video is increasingly being stored in digital form: produced by portable digital ‘camcorders’; or recorded from broadcast television with a digital recorder or a TV capture card in a personal computer;<sup>1</sup> or downloaded from the Internet. Many of the applications described below, using motion segmentation techniques, are already beginning to leave the research laboratories and starting to enter the marketplace.

### 1.2.1 Video coding and compression

The storage space occupied by digital video is a major concern, and some form of compression is almost always required. A typical television signal contains  $720 \times 576$  pixels at 25Hz, and so at a conservative 12 bits per pixel this uncompressed signal requires a data rate of 124 million bits per second. By comparison, an audio compact disc requires only 1.4 million bits per second. The capability to provide video on CD-sized objects has required two innovations: Digital Versatile Discs (DVDs) with

<sup>1</sup>Digital video recorders are now commercially available, for example systems from TiVo (<http://www.tivo.com>) or ReplayTV (<http://www.replaytv.com>).

a capacity 13 times greater than that of CDs; and video compression. DVD-Video uses MPEG-2 compression [81, 99], which provides a compression ratio of around 40:1. Naturally, customers will continue to demand even higher quality for even less bandwidth, and here motion segmentation can provide assistance.

Video compression is achieved by observing that successive frames in a video sequence are usually very similar. The content of one frame can largely be predicted by extrapolating from the previous frame and so, rather than store each frame individually, the next in a sequence may be encoded by describing just the changes between the frames. Existing video compression schemes (e.g. MPEG-1 [80, 99] and MPEG-2) are *block-based*—the image is segmented into a regular array of rectangles. The motion of each block is described, and then any remaining pixel changes within the block. Motion segmentation can help here because it enables usefully-shaped areas, i.e. the entire moving object, to be considered rather than arbitrary rectangular blocks. This enables more sophisticated coding techniques to be applied.

One such technique is to provide a *mosaic* of the background—a single image of the backdrop to the scene, made by stitching together the background region of each frame [75, 121]. This may be coded once and then only the foreground objects and residuals need to be coded per frame. This approach can in fact yield a higher quality backdrop than that in the original sequence, since the information from the various frames can be combined to make a super-resolution image of the background [77]. The MPEG-4 standard [82, 129] is designed to use this layered approach (background plus foreground objects). It was initially designed for multimedia presentations, but automatic motion segmentation techniques enable conventional videos to be encoded using this standard.

The semantics of the scene may also be used to provide higher rates of compression in cases where, perhaps, the overall frame quality is not as important as representing some particular areas well. One example of this is video telephony over low bandwidth links. In this case it is important to represent the speaker, particularly their head, at full frame-rate and high resolution, whereas the background does not require this level of quality. A motion segmentation of the sequence can direct the bandwidth to the areas where it is most required.

### 1.2.2 Video indexing

Once video compression has enabled a large number of videos to be stored on a personal computer, or on the World Wide Web, this introduces the additional problem of searching this repository to find a particular video, or part of a video. Powerful

tools exist to index and search text archives, most notably the World Wide Web (using search engines such as Google or Alta Vista).<sup>2</sup> The automatic retrieval of images is an active research field, with many systems proposed, including IBM's 'QBIC' [55] and Berkeley's 'Blobworld' [34]. The aim of *video indexing* is to be able to perform similar indexing and searching on video sequences.

Video indexing applications already exist which use text queries derived from closed captions (e.g. Virage's 'VideoLogger',<sup>3</sup> and AT&T's 'AT&TV' [98]), but the real interest is in schemes which consider the image content. By performing a motion segmentation of a frame, the moving objects and the background can be analysed independently. A simple implementation can consider a single frame from a sequence: queries can be posed in terms of the shape, colour or texture in exactly the same way as for a query on a still image, only now they may be phrased as properties of the background, or a foreground object.

The use of a mosaic of the background (as also used for image coding applications) is commonly proposed, for example in [57, 73]. By forming a single image of the backdrop to the scene, this generates another still image which may be used to identify the context of the scene. Perhaps more importantly, however, the foreground objects' frame-to-frame motions may be marked on this single image and described against this common frame of reference (this is called either a 'synoptic frame' [57], or a 'synopsis mosaic' [73]). Object motions and their interactions can then form part of the video description. This mosaic representation also provides an intuitive means of summarising and browsing through a sequence.

### 1.2.3 Video interpretation and annotation

The analysis of object motion, and of the interaction between different objects, is an essential stage in providing a higher-level interpretation of the sequence. This is useful not only for the purposes of indexing and retrieving a sequence, but also in other expert systems.

There have been a number of systems developed for automatically annotating (and commentating on) sports events, including soccer matches [3, 72, 161], American football [71], and basketball [120]. In a more serious domain, summarising video-taped presentations has been considered [86]. A major commercial application of motion analysis and interpretation techniques is surveillance, typically identifying unusual behaviour. Example applications include traffic monitoring [32, 53], or

---

<sup>2</sup>See <http://www.google.com> and <http://www.altavista.com>

<sup>3</sup><http://www.virage.com.products/videologger.html>

interactions in car parks [114] or on university plazas [108]. A key element to all of these applications is *semantic event detection*. This requires first identifying motion events—changes in the motion of a foreground object, or the joining or splitting of two areas with two different motions—and then inferring some meaning to these events, labelling important incidents with some warning or commentary.

Not all of the authors referenced above present a complete system—many concentrate on the analysis *after* the motion segmentation, but all these systems require the identification of the motions in the scene and the location and extent of each of those motions. They all require a motion segmentation.

### 1.2.4 Other applications

One other application has already been mentioned, that of *resolution enhancement*. A video sequence gives a series of views of the same object. If each of these similar views can be identified and the relevant sections of the frames registered (by using the known image motion to map them to a common co-ordinate frame), the combined information can be used to generate an enhanced image [77]. A related application is that of *video restoration*. Where a video film has become degraded, perhaps by noise or something more severe, such as dirt or scratches, these errors must first be detected (by comparison with the expected image predicted from neighbouring frames), and then repaired by using the relevant areas from other frames [90, 100].

A further application is that of sequence interpolation, which is particularly useful for *frame-rate conversion*. The European video standard is 25Hz, whereas in Asia and North America it is 30Hz, and so to convert European videos for these markets, 6 images need to be generated for each 5 images in the original. This requires interpolating between most pairs of frames. If the image segmentation and the segment motions are known for these frames, the appropriate fraction of each motion may be applied to each segment. This approach may also be used to generate slow motion sequences.

If a high-quality cut-out motion segmentation can be achieved, it can be used in the video *special effects* industry as an alternative to the ubiquitous ‘blue screen’ which is currently used. The actors can be removed and placed in front of a new background, regardless of the original background. This process is occasionally done at present without a blue screen, by using a hand segmentation. Automatic motion segmentation techniques can automate or semi-automate this process.

## 1.3 Synopsis

### 1.3.1 Contributions

This dissertation concentrates on the initial motion segmentation problem, with the emphasis on both providing an accurate cut-out of foreground objects and obtaining this without excessive computation cost. To achieve this, a new approach is proposed which uses only the edges in the image. The dissertation makes the following novel contributions:

- The theory linking image edges and regions is developed. It is shown that edges, and region reasoning, are both necessary and sufficient to determine a complete segmentation, up to unsolvable ambiguities.
- A Bayesian approach to this new edge-based motion segmentation is derived.
- An implementation of this Bayesian edge-based motion segmentation technique is presented for the analysis and segmentation of two motions between two frames.
- The segmentation implementation uses a new image segmentation scheme developed by Sinclair [130]. Its integration with a motion segmentation scheme is novel, and an improvement has been made over the basic scheme.
- The use of multiple frames is advocated, to improve edge labels and resolve ambiguities. A novel approach is presented which allows deforming objects to be accurately segmented and to propagate, and accumulate, edge statistics between frames.
- This implementation is extended to segment multiple motions. A novel initialisation stage for the EM algorithm is presented, which avoids local minima and identifies the correct number of models (joint work with Tom Drummond and Rob Fergus). The Expectation-Maximisation (EM) algorithm [43] is also extended to include a constraint step for a global optimisation.

The implementations are tested on a wide range of video sequences, with excellent results. Figure 1.2 highlights some of the best of these.





Figure 1.2: *Example edge-based segmentations from this dissertation.* A selection of results produced by the system described in this dissertation, taken from Chapter 6 and Appendix D.

### 1.3.2 Thesis outline

This dissertation is organised as follows:

**Chapter 2** contains a survey of existing techniques. Motion estimation is introduced, including both pixel- and feature-based techniques, and then a review of motion segmentation schemes is presented. The literature shows that pixel-based schemes dominate motion segmentation, despite feature-based schemes also being effective for motion estimation. It indicates that the use of image edges and regions is necessary for an accurate segmentation, and that an approach using edge features would be robust and computationally efficient.

**Chapter 3** presents the major contribution of this dissertation—the use of edges for motion estimation and accurate layered motion segmentation. It is shown that if the edges in the image are labelled according to their motions, this is sufficient to label the entire image. The rest of the image is divided into regions using a *static* segmentation of the frame, and the logical reasoning which enables these regions to be labelled from the edges is developed. It is shown that such reasoning is necessary for a complete segmentation, as this enables the relative depth ordering to also be identified. A Bayesian framework is presented which allows a maximum likelihood segmentation of the frame to be performed.

**Chapter 4** describes an implementation of the Bayesian framework developed in the previous chapter. This novel algorithm segments a frame from a video sequence into two motions (foreground and background) using information from two frames (the frame to be segmented, and the next in the sequence). Two maximisation stages are required to find the Maximum Likelihood segmentation: this implementation uses Expectation Maximisation (EM) to find the edge motions, and then simulated annealing to label image regions. Nonetheless, the implementation is efficient, segmenting a frame in a few seconds on conventional hardware.

**Chapter 5** evaluates the performance of the two-motion, two-frame implementation. Four test sequences are considered in detail, and results from a further thirty sequences are also considered. A comparison with some other motion segmentation schemes is also presented.

**Chapter 6** extends the two-frame approach of the previous chapters to use information from more frames. This improves the reliability of the edge labelling,

resolves motion ambiguities, and enables the segmentation of a sequence of frames. An important contribution is the means by which edges and motion probabilities from previous frames can be propagated into the new frame to assist both the motion estimation and segmentation.

**Chapter 7** outlines how the implementation of the previous chapters can be extended to segment more than two motions. This requires the development of a new robust initialisation scheme for EM (to avoid local maxima), together with labelling constraints to resolve ambiguities. These constraints are integrated into the EM loop to give an ‘EMC’ algorithm, which is also described. The identification of the number of motions present, using the Maximum Description Length principle, is also performed as part of the initialisation scheme. It is shown that the edge-based segmentation framework can be generalised to segment a sequence containing any number of motions.

**Chapter 8** contains a summary of the dissertation and presents avenues for further research.



# A survey of motion estimation and segmentation

---

## 2.1 Introduction

There is a large body of existing work on the subject of motion segmentation, and on the wider issue of motion estimation—the measurement of motion in the image. At some point in the process, all motion segmentation schemes must also determine the motion in the scene, and in most cases the process is sequential: first motion estimation, and then motion segmentation. This survey gives an overview of the motion estimation process, and the different approaches available. The different approaches to motion segmentation are then considered.

## 2.2 Motion estimation

Almost all work on image sequences begins by trying to find out how the image changes with time, analysing how different elements in the frame move. This subject of *motion estimation* has been considered by many authors over the past twenty years; excellent reviews have been presented by S. M. Smith [135], and Barron et al. [8], so this section discusses only the most popular techniques.

Motion estimation techniques fall into two broad categories, referred to here as *pixel-based* and *feature-based*. Pixel-based schemes consider a minimisation of image

quantities (typically image gradients) over every pixel in the image.<sup>1</sup> This gives an estimate of the motion of each pixel in the image (e.g. Figure 2.1). Feature-based schemes concentrate on measuring the motion in areas where it can be measured reliably (e.g. Figure 2.2, later in this chapter). This motion can then, if necessary, be used to guide the estimation process in other regions of the image. There are strong proponents of each approach in the literature; a recent debate on the subject produced complementary papers from Torr and Zisserman [148] (pro-features) and Irani and Anandan [74] (pro-pixels). A transcript of the debate may be found in [152, pages 294–297].

In addition to two main approaches described above, the MPEG-1 and MPEG-2 video coding schemes [80, 81, 99] use a *block-based* approach to motion estimation. Here, the image is arbitrarily divided up into small blocks (typically  $16 \times 16$  pixels). For each block, a translational motion is estimated by making a search in the next frame for the most similar block, as described by Jain and Jain [83]. A block-based scheme only provides a coarse motion field, which is insufficient for motion segmentation. In addition, the emphasis in these techniques is on obtaining the best coding performance, rather than best representing the motion of the underlying object. Block-based techniques are, however, used as an intermediate stage in some pixel-based approaches, and these are described in the relevant parts of this chapter.

### 2.2.1 The pixel-based approach

#### The Brightness Change Constraint Equation

The starting point for most pixel-based techniques is the ‘brightness constancy constraint’ [65]. This makes the assumption that the intensity of points in the scene only changes slowly over time. This is only strictly true for Lambertian surfaces under time-invariant illumination, but is usually a satisfactory approximation.<sup>2</sup> In perhaps the best known work on pixel-based motion estimation, Horn and Schunk [66] expressed this constraint by saying that, to first order, the rate of change of intensity must be zero:

$$\frac{d}{dt}I(x, y, t) = 0 \quad (2.1)$$

---

<sup>1</sup>These schemes are commonly referred to in the literature as *direct methods* and their output as *optic flow*. However, the precise definition of these terms varies from author to author, and so in order to avoid confusion the term *pixel-based* will be the only one used in this dissertation.

<sup>2</sup>A Lambertian (or diffuse) surface is one which scatters light equally in all directions, so its appearance depends only on the illumination, and not on the viewing direction.

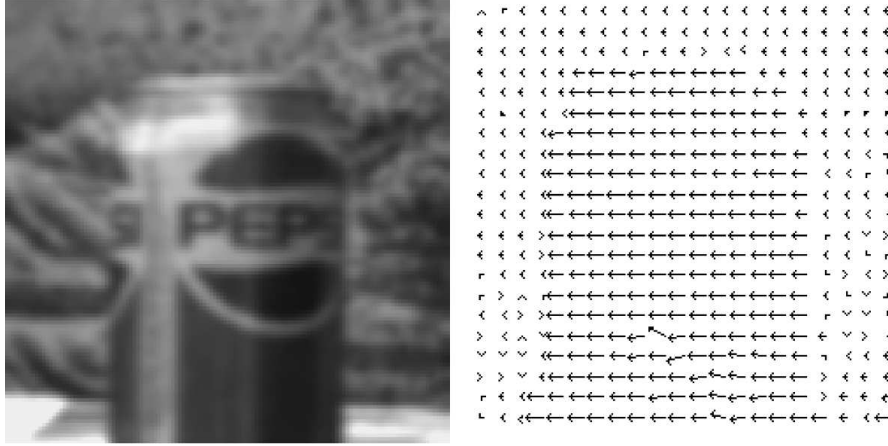


Figure 2.1: *Pixel-based motion estimation*. A motion field is computed across the whole image using spatiotemporal image gradients. Smoothing is required to determine a reasonable motion in areas of low gradient. (From Black and Anandan [17].)

which is accurate for small motions. They express this as the total derivative

$$\frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} + \frac{\partial I}{\partial t} = 0 \quad (2.2)$$

or

$$I_x u + I_y v + I_t = 0 \quad (2.3)$$

where  $(I_x, I_y)$  are the spatial derivatives of the image brightness,  $I_t$  is the difference between consecutive frames, and  $u(x, y)$  and  $v(x, y)$  are the components of the motion. Equation (2.3) is commonly known as the *brightness change constraint equation* (BCCE).

Equation (2.3) can be rewritten in vector form, using  $\mathbf{v} = (u \ v)^T$ ,

$$\nabla I \cdot \mathbf{v} = -I_t \quad (2.4)$$

which highlights the problem with gradient based methods: that the BCCE only provides a constraint on the component of motion *perpendicular* to the image gradient,  $\nabla I$ . This is an instance of the well-known ‘aperture problem’ [94], discussed further in Section 4.2 and, as a result of this, the BCCE cannot on its own fully determine the motion field. In motion estimation problems this is typically resolved either by smoothing or by parameterising the motion, both of which are described below.

### Smoothness constraints

In [66], Horn and Schunk resolved the aperture problem by an additional constraint which encouraged a smooth isotropic variation in the motion field. They defined an energy function which combined the BCCE with a second smoothness term, and found the motion field by an iterative approach.

This isotropic smoothness constraint has the clear disadvantage that it will perform poorly where there is a discontinuity in the motion field (either due to a sudden change in depth, or to an independently moving object). In these cases it will smooth over the discontinuity, which is particularly unwelcome in the case of motion segmentation when these discontinuities are exactly what need to be detected.

In [106], Nagel addressed this problem by introducing an ‘oriented smoothness’ constraint. He introduced a different smoothing cost term, which only penalises motions *along* the intensity gradient. Thus discontinuities are better preserved, and smoothing is only encouraged perpendicular to the gradient i.e. in the direction which is not constrained by the BCCE.

However, it is clear that the BCCE can only go so far in determining the motion on a pixel-by-pixel basis. It is only well defined in areas of the image with high gradient, and then it is the results from these areas which must then be spread into the other areas of the image. The computation of a dense flow field is an underconstrained problem, and to determine the field some assumptions must be made. Smoothness is only one possible assumption.

### Parameterised motion

An explicit assumption which could instead be made is to model the motion field for an object as a 2D parametric motion. All of the pixels which belong to an object should move in a similar manner, and this parametric modelling of the image motion is reasonable between the frames of a video sequence [5, 9, 78, 107]. Representing the vector of motion parameters by  $\alpha$ , this approach describes the image motion components  $u$  and  $v$  motions by the functions  $U(x, y, \alpha)$  and  $V(x, y, \alpha)$  respectively. The BCCE (2.3) then becomes

$$I_x U(x, y, \alpha) + I_y V(x, y, \alpha) + I_t = 0 \quad (2.5)$$

This can be solved directly for  $\alpha$  by standard parameter estimation techniques, given sufficient pixel measurements (i.e. at least as many pixels as there are pa-



rameters).<sup>3</sup> Parameterised motion models are a powerful solution to the motion estimation problem. They are used in the majority of existing motion segmentation schemes, usually as part of a *layered* model, described in Section 2.3.2.

### Image mosaicing

The pixel-based approach is commonly used in *image mosaicing* applications. Here the parameterised camera motion for a sequence is recovered.<sup>4</sup> This allows the images to be converted to a common co-ordinate frame, and to be stitched together into one large image. These mosaics have a number of applications, among them motion segmentation, where they are used to represent an image of the background, and foreground objects may be detected as outliers to these. Examples of this work can be seen in papers by Irani [73, 75, 76] and Sawhney and Ayer [121], while Szeliski presents a good overview of image mosaicing in [138].

### Coarse-to-fine estimation

The BCCE (2.3) relies on a first-order expansion of the intensity function and is only a good approximation when the motion is small (i.e. less than one pixel). This is insufficient for video sequences, which typically have a motion of several pixels. The motion range can be significantly enhanced by using an iterative coarse-to-fine approach, as, for example, suggested by Anandan in [2]. Here, the image is repeatedly filtered and sub-sampled to produce a Laplacian pyramid (Burt and Adelson [29]); typically three or four levels are used. The induced image motion decreases as the resolution decreases, and at the coarsest resolution level the motion estimation performs well, so it is here that the initial motion is estimated. Once the motion has been found at this level, the results are projected into the next resolution level and the motion refined. The process is repeated at each level of the pyramid until the motion field for the original full-resolution image is found. A more complete description of the coarse-to-fine approach can be found Bergen et al. [9].

Using this approach, motions of 10–15% of the image size can be accommodated [74]. This is then sufficient for video sequences, and forms a sound basis for dense motion estimation from video. With an initial estimate from some other approach (e.g. a feature-based approach), even larger motions may be handled.

<sup>3</sup>The parametric motion model most commonly used in motion estimation and segmentation is the *affine* model, which uses six parameters encompassing translation, rotation and shear [51, 64]. More details can be found in Section 4.3.3.

<sup>4</sup>The change between images from a camera fixed in space, but zooming and rotating, can be described by an eight parameter model: a 2D projective transformation, also known as a *homography* or *collineation* [51, 64, 138].

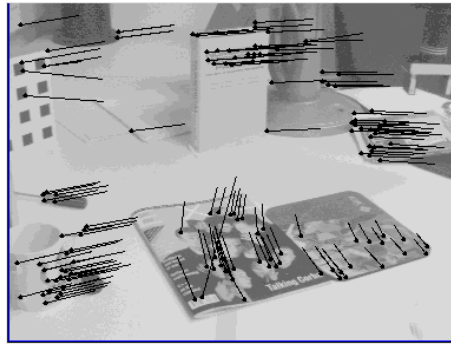


Figure 2.2: *Feature-based motion estimation.* Corner features are identified and these are matched between images to identify their motion. This gives a sparse representation of pixel motion, but only uses pixels whose motion can be well-determined. (Corners found using the Harris corner detector [62], matched using cross-correlation and filtered with the Median Flow filter [134].)

## 2.2.2 The feature-based approach

The pixel-based approaches discussed above rely on the image gradient to constrain the motion. This means that the motion can only be well resolved in areas of high image gradient, and the motion in other areas of the image must be found by smoothing, or by fitting a parametric model. Feature-based approaches acknowledge this problem by concentrating only on the areas of the image that are likely to yield good motion information. If necessary, the motion estimated from these features may then be used to guide estimation in the rest of the image.

In contrast to the global minimisation of the pixel-based approaches, which solves for both motion and correspondence simultaneously, feature-based methods separate the two. Features of interest are first detected and correspondences found (using image quantities such as cross-correlation), and then the motion is found.

### Feature extraction

Feature-based methods concentrate only on areas of the image which can be well-localised and tracked between images—features such as edges or corners in the image. The first stage in feature-based motion estimation is to identify these image structures.

The two main classes of features commonly used are *edges* and *corners* [51, 64]. ‘Edges’ are one dimensional image features—they are a chain of pixels where there is a sharp change in the image intensity in one direction. The standard algorithm for detecting edges is the one proposed by Canny [33], although there are many others. More popular than edges, however, are ‘corners’, which are the points where the image edge has high curvature. Because they are highly localised, and the image

changes rapidly with a small motion in any direction, these features are ideal for correlation matching. This makes corners excellent for motion estimation since, by identifying a corner's position in the next frame, its image motion can be exactly measured. As a result, this literature review concentrates on corner features; motion estimation from edges is discussed in Chapter 4.

Several methods have been proposed for the identification of corners. The method of choice is usually that developed by Harris [62], but for a survey and comparison of possible techniques, see Schmid et al. [123] or S. M. Smith [135]. Typically hundreds of feature points are identified in each image, and these features are then matched between images. For example, Figure 2.2 shows corner features, marked by the black dots, and their detected motion.

### Feature matching

The extracted features are used to estimate the motion, and in order to do this the inter-frame motion (image displacement) of each feature must be measured. Each feature in the first image must be compared with features in the next image to find the location to which it has moved. Usually, for speed, a search is only made over a small window centred on the earlier location.

The best match is found by comparing a small neighbourhood around the feature point (a few pixels in size) with a similar neighbourhood around each possible match. The cross-correlation is usually used to score matches but there are many other measures of similarity that could be used. P. Smith et al. [134] present a survey and comparison of the main matching methods.

One of the advantages of features is that they are invariant to a wide range of photometric and geometric changes—they change little as the illumination or view-point changes. A corner or an edge will still be detected as a corner or an edge under a range of different viewing conditions (as demonstrated by Schmid et al. [123]).<sup>5</sup> Even with quite large changes, cross-correlation should (in general) still identify the correct match. Techniques have also been developed which enable features to be matched under severe distortion, for example by Pritchett and Zisserman [113].

### Motion estimation

Given the image displacements of each of the feature points, the motion may be estimated. In feature-based methods a dense motion field is not calculated and

---

<sup>5</sup>In contrast, pixel-based techniques using the BCCE assume that the illumination only changing slowly with time, and are so are far less invariant to such changes.

instead the motion means one of two things: either the parametric image motion, or the 3D camera motion, both of which may be calculated directly.

The 2D parametric image motion may easily be calculated from matched points by a simple minimisation of the error between the predicted and actual image location in the next image, for example:

$$\text{Error}(\boldsymbol{\alpha}) = \sum_{\text{all corners } x,y} \left[ (x' - (x + U(x, y, \boldsymbol{\alpha})))^2 + (y' - (y + V(x, y, \boldsymbol{\alpha})))^2 \right] \quad (2.6)$$

where a feature at  $(x, y)$  is matched to  $(x', y')$  in the next frame, and  $U()$  and  $V()$  are the image motions as defined earlier. This error function can be minimised directly to find the parameters  $\boldsymbol{\alpha}$ .

As with the pixel-based methods, the image motion may be used to form an image mosaic, and a number of authors advocate a feature-based rather than a pixel-based approach. These include Pritchett and Zisserman [113] and Cham and Cipolla [35], both of whom also tackle the issue of matching highly dissimilar images. Zoghiani et al. [164] also use feature matching, although they use a more sophisticated model of a corner.

Feature points are commonly used in 3D reconstruction, where they are used to estimate the fundamental matrix [50, 63].<sup>6</sup> This is another form of motion estimate, since it can provide the position of the camera for each image (leaving either a projective or Euclidean ambiguity [96, 111]). Standard techniques for calculating the fundamental matrix include that of Zhang et al. [163], and Torr and Murray [146].

## Robust estimation

It is vitally important in feature-based methods to use *robust estimation*, since a considerable number of the corner matches identified by cross-correlation will be incorrect [134]. Since the motion that is being fitted is either parametric or is otherwise constrained (for example by the fundamental matrix), it becomes relatively easy to detect and remove outliers. Nevertheless, this is still an essential part of the process. A good survey of techniques is provided by Torr and Murray [146].

The most successful approaches are Fischler and Bolles's RANSAC algorithm [54] and Rousseeuw's Least Median of Squares [118]. This latter technique is used for, example, by Zhang et al. in [163]. Various improvements have been made to

---

<sup>6</sup>'3D reconstruction' in this context is the process of making a three-dimensional model of the viewed scene. The fundamental matrix encodes a relative camera positions and also the internal camera parameters, such as the focal length.

these, notably by Torr and Zisserman [149] and Cham and Cipolla [35], both of which develop a probabilistic version of RANSAC.

### 2.2.3 Pixel vs feature-based methods

Feature-based methods have a several advantages over pixel-based methods for motion estimation. By concentrating on only a fraction of the total image area, the computation cost for feature-based methods is far lower. In addition, the areas which are used are those which have a high degree of invariance to change between images, and so more reliable results can be obtained.

A major advantage of feature-based methods is that they lend themselves to statistical techniques and modelling. It is possible to model the noise and typical errors in feature-matching and, with discrete features, statistical independence is usually a valid assumption. As a result, ‘least squares’, which assumes independent Gaussian errors, is a valid approach and techniques such as bundle adjustment [151] and RANSAC may be applied. The validity of these approaches in pixel-based methods is much less certain.

Where pixel-based methods do have advantage is that they produce an immediate dense labelling of the image. Feature-based methods only label the feature points, giving a sparse representation. This may then be used to initialise a dense labelling, but a pixel-based approach is often deemed to be more elegant. For motion segmentation the task is to label each pixel in the frame according to their motion, thus requiring a dense labelling of the image. As a result, pixel-based methods are the most popular in the field of motion segmentation, despite the advantages offered by feature-based approaches.

## 2.3 Motion segmentation

Motion segmentation is the act of labelling pixels in a frame (or frames) from a sequence according to the motion that they obey. There are several ways of achieving this. One is to take a dense motion field (such as is produced by the pixel-based motion estimation techniques described above) and cluster together pixels with similar motions. However, the most successful techniques use a *layered* representation [159, see later in this review], where the pixel motion is constrained to obey one or another parameterised motions. Both of these approaches are described below. As a *segmentation*, many techniques also make use of the image structure to assist

the motion segmentation, either in combination with the motion estimation or as a separate pre- or post-processing stage.

Conceptually, segmentation has been compared to Gestalt grouping. Gestalt theory [89] maintains that visual stimuli appear as grouped entities based on the principles of similarity, proximity, symmetry, continuity and closure. The motion segmentation schemes in the literature generally only recognise the first two of these principles, clustering together neighbouring pixels which share a similar motion or intensity. Much work remains to be done to automatically provide segmentations which are consistent with human perception.<sup>7</sup> In particular, most motion segmentation approaches consider only *local* measures. It will be argued in Chapter 3 that a motion segmentation necessarily also requires *non-local* reasoning.

### 2.3.1 Motion field segmentation

#### Finding surfaces

Early work on motion segmentation concentrated on segmenting a dense motion field. Adiv [1] clustered together pixels which appeared to obey the same planar motion (using the Hough transform [7]). These planar surfaces were then further merged into objects obeying the same 3D motion. Murray and Buxton [105] also fit models to the flow field, using a set of planar facets.

However, this direct use of the motion field is rather naïve, as it employs none of the constraints that are known about the motion or the image. In particular, the motion field is calculated as a first stage and, as with many pixel-based methods, is assumed to be smooth over the image. The motion is only well determined in regions of high image gradient, and only in the direction of the image gradient, which makes the smoothing necessary. This smoothing means that when there are different moving objects present (as would be expected in motion segmentation applications), the discontinuities in the motion field at the object boundaries are also smoothed. As a result, the object boundary can only be approximately identified unless some explicit modelling of the boundary is used at the time the flow field is produced.

#### Modelling discontinuities

Black [15] has published a number of papers addressing the issue of obtaining a good motion field in the presence of motion discontinuities. This naturally involves

---

<sup>7</sup>Although different observers may perceive, or require, a different segmentation. See the recent paper by Martin et al. for a study into the human labelling of static scenes [95].

identifying these discontinuities. His approach [16, 17] combines the standard motion constraint equation with spatial smoothness and ‘temporal coherency’ constraints.<sup>8</sup> The former encourages neighbouring pixels to be similar, the latter tracks patches over a number of images and states that the motion should only change slowly. These are all thrown into a global minimisation scheme. By tracking patches, he identifies areas of occlusion and disocclusion (areas where two patches coincide, or areas where there are no tracked patches). These regions are marked as much more uncertain, and smoothing is not performed across them. This results in a much improved flow field (this was the example used in Figure 2.1), but the precise localisation of the boundary is still not possible.

### Piecewise fitting

A fundamental realization for motion segmentation is that the motion field should contain several disparate motions, and the most successful approaches consider fitting multiple motions, spread over different areas of the image. In [49], Etoh and Shirai initialise an array of different motions across the frame and allow these to ‘learn’ their own local smooth motion field, and the region to which this applies. This process is assisted by also considering the colour of pixels. As shall be seen later, this consideration of image colour or intensity information in addition to the motion field provides valuable assistance to motion segmentation schemes.

Other authors divide the image into small patches and fit motions to these to determine motion boundaries: these occur in patches which are best explained by two motions, rather than one. Jepson and Black [85] consider the motion in  $32 \times 32$  blocks in the image, and robustly fit two motions. They then determine whether the two motions should be merged, giving either one or two motions per block. Bergen et al. [10, 11] also consider the problem of robustly fitting one or two motions to small image blocks. These approaches identify the blocks which contain the object boundaries, but do not perform well at exactly localising the boundary within a block.

This multiple-motion approach may be taken further, considering the whole image as one block, and is the basis for the layered motion representation discussed below. Approaches using this layered representation produce the best motion field estimates in motion segmentation applications.

---

<sup>8</sup>His spatial constraint uses a Markov Random Field [36, 59], discussed later.

### Feature-based motion segmentation

Pixel-based approaches are by far the most popular for motion segmentation, but some authors have advocated a feature-based approach. The classic work in this field is that of Torr [143], who uses the RANSAC algorithm [54] to divide corner matches into clusters obeying different rigid 3D motions. Once these motions are estimated, the pixels may be densely labelled according to the motion that they best fit. S. M. Smith's ASSET-2 system [136] is a real-time implementation of corner finding and matching for motion segmentation. In this system, an estimate of the shape is made by taking the convex hull around the corner features identified with each object.

Both of these approaches produce a good motion estimate and feature labelling, but only attempt a simplistic dense labelling. Labelled corners provide only a sparse representation of the segmentation, and this is insufficient for a complete labelling.

### 2.3.2 Layered motion

Wang and Adelson [158] argue that motion segmentation at the pixel level is too abstract, and that it is necessary to consider the *objects*. In Figure 2.3, extracted from [158], two moving objects are shown, and the frames in the sequence can be composed as an animator would produce them—as the superposition of two separate *layers*, each undergoing a different motion. In their *layered motion representation* these two layer motions are estimated and then the binary *support map* is determined, which is the layer to which each image pixel belongs. This approach was also proposed by Darrell and Pentland [42].

In the layered framework, the motion field is smooth across the pixels in a single layer, but the motion in different layers is independent. This representation inherently allows motion discontinuities to occur between layers. In addition, each layer represents a different object in the sequence, so the assignment of pixels to layers also provides the motion segmentation. This framework forms the basis of many current motion segmentation schemes, and has proven to be very effective.

As well as providing a motion segmentation, layered representations may also be used to generate a good, dense, optic flow field in cases where it is not exactly described by the parametric motion. A smoothed per-pixel flow is calculated as a small offset to the global motion for the layer, as for example in work by Hsu et al. [67], and Black and Jepson [19]. This explicitly models and maintains the discontinuities between layers. Layered approaches have also been adopted for 3D



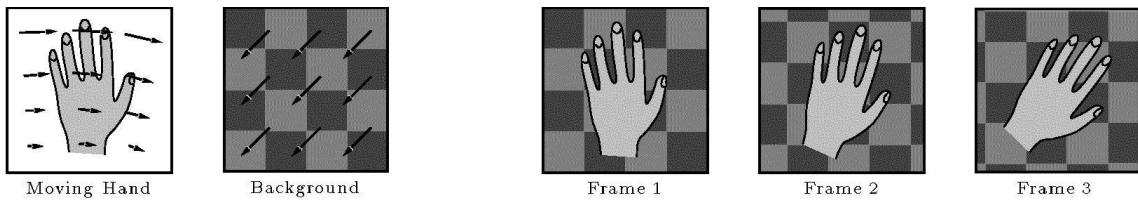


Figure 2.3: *Layered motion example sequence.* (from Wang and Adelson [158].) The hand rotates while the background moves down and to the left.

modelling from stereo images, building a model similar to theatre sets, from a series of 3D planes [6, 147].

### 2.3.3 Layered motion extraction

The majority of motion segmentation literature in recent years make use of the layered representation described above, and there are a number of different approaches to extracting the different layer motions. The extraction of multiple motions is a circular problem—a motion cannot be estimated until a region of support is known, but identifying the region of support relies on the motions being known. There are three main solutions to this problem: motion clustering, the dominant motion approach, and simultaneous motion estimation.

#### Motion clustering

In their original papers on the layered representation, Wang and Adelson [158, 159] divide the image into rectangular regions, and estimate a single affine motion in each region to provide a list of likely layer motions. If any region crosses an object boundary, the motion fitted will exhibit a large residual error, and these motions are eliminated. The candidate motions are then grouped in affine parameter space by  $k$ -means clustering [140] to provide the final layer motions. Darrell and Pentland [42] also form candidate layer motions but then find the subset of their candidate layer motions which ‘best’ describes the complete frame motion using the Minimum Description Length (MDL) principle [116, and see Section 7.2.2].

Once the layer motions have been determined, the regions of support can be identified. The approach proposed in most schemes, and which is simple and effective, is to label each pixel with the motion under which the intensity error is minimised. To label a pixel, it is projected into the next frame according to each layer motion and the image intensity at that new location is compared with the intensity in the original frame. It is then assigned to the layer under which it finds the

closest match. This works well apart from in areas of smooth intensity, where the pixel labelling is ambiguous. These areas are the same ones which caused problems to the pixel-based schemes, and the solution requires (again) smoothing, or further image information. Both of these approaches are discussed later.

### Dominant motion

A very popular approach is the *dominant motion* technique [5, 30, 40, 78, 107, 121, 122]. This scheme calculates one motion at a time, firstly fitting one motion to all the image pixels (the dominant motion). Once this motion has been estimated, the pixels are tested to find the region of support for this motion. These pixels are removed and the remaining pixels (the ‘non-conforming regions’ in the terminology of Odobez, Csurka and Bouthemy [40, 107]) are identified as belonging to independent objects.<sup>9</sup> This dominant motion process can be repeated recursively on the non-conforming pixels to separate out further independent objects if desired.

The two key elements to this approach are the estimation of one motion in the presence of others, and the identification of conforming pixels. In [78], Irani et al. use a hierarchical approach, using the observation that the fitting of a translational motion is robust to other moving objects (Burt et al. [30]). First the dominant 2D translation in the image is calculated and its region of support is identified. A higher-order parametric 2D motion (e.g. affine) is then calculated for this region and the region of support for this new motion is then calculated across the whole image.

Most dominant motion estimation implementations make use of robust methods [146] to fit the motion. Many authors recommend the use of M-estimators [70, and also Appendix A], which reduce the effect of gross outliers in parameter estimation. Examples using M-estimators include Sawhney and Ayer [121], Odobez and Bouthemy [107] and Huang et al. [68]. Other robust estimators may also be used; Ayer et al. [5], and Meier and Ngan [97] use Least Median of Squares [118] as their robust estimator.

Identifying whether a pixel is ‘conforming’ or ‘non-conforming’ is a non-trivial task, since with only one motion proposed, a direct comparison cannot be made between different motion hypotheses. A simple threshold based on an intensity comparison is also not considered to be robust enough [78, page 7]. Irani et al. in [78] calculate a dense motion field in both images and define a ‘motion measure’ and

---

<sup>9</sup>It is commonly assumed that the dominant motion represents the background motion (the motion layer furthest from the camera). This is often the case, but should not be relied upon in a general sequence (see the Car sequence in Chapter 5).

a ‘reliability’ which are calculated for each pixel and used to identify conforming pixels. In [107], Odobez and Bouthemy use a statistical regularisation approach.

The labelling of pixels can be made more reliable by considering several different frames. Giaccone and Jones [60] label their pixels using motion information across three frames using a probabilistic classification. A multiple-frame approach to pixel labelling (‘temporal integration’) is also encouraged by Irani et al. in [78].

The dominant motion approach works very well when most of the frame consists of background pixels. In some applications (e.g. [107]), it is sufficient to estimate one motion and remove these background pixels. However, this technique can perform poorly in cases where there is no one dominant motion—where the foreground objects are large, or there are many motions. Further problems with this approach are discussed in Chapter 4.

### Simultaneous motion estimation

The problems associated with robustly finding one motion at a time, and labelling conforming pixels according to one motion at a time, may be avoided by estimating all the motions simultaneously. Pixels may then be labelled by a direct comparison with all the proposed motions. This is an approach followed by a number of authors [4, 27, 48, 121, 122, 160], all of whom use the Expectation-Maximisation (EM) algorithm [43] to perform this simultaneous estimation. This begins with an initial guess of the motions and then iteratively determines the regions of support (by comparing the pixel intensities under each possible motion), and estimates the motions given these regions of support.

The number of motions present must also be determined initially (rather than recursively as in the dominant motion case). This may be done by fitting too many motions and merging similar motions, as proposed by Weiss and Adelson [160], but the usual approach is to try different numbers of motions and use the Minimum Description Length principle [116] to determine the best number of motions. This is a scheme supported by Ayer and Sawhney in [4, 121], and also adopted by Brady and O’Connor [27] and by Elias [47, 48]. The approach places the motion segmentation problem on a sound statistical footing, providing the *maximum likelihood* segmentation. These approaches work well, although care must be taken to avoid local maxima in the EM process.<sup>10</sup>

<sup>10</sup>In common with many iterative schemes, each step EM takes is in a direction which is locally favourable. This can result in it finding a solution which is locally the best, while not considering a better solution elsewhere. See Chapter 7 for a discussion of this, and a proposed solution.

### 2.3.4 Enforcing spatial coherency

Given a set of motions, the assignment of pixels to layers requires determining which motion they best fit, if any. This can be done by comparing their colour or intensities under the proposed motions, but this presents several problems. Pixels in areas of smooth intensity are ambiguous as they can appear similar under several different motions and so, as with the optic flow techniques discussed earlier, some form of smoothing is required to identify the best motion for these regions. Pixels in areas of high intensity gradient are also troublesome, as slight errors in the motion estimate can mean that a pixel of a very different colour or intensity is observed, even under the correct motion. Again, some smoothing is usually required.

#### Markov Random Fields

A common solution is to use a Markov Random Field (MRF) [36, 59], which encourages pixels to be labelled the same as their neighbours. Weiss and Adelson [160] suggest this as a possible approach, and Bouthemy et al. have produced a number of approaches making use of MRFs to help smooth the motion field [24, 40, 107]. In a paper which preceded the formalising of the layered approach, Murray and Buxton [105] modelled the flow field as a set of planar facets, and pixels were assigned to these planes with the help of a spatiotemporal MRF, which encouraged coherency between neighbours, and consistency across frames.<sup>11</sup>

These schemes can work well, but can often lead to the foreground objects ‘bleeding’ over their edge by a pixel or two if the relative weights of the clustering term and the motion term are imbalanced—if the system is keener to accumulate more pixels than change motion. It is possible to include knowledge of discontinuities into MRFs, but it has already been acknowledged that it is not possible to accurately identify the location of these discontinuities from the motion field alone.<sup>12</sup> In order to produce accurate motion boundaries, additional information is required. This can be provided by the pixels present in the image.

### 2.3.5 Using intensity information

All of the techniques considered so far try to solve the motion segmentation problem using only motion information. This, however, ignores the wealth of *a priori* information that is present in the form of the existing image structure; Weiss and Adelson

<sup>11</sup>Markov Random Fields are also popular in other fields which require the statistical modelling of spatial systems, for example image and video reconstruction [59, 100].

<sup>12</sup>Markov random fields can handle discontinuities by allowing sites to be unconnected to their neighbours, for a certain cost. For examples, see [14] or [105].

[160] argue that there is an excessive reliance on motion data in the field of motion segmentation. Some approaches which make use of further image information are described below.

### Discontinuous Markov Random Fields

Markov Random Fields, as mentioned above, are a popular means of encouraging spatial coherency [24, 40, 107, 160]. However, spatial coherency is *not* required at segmentation boundaries, so it is desirable to modulate the MRF probabilities according to the prior likelihood of there being a discontinuity at that location. This can be achieved by considering the local image colour—a motion discontinuity is more likely where there is also a colour discontinuity (different objects are often different colours). Boykov et al. [25] describe how such an intensity term may be introduced, and demonstrate good boundary localisation in their 3D reconstruction examples.

In [14], Black proposes a scheme using a pixel labelling which combines three terms: motion, image intensity and boundary locations. Spatial coherence is encouraged in motion and intensity via an MRF, but the boundary locations are also iteratively estimated and the smoothness constraints may be violated at these locations. Such approaches are effective, but do add to the computation time. It will also be seen in Chapter 3 that there are circumstances where local measurements alone are not sufficient to correctly determine a labelling.

### Normalized cuts

Shi and Malik’s normalized cuts framework [126, 128] is a general image segmentation scheme. This treats segmentation as a graph partitioning problem, where the image pixels are nodes on a graph and each node is connected to each other node by an ‘edge’.<sup>13</sup> Each edge is weighted according to some measure of similarity between the pixels. Their ‘normalized cut’ is the means of segmenting this graph in such a way that both maximises the similarity *within* groups, and maximises the dissimilarity *between* groups. It relies on finding the eigenvectors of an  $n \times n$  matrix, where  $n$  is the number of nodes (pixels). Even though this matrix is sparse, this approach is clearly computationally expensive, taking about 2 minutes for a  $100 \times 120$  image on a 200MHz PC [128] (although they state that for larger images a multi-resolution approach can reduce the implementation time significantly).<sup>14</sup>

<sup>13</sup>This has nothing to do with the one-dimensional image features also referred to as edges.

<sup>14</sup>In practice, to make their scheme useable, they only form connections to a random selection of pixels in the local neighbourhood, using only 10% of the possible connections.

For motion segmentation, they present a scheme [127] which assigns edge weights according to the similarity of the (pixel-based) motion vector at each pixel. This only considers the motion information, but they point out that a measure of the pixel intensity could also be included. Their results are good, although without the intensity information, some bleeding is observed around the edges, as with the MRF approaches discussed earlier.

Alternative image segmentation techniques using graph cuts are proposed by a number of other authors, including Ishikawa and Jermyn [79] and Boykov and Jolly [26], although their applicability has not yet been demonstrated in motion segmentation scenarios.

### Using edge features

If the moving objects are to be seen at all, they must be a different colour or intensity to their background, and as a result the boundary edge will be visible in the image. Constraining the motion field using these image edges is an obvious approach, but one that has been neglected in the literature. One work, by Meier and Ngan [97] does make explicit use of edges in their final segmentation. They perform a dense labelling of the foreground pixels (which they assume are those pixels which do not obey the dominant motion) but then use this to label edge features as foreground. The pixels interior to the foreground edges are then filled in by simple scanning technique. This approach is limited to segmenting objects from a dominant background, but their use of edges ensures very good results in these cases, with highly accurate motion boundaries. This promising work is yet to be followed up, and it will be seen in this dissertation that edges should play a far more important role in motion segmentation than merely cleaning up a dense labelling.

### Using image regions

The most popular intensity-based approach involves, as with the layer-based approach, taking a step back and working at a higher level than individual pixels. It has been acknowledged at various intervals throughout this review that it is very difficult to determine the motion of pixels in areas of smooth intensity, and that the motion in these areas must invariably be found by extrapolating from nearby features. These smooth areas of the image can be determined prior to any motion analysis by performing an initial segmentation based purely on intensity (or colour) to combine these smooth areas into individual *regions*. This provides an over-segmentation of the image, compared with that desired for a motion segmenta-

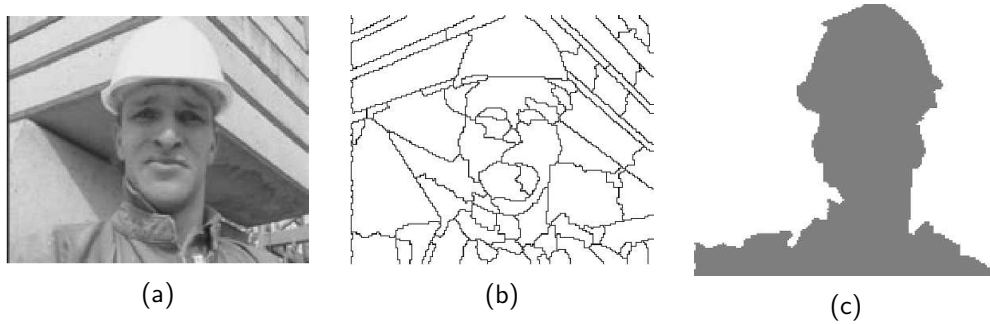


Figure 2.4: *Region merging example.* (a) Frame to be segmented; (b) Regions of similar image intensity; (c) Regions merged according to their motion, giving the final motion segmentation. (From Bergen and Meyer [12].)

tion. The motions of these regions, rather than the pixels, can then be determined and these regions clustered together according to their motions. Similar approaches have also been proposed for 3D reconstruction, for example by Tao and Sawhney [139].

### 2.3.6 The region merging approach

In region merging approaches to motion segmentation, the image is first segmented into regions according to the image structure. Each region is then associated with a motion and this motion is used to merge regions belonging to the same object (for an example, see Figure 2.4). This implicitly resolves the problems identified earlier, which required smoothing of the optic flow field, since the static segmentation process will group together neighbouring pixels of similar intensity and they will automatically be labelled with the same motion. Regions will be delimited by areas of high gradient (edges) in the image and it is at these points that changes in the motion labelling may occur.

The static segmentation scheme used as the initial stage in these algorithms is not of great concern, so long as it provides a reasonable over-segmentation of the image containing regions of similar intensity. A segmentation using the *watershed* algorithm [157] is a popular choice (see [12, 27, 110], and Figure 2.4(b)).<sup>15</sup> It is important, however, that the static segmentation is as accurate as possible since the final motion boundaries will be a subset of the static region boundaries.

<sup>15</sup>The watershed algorithm treats the image as a ‘landscape’, where the height at each pixel is given by the magnitude of the image intensity gradient. Regions are then created by filling this landscape with ‘water’, forming pools which eventually join as they pass over ridges in the landscape—the ‘watersheds’.

As with the per-pixel optic flow methods, the region merging approaches estimate the motion either by motion clustering, or a dominant motion approach, or simultaneous motion estimation.

### Motion clustering

Since each image region consists of a number of pixels, it is usually possible to reliably estimate the parametric motion for a single region from its pixels.<sup>16</sup> Many authors estimate a different motion for each region and then merge regions which have similar motions. One of the first papers to consider a static segmentation, by Thompson [141], followed this approach, although he only calculated a region's motion from the pixels along its edge (since the pixels in the interior of a region are, by definition, similar in intensity and so produce unreliable motion estimates). In rather more recent papers, Dufaux et al. [46] perform region merging by  $k$ -means clustering in motion parameter space. Moscheni et al. have developed a graph-based region merging scheme [101–103] which uses a Modified Kolmogorov-Smirnov test to generate the weighting of the links between regions. Bergen and Meyer [12] consider an exhaustive set of pair-wise region merges, keeping the ones with reasonable residuals.

By estimating the motion in each region independently, the estimated motions can sometimes be inaccurate, particularly in small regions. It is perhaps because of this that these techniques frequently mislabel regions or split the image into too many regions. A more reliable approach is that of Tweed and Calway [153], who estimate an array of motions from rectangular blocks in the image and then use these to label the statically-segmented regions.

### Dominant motion

The dominant motion approach has been used less for region merging situations than for the pixel-based layer estimation described earlier. In [5], Ayer et al. estimate the dominant motion over all pixels and classify the (statically segmented) regions which obey that motion as one object before repeating. Huang et al. [68] follow a similar approach. In [57], Gelgon and Bouthemy perform motion clustering but also calculate the dominant motion in the scene, in order to parameterise and describe the camera motion (by assuming that this is the cause of the dominant motion). As with the pixel-based dominant motion schemes, these work very well when segmenting one

---

<sup>16</sup>A reliable estimate requires that each region is large enough, and has sufficient local structure, to enable the motion to be estimated. This is usually, although not always, the case.



relatively small foreground object from the background, but perform more poorly when all the moving objects are similar sizes.

### Simultaneous estimation

Simultaneous estimation is also sometimes used, with the Expectation-Maximisation (EM) algorithm being the usual choice. Weiss and Adelson [160] suggest this (as well as their MRF approach mentioned earlier). In [27], Brady and O'Connor use the initial segmentation to constrain the EM solution by using an additional 'contextual' step in the iteration. They also use the Minimum Description Length principle [116] to determine the best number of motions. In [110], Patras et al. use an iterative approach similar to EM, alternating the labelling of regions and motion estimation, and also adopt an MRF approach to assist a coherent motion labelling of regions. Apart from the usual local maxima problems, these also work well. All of the region merging schemes produce a good 'cut out' if the regions are correctly labelled.

#### 2.3.7 The depth of objects

The various motion segmentation schemes discussed above provide regions of pixels, or layers, which correspond to objects with different motions. What is not generally considered is the relative depth ordering of these layers, i.e. which is the background and which are foreground objects. If necessary, it is sometimes assumed that the largest region or the dominant motion is the background (for example in [57, 78]). Pixel occlusion *is* commonly considered, but only in terms of a problem which upsets the pixel matching and so requires the use of robust methods.

The layer ordering may be identified by examining this pixel occlusion between frames. Wang and Adelson [158] and Bergen and Meyer [12], identify the occasions when a group of pixels on the edge of a layer are outliers to the layer motion and use these to infer that the layer is being occluded by its neighbour. Tweed and Calway [153] use similar occlusion reasoning around the boundaries of regions as part of an integrated segmentation and ordering scheme. These all perform well.

Depth ordering has recently begun to be considered as an integral part of the segmentation process. Black and Fleet [18] have built a model of the optic flow in the region of occlusion boundaries, and this also allows occluding edges to be detected and the relative ordering to be determined. Gaucher and Medioni [56] study the velocity field to detect motion boundaries and infer the regions and occlusion relationships from these.

The study of occlusion is assisted by considering the motion between several frames in the sequence. Giacomme and Jones [60] use three frames, with a Markov chain [61] to describe the process of pixels being occluded or disoccluded. In the general motion segmentation problem, while good results can be obtained when only using two frames (e.g. [12, 107]), the use of multiple frames, for example by Ayer et al. [5], Irani et al. [78], or Elias [47] also provides much greater robustness.

Occlusion has also been considered in the context of two or more views of a static scenes, for the purposes of forming a 3D model of the scene. Of particular relevance to this dissertation are a few papers considering edges. Paletta et al. [109] labelled edges in an image as one of three types: surface markings; face junctions; or occluding boundaries, by modelling the pixels on either side as two planes and comparing these planes, with good results. The detection of edge *junctions* in images was considered by Malik [92] and Broadhurst and Cipolla [28]. They consider the case where an occluding edge in the image chops across another edge in the image, forming a T-junction in the image. This then allows the foreground object to be identified. A similar approach is described in Section 3.3.5 of this dissertation for the purpose of identifying the relative depths of motion layers.

## 2.4 Summary

Most existing motion segmentation schemes consider the motion at every point in the frame, since the segmentation requires a labelling for every pixel in the frame. However, the per-pixel motion is an underconstrained problem and some smoothing or modelling is required. The most successful approach for motion segmentation is the layered representation, where each layer represents a different moving object. All the pixels on that layer obey the same smooth parametric motion but motion discontinuities can occur at layer boundaries. This is a good model of the image motion, and will be adopted in this dissertation.

The challenge with motion segmentation schemes is not the estimation of the layer motions, but the assignment of pixels to different labels. Schemes which are purely motion-based provide a poor cut-out—their localisation of the object boundary can be in error by several pixels. Schemes which also consider the image intensity, usually in the form of a static segmentation, provide a much more accurate cut-out. This region merging approach is only a recent development and is worthy of more study, as the following chapters will show.

Considering every pixel in the frame is often slow, particularly when combined with iterative labelling schemes such as a Markov Random Field. Feature-based

approaches to parametric motion *estimation* are popular, both for efficiency and robustness, but features are not currently considered for dense motion estimation since corner features only provide a sparse representation.

This dissertation considers a feature-based approach to motion segmentation, providing both an efficient implementation and access to robust statistical methods. It will be shown that edge features, as opposed to corners, *do* provide a representation sufficient to label the frame. By making explicit use of the image edges, the motion boundary can be accurately localised, and a region merging approach is followed to label the other pixels.

The relative depth ordering of motion layers is not often considered in the literature, with the assumption that the dominant layer is the background. This dissertation argues that the layer ordering should be considered as an integral part of the segmentation process, and shows that reasoning between the labelling of edges and regions is necessary for a complete segmentation.



---

# Edge-based motion segmentation

---

### 3.1 Introduction

This chapter presents the theoretical foundations of edges and regions for motion segmentation. The previous chapter presented the current state-of-the-art in motion segmentation, where it was seen that almost all existing techniques estimate the image motion on a per-pixel basis and then cluster together pixels or regions with similar motions. An accurate segmentation requires an accurate identification of the motion boundaries—the edges of the flow field and also the edges in the image; edges are fundamental to an accurate motion segmentation. However, from the review in Chapter 2 it is seen that no existing approach makes particular use of image edges. This chapter develops a novel approach to layered motion segmentation which concentrates solely on the edges in the image.

This thesis of ‘edge-based motion segmentation’ is based upon three assertions, which will be proved in this chapter:

**Good motion information is only available for edges.** Edges are the only features in the image which can be reliably detected and tracked, and it is only edge pixels which can be accurately labelled according to a motion (Section 3.2).

**Edges are *sufficient* for a motion segmentation.** They provide enough information to complete the labelling of the other pixels in the image, up to unsolvable ambiguities (Section 3.3).

Edge and region reasoning is *necessary* for a motion segmentation. The reasoning required for an accurate motion segmentation is non-local and non-symmetric between foreground and background. It is only by considering larger-scale features—edges and regions—that an accurate layered segmentation can be determined (also Section 3.3).

This chapter concludes by drawing the logical reasoning of the earlier sections into a Bayesian framework which allows the most likely segmentation to be deduced in real sequences

## 3.2 Edges for motion estimation

Motion segmentation consists of two parts: motion *estimation* across the image and then a motion *segmentation* of the image. This section explains that edges in a frame are the only source of good motion information and it is only at edges that either of these two stages in the process can be accurately performed.

### 3.2.1 Edges and motion estimation

Edges are image features, and are defined to occur at areas of high image gradient, as shown for example in Figure 3.1. A good motion estimate can only be obtained in these areas of high gradient—it is only in these areas that a small movement in the image gives a change in appearance. Corner features (where the image changes in two directions) are commonly used for motion *estimation*, but they are too sparse for the remaining pixels in the image to be filled in, as is required for a segmentation. Edges are macroscopic features (they have a long extent), and it will be shown in Section 3.3 that they do provide enough information to label the remaining image pixels.

Concentrating motion estimation *only* on these edge pixels yields a number of advantages:

**Invariance** Pixel-based methods assume that the image intensity of each observed point in the world does not change between frames. By contrast, feature-based approaches simply require that the feature is extracted in both frames, and that it maintains a similar appearance. The matching and tracking of features therefore has a wide range of invariance to both photometric and geometric changes, and so these schemes are more widely applicable.

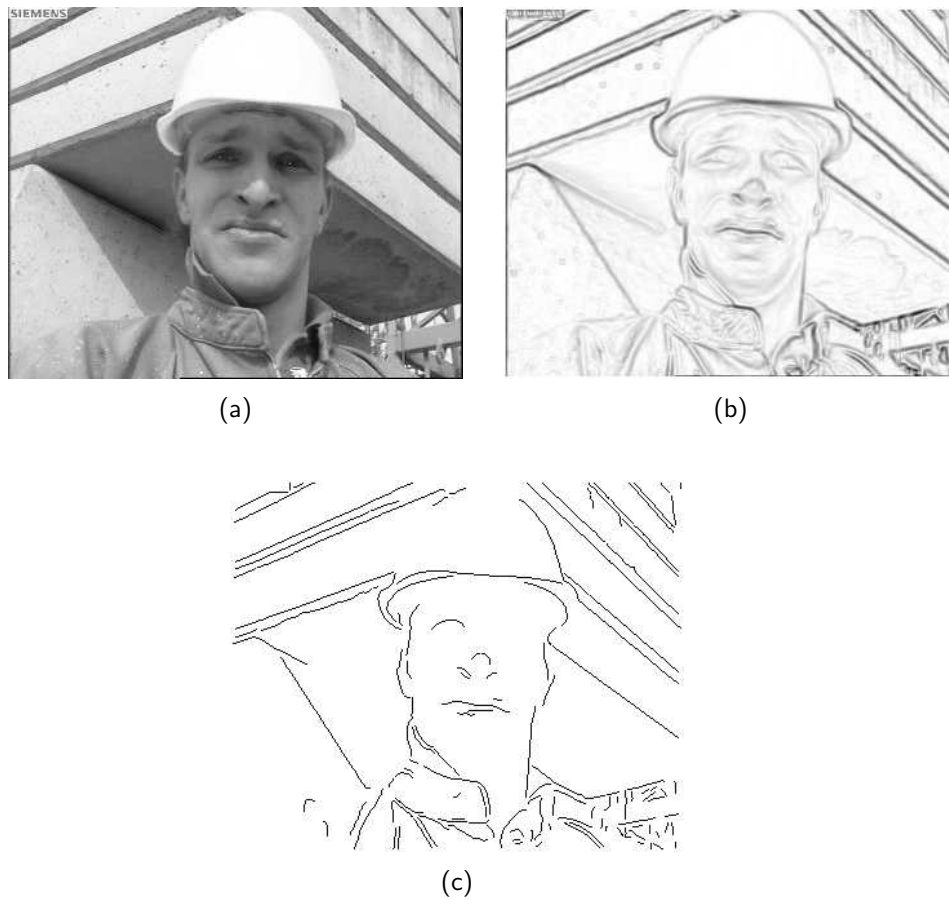


Figure 3.1: *Image intensity and edges in a frame.* (a) A frame from the Foreman sequence; (b) An image of the intensity gradient (using the Sobel gradient operator [137]), where darker areas indicate a high gradient; (c) Edges in the image—connected chains of high image gradient (using Canny [33]).

**Reliable detection** Because of its long extent, an edge detected in one frame is likely to also be present in the next, in full or in part. In contrast, individual corner features are less likely to be detected frame after frame. As a result, edges can be reliably tracked between frames and their motion estimated.

**Robust motion estimation** Each edge in the image can be reasonably assumed to correspond to a 3D contour—part of an object—in the world. In this case, all of the pixels along an edge will obey that object’s motion and, between frames, a similar image motion. This motion can be estimated by combining motion estimates made at a number of places along the edge. This approach is robust to errors which may occur along part of its length. No such clustering of measurements is appropriate in pixel-based methods without some high-level analysis of the scene; the use of edges provides this.

**Statistical models** The edge motion is found by locating a matching edge pixel in the next frame and measuring the image displacement at a number of points along its length. Statistical models can be developed which describe these displacements, and the probability that an edge fits a particular motion. While the pixel-based case may also be modelled, it is complicated greatly by the smoothing used, and the different influences of different pixels. In particular, in the pixel-based case, the smoothing across pixels means that it is inappropriate to assume that the motion detected at each pixel is independent of that at other pixels.

**Computational efficiency** The edge pixels shown in Figure 3.1(c) account for only 3.6% of the total number of pixels in the image. By analysing only these pixels the time taken for the motion estimation process can be significantly less than for a pixel-based approach. Alternatively, more sophisticated techniques (such as non-linear optimisations) may be applied, which would otherwise be prohibitively time-consuming.

These advantages are in addition to the fundamental thesis that edges are necessary and sufficient to the motion segmentation problem.

### 3.2.2 Edges and motion segmentation

A motion segmentation is the act of labelling pixels according to their motion. The motions are first estimated, and then each pixel must be assigned to one of the layers on the basis of the motion that it best fits. Figure 3.2 demonstrates the problems of



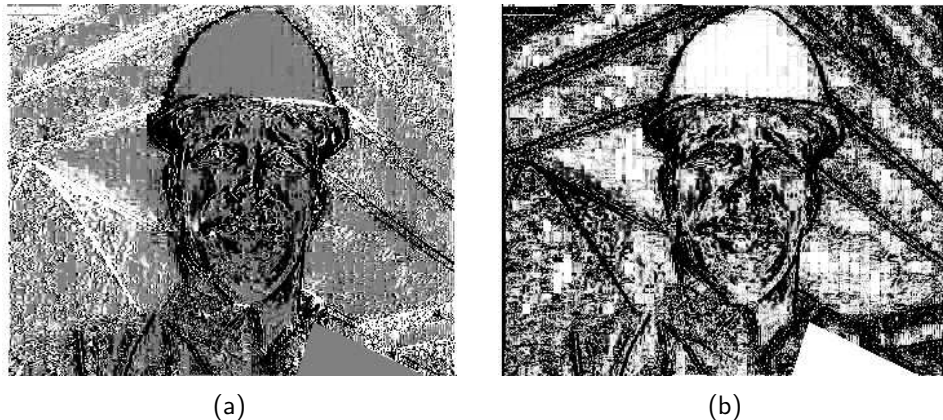


Figure 3.2: *A per-pixel motion labelling.* Given two motions: the head motion and the background motion, each pixel is labelled. (a) Pixels labelled according to their probability of obeying the each motion, where white is the background motion and black pixels are more likely to belong to the head; (b) The confidence of each pixel (the probability of its most likely label), ranging from 0.5 (white) to 1 (black). Compare (b) with Figure 3.1 and it is clear that the pixels with the best information are those near edges.

trying to do this on a per-pixel basis. Here, two motions have been estimated (by the method presented later in this dissertation): one for the head and one for the background. Each pixel in one frame is projected into the next according to each of the motions, and is labelled with its probability of obeying each motion.<sup>1</sup>

Figure 3.2(a) shows the resulting pixel label probabilities, where the whiter a pixel, the higher the probability of obeying the background motion, and darker pixels obey the foreground. The head can be discerned, but it can be seen that there are many grey areas where the pixel labelling is uncertain.<sup>2</sup> This is made clearer in Figure 3.2(b) where now each pixel is labelled according to the probability of it obeying its most likely label i.e.

$$\text{label} = \max(P(\text{head}), P(\text{background})) \quad (3.1)$$

which gives an image of the labelling confidence, from 0.5 to 1 (white to black). It is clear that areas of smooth intensity (such as the hat) are the most uncertain

<sup>1</sup>The probability is calculated by assuming that the pixel colour (red, green and blue components) is unchanged between frames apart from isotropic Gaussian noise of standard deviation 3 (out of a dynamic range of 255). The probability of it being the same pixel in the new location is calculated, for each motion model. These are normalised to give the probability of each motion.

<sup>2</sup>Two sequence-specific artifacts are also visible in both images, and should be ignored. The square blocks visible in the image are the coding artifacts of the MPEG-1 format that has been used to code this motion sequence. The flat expanse at the bottom-right covers the pixels where (to the nearest pixel) both motions are exactly the same.

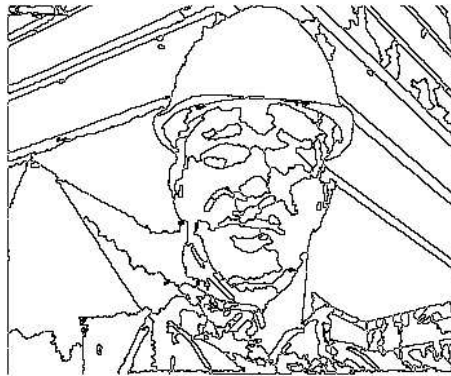


Figure 3.3: *Segmented image regions*. An example of a static segmentation of the Foreman image from Figure 3.1(a). Regions consist of pixels with similar colour, and edges in the image form region boundaries. (Using the method of Sinclair [130, and see Chapter 4]).

areas. A white pixel on the hat moves to another white pixel under either motion, and the same is true in any areas where groups of pixels share a similar colour. If Figure 3.2(b) is compared with Figure 3.1(c), it can be seen that the only areas with high labelling confidence are the edges in the image, areas with significant image structure. Only edge pixels can be confidently labelled according to their motion, and obtaining an accurate labelling in other areas of the image is not possible from motion alone. In order to label these areas, some additional prior knowledge is required.

A common approach to enforce better coherence in pixel-based methods is to apply local smoothing, or use a Markov Random Field (MRF) to encourage uncertain pixels to adopt the labelling of their neighbours. Unfortunately, not all pixels *should* obey the motion of their neighbours—pixels on the boundary of an object should only conform with *some* of their neighbours. In order to correctly enforce this, some non-local reasoning is required to identify these boundaries and the correct labelling (this is discussed further in Section 3.3.4). A better approach is to mark each area of smooth intensity as an individual region and label all the pixels in that region as one. An initial static segmentation, as shown for example in Figure 3.3, divides the image intensity structure into such regions of smooth intensity. Naturally, the edges of these regions are the pixels where the intensity changes rapidly and, again, these are the edges in the image.

The edges in the image are the only areas which can be reliably labelled according to their motion. By dividing the image areas between edges into regions based on colour, a prior clustering of pixels is performed which enables coherent groups of

similar pixels to be labelled as one, but still allows clean discontinuities at potential boundaries, the edges.

### 3.3 Edges and regions for motion segmentation

This section develops the theory linking the labelling of edges and regions; this approach to motion segmentation is the core contribution of this dissertation. It is shown here that, as well as being good sources of motion information, edges are sufficient and necessary for an accurate motion segmentation.

#### 3.3.1 Prior assumptions

There are a number of fundamental assumptions which underly this theory. These are valid in virtually all real sequences, but should be stated here for completeness:

**Edge formation** Edges in an image are generated as a result of the structure of objects. Edges in an image may also be due to material or surface properties (texture or reflectance). It is assumed that edges due to the latter two types do not occur; this is a valid assumption in many sequences, but can cause problems otherwise.<sup>3</sup> The most important edges formed are those which are the occluding boundary (outline) of objects in the image—these are the edges of the object, which demarcate the area to be segmented.

**Edge motion** As an object moves, all of the edges associated with the object move, and hence edges in one frame may be compared with those in the next and partitioned according to different real-world motions.

**Layered motion** It is assumed that the motion in the sequence is layered, i.e. one motion takes place completely in front of another. Typically the layer farthest from the camera is referred to as the background, with foreground layers in front of this.<sup>4</sup>

It is further assumed that the pixel (and thus edge) motion on each motion layer may be reasonably described by a simple parametric motion model.<sup>5</sup>

<sup>3</sup>See the Car sequence in Chapter 5 for an example of surface reflections.

<sup>4</sup>This is almost always the case. However, for example, a person walking behind a lamppost (which is part of the ‘background’) may be considered a violation of this assumption. Another example is shown in the Coastguard sequence considered in Chapter 5, where it is shown that minor violations of this assumption do not greatly affect the solution.

<sup>5</sup>The criterion here is that a parametric motion will describe the motion to within a few pixels. The errors under a parametric motion are investigated both in Chapter 4 and Chapter 6.

### 3.3.2 Conditions for a correct segmentation

There are two further conditions which must be met for an *accurate* segmentation:

**Visible occluding boundary** The occluding boundary of foreground objects must be visible as an edge in the image. If this edge cannot be seen (i.e. both the object and its background are the same colour) then one image region will span both the foreground and background. Without an image edge it is difficult to tell where in this region the correct object boundary is.<sup>6</sup>

**Foreground and background edges must intersect** The edge labelling is only sufficient for a complete region labelling if there is some edge interaction between the two motions. If not, the labelling of some regions will be ambiguous, and the relative depth ordering of the motion layers cannot be determined.

Both of these are an example of an *unsolvable ambiguity*. As will be discussed later, situations which cause these ambiguities are unsolvable under any segmentation scheme. In all other cases, edges are sufficient.

### 3.3.3 Edge labels

It is assumed that the frame to be segmented has already had a *static* segmentation performed (as opposed to a motion segmentation). This divides the frame into regions which consist of adjacent pixels of similar colour (i.e. containing no motion information), and these are bounded by image edges (which provide good motion information). There are many static segmentation schemes in the literature, and Figure 3.3 shows the output of one such scheme, by Sinclair [130] (described in Chapter 4). This section considers the relationship between the motion labelling of edges, and that of regions. First, it considers the labelling of edges from regions.

Each region which represents part of a foreground object moves with that foreground motion.<sup>7</sup> Being foreground, it occludes the background and so all of its edges

---

<sup>6</sup>Where there is no image or motion information, additional information must be used to infer the existence and location of these edges. If an edge is fragmented it may be completed by assuming continuity, and an ‘illusory contour’ formed. These approaches will not be considered here, as they are rarely required, but are a suggested avenue for further research in Chapter 8.

<sup>7</sup>Although phrased in terms of ‘foreground’ and ‘background’, this does not restrict this theory to only two motions. The terms should be read as referring to the *relative* foreground and background—considering the interaction between two motion layers (of perhaps many overall) in an area of a frame. One of these layers will be closer to the camera than the other and is, relatively, the foreground.

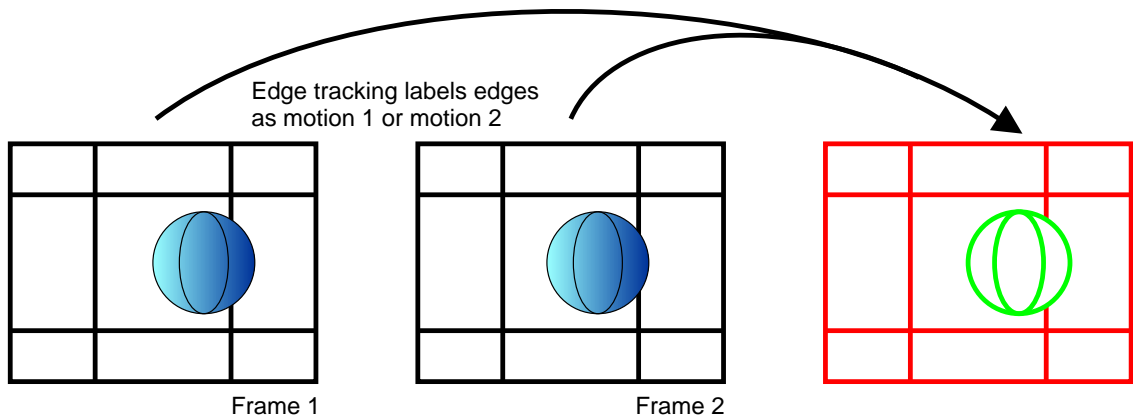


Figure 3.4: *Tracking and labelling edges.* Edges are tracked between frames and labelled as motion 1 (red) or motion 2 (green). All of the edges of the foreground regions (the ball) move with the foreground motion (green). The other edges are where two background edges meet, and they move with the background motion.

are visible and also have the foreground motion. A region which obeys the background motion may be bounded by some of these foreground edges, but if it were entirely surrounded by foreground edges it would be indistinguishable from a foreground region. Where two background regions meet, the edge obeys the background motion.

Consider the example shown in Figure 3.4. Here, the background remains stationary and the ball moves to the left between frames. There are three foreground regions, the segments of the ball, and *all* of their edge obey the foreground motion. The other edges, between background regions, obey the background motion. All regions which are unambiguously background will have at least one background edge.

A motion labelling of regions, and knowledge of the relative depth ordering of layers, therefore completely defines the motion labelling of the edges, according to the following labelling rule:

**Labelling Rule.** *The layer to which an edge belongs is that of the nearer of the two regions which it bounds.*

The next sections consider the reverse, and more useful, process of labelling regions and finding the relative depth ordering from an edge labelling.

### 3.3.4 Region labels

Image regions can be labelled from an edge labelling by considering the implications of the Labelling Rule given above. Any region completely surrounded by edges of

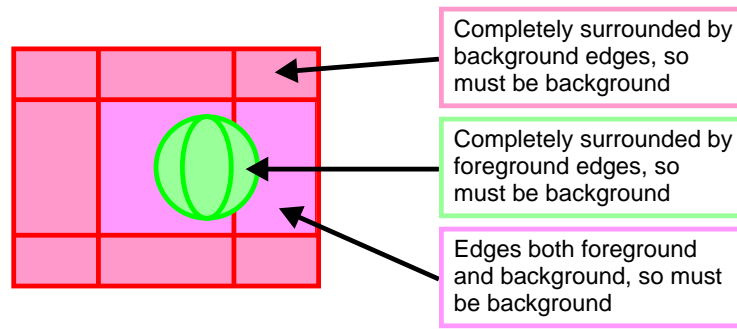


Figure 3.5: *Labelling regions from edges.* Edges which are entirely surrounded by edges of one motion must themselves obey that motion. Regions with edges of both label must belong to the further of the motion layers, the background (red in this case).

one label must themselves obey that motion. The remaining case, regions which are bounded by edges of different motions, must obey the motion of the furthest of those edges. Otherwise, edges on the layer obeying that motion would have been occluded by the region in question.

If the relative depth ordering of the motions is known (i.e. which edges are ‘foreground’ and which are ‘background’), the region labelling is trivial. Consider the edge labelling in Figure 3.4, for which the region labelling process is outlined in Figure 3.5. Only regions entirely surrounded by the edges of the foreground motion—the segments of the ball—can be foreground. All other regions (those entirely surrounded by background edges, or by edges of both labels) must be background.

In the case where the depth ordering is known, an edge labelling is therefore sufficient to perform a complete dense labelling of the image. The next section will show that an edge labelling is in fact sufficient to determine the depth ordering, which is a necessary condition for a complete segmentation of an unknown scene.

Referring back to Figure 3.5, it can be seen that foreground and background edges are not symmetrical:

**Background edges** always separate two background regions.

**Foreground edges** may be the boundary to one foreground region (if it is an occluding edge), or to two foreground regions (if it is an internal edge).

This non-symmetric behaviour—the various interpretations of a foreground edge—are the cause of the non-local reasoning mentioned earlier in this chapter.

Pixels near a foreground edge may perhaps only be foreground on one side of the edge, or on the other side, or pixels on both sides may be foreground. If it is known (for example from edges elsewhere in the frame) that the region on one side of a foreground edge belongs to the background, *then* pixels on the other side of the edge

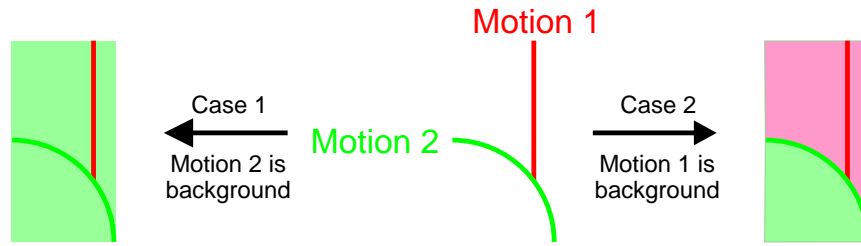


Figure 3.6: *Labelling a T-junction.* Where edges of different motion labellings meet there is only one consistent layer ordering. In Case 1, motion 2 (green) is labelled as background, which means that all regions must be background, which is inconsistent with the red edge. Selecting motion 1 (red) as the background motion (Case 2) gives a consistent solution.

must be labelled as foreground. This pixel labelling cannot be known from purely local reasoning—just because a pixel is near a foreground edge, it does not mean it is foreground. The naïve MRF approach is therefore not appropriate and some higher level reasoning must be applied; edges and regions provide this. Edge and region reasoning are therefore *necessary* to determine the correct region labelling, local measures are not enough.

This asymmetry in the edge labelling enables the relative depth ordering to also be determined from the edges, making them *sufficient* for a complete region labelling and depth ordering.

### 3.3.5 Depth ordering

The depth ordering of layers relies on there being some interaction between edges of different motions. This occurs when foreground regions and edges occlude background edges, leaving T-junctions at which edges with different motion labels meet. Figure 3.6 highlights such a T-junction (extracted from Figure 3.4). It is always the case that, of the three edge fragments meeting at the junction, two have the same motion (this is proved later in this section).

With two possible foreground motions in this case, there are two possible interpretations of the edges in such a T-junction: either the red motion is foreground, or the green motion is foreground. However, in all such T-junctions, one and only one of these possibilities is consistent with the logical reasoning developed above.<sup>8</sup> Figure 3.6 demonstrates the consequences of each of these two possibilities. In Case 1, red is assumed to be foreground and green background and so all regions divided by a green edge must also be background. This implies that all three regions here

<sup>8</sup>T-junctions were also considered by Malik in [92] in the context of stereo images. However, these were explicitly identified in the image by edge matching, rather than being identified from edge labels. In Malik’s case he developed theory enabling the relative depth to be extracted; this is not possible here since there is an ambiguity between the size of the motion of a layer and its depth.

are background, which is inconsistent with a red (foreground) edge being present. Alternatively, Case 2 is where red is background. Here, only the top two regions must be background. The green (foreground) edge is therefore an occluding edge, and the bottom region is foreground. This is completely consistent with the edge labels, and so green must be the foreground motion and this region labelling the correct dense solution.

This theory of T-junctions may be formalised by the following theorem, which draws on the labelling rules developed earlier.

**Theorem.** *No junction may have a single foreground edge. At edge junctions where two different layers meet, two of the edges must belong to the foreground motion.*

*Proof.* If one edge at a junction obeys the foreground motion then one of the regions that it bounds must have the foreground motion. A foreground region has all of its edges labelled as foreground. Each region at the junction is bounded by two of the junction's edge segments. The foreground region must therefore have two edge segments meeting at the junction, and both of these must be foreground.  $\square$

This theorem is sufficient to deduce the layer ordering. Of the three edges at a T-junction, the motion which appears twice is the foreground motion. By this form of reasoning, or by the hypothesise-and-test approach initially described, the edge labels therefore completely determine both the depth ordering, and consequently the complete dense region labelling of a frame.

## 3.4 Unsolvable ambiguities

Section 3.3.2 stated two conditions for a correct segmentation: the occluding boundary must be visible, and edges from different layers must intersect (i.e. there must be T-junctions). This section investigates what happens if either of these conditions are not met.

### 3.4.1 Missing occluding boundary

When part of the foreground object is the same colour as the background, no occluding edge will be visible for that part of the object. This will therefore not be included in any initial region segmentation of the image, and any region which should have belonged to that part of the object will be subsumed into the neighbouring background region. Figure 3.7 illustrates such a case, where the left-hand edge of the ball is no longer present. In these cases this combined region will, by



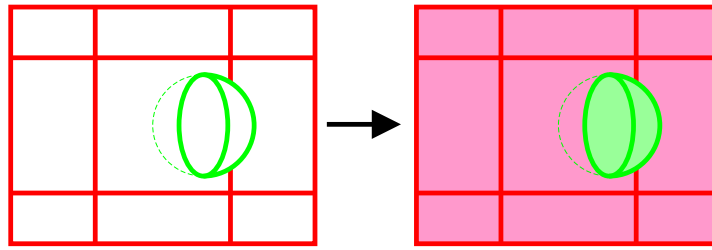


Figure 3.7: *Unsolvability ambiguity: Missing occluding boundary.* If part of the occluding boundary is missing (shown by the dashed line here), part of the foreground will become merged with the background.

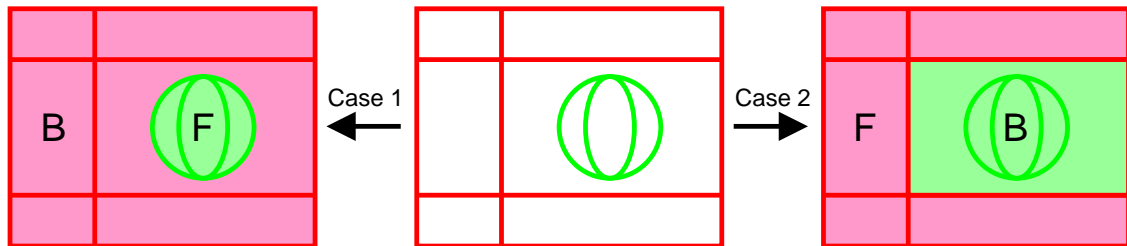


Figure 3.8: *Unsolvability ambiguity: No T-junction.* If there is no interaction between the edges of the two objects, there are two possible interpretations of the edge labelling. Either of the two motions could be foreground, resulting in slightly different region labelling solutions. In case 1, the ball is the foreground object (F); in case 2 the green edges are on the background (B), viewed through a rectangular window in the red foreground.

virtue of having both background and foreground edges, be labelled as background, leaving a reduced foreground object.

Unless this edge is visible (i.e. there is a difference between the foreground and background pixels at this point), it is not possible to distinguish this part of the foreground from the background under any motion segmentation scheme, either edge- or pixel-based. In these cases the object could only be segmented with some higher-level knowledge, such as a model of the expected shape of the object.

### 3.4.2 No T-junctions

Where there is no interaction between edges of different motions in the image (no T-junctions), the layer ordering is ambiguous. Figure 3.8 shows a case where the background edge occluded by the ball is no longer present and, with no T-junctions, there are two possible interpretations. One interpretation is that the ball is the foreground object, moving in front of a featureless background (Case 1). Alternatively, the disc is simply part of the texture on a larger moving object visible through a rectangular window in the foreground (Case 2). Both of these interpretations are completely consistent with the edge labelling.

Situations such as this, where there is genuinely no image structure occluded by the foreground object, are ambiguous under *any* motion segmentation scheme. The system presented here can identify the lack of T-junctions and acknowledge that there is an ambiguity.

### 3.5 Bayesian formulation

The previous sections determined that a segmentation using edges and regions is necessary and sufficient for a dense motion segmentation of a sequence. Edges are good features for motion estimation for a number of compelling reasons, including the opportunity to perform rigorous statistical analysis of their motion labelling. This section develops the statistical framework which enables a dense labelling to be performed from labelled edges.

In a real sequence a complete and self-consistent edge labelling cannot usually be determined, due to noise, and to objects and motions which do not conform fully to the assumptions outlined earlier in this chapter. In this case, each edge can be labelled with a *probability* of their obeying each motion, but not a definite labelling.<sup>9</sup> Given these probabilities, a ‘best’ segmentation must be determined, and this section uses Bayesian methods [58, 84] to find the solution with the maximum *a posteriori* (MAP) probability. This not only considers how well an interpretation explains the measured data, but also allows prior expectations of a sensible solution to be incorporated.

#### 3.5.1 Parameters and maximum likelihood solution

There are a large number of parameters which must be solved to give a complete motion segmentation: the labelling for each image region and the ordering of the different layers. Given that the task is one of labelling the regions of a static segmentation, finding their motion, and determining the layer ordering, the complete model of the segmentation  $\mathbf{M}$  consists of the elements  $\mathbf{M} = \{\mathbf{\Theta}, \mathbf{F}, \mathbf{R}\}$ , where

$\mathbf{\Theta}$  is the parameters of the motion models,

$\mathbf{F}$  is the foreground-background ordering of the motion layers,

$\mathbf{R}$  is the motion label (layer) for each region.

---

<sup>9</sup>Chapter 4 describes how these probabilities may be estimated.

The region edge labels are not an independent part of the model, as they are completely determined by  $\mathbf{R}$  and  $\mathbf{F}$ , as defined by the Labelling Rule of Section 3.3.3.

Given the image data  $\mathbf{D}$  (and any other prior information assumed about the world), the task is to find the model  $\mathbf{M}$  with the maximum probability given this data:

$$\arg \max_{\mathbf{M}} P(\mathbf{M}|\mathbf{D}) = \arg \max_{\mathbf{RF}\Theta} P(\mathbf{RF}\Theta|\mathbf{D}) \quad (3.2)$$

where both  $P(\mathbf{M}|\mathbf{D})$  and  $P(\mathbf{RF}\Theta|\mathbf{D})$  are the probability of the model given the data. This can be further decomposed, without any loss of generality, into a motion estimation component and region labelling:

$$\arg \max_{\mathbf{RF}\Theta} P(\mathbf{RF}\Theta|\mathbf{D}) = \arg \max_{\mathbf{RF}\Theta} P(\Theta|\mathbf{D}) P(\mathbf{RF}|\Theta\mathbf{D}) \quad (3.3)$$

At this stage a simplification is made: it is assumed that the motion parameters  $\Theta$  can be maximised independently of the others, i.e. the correct motions can be estimated without knowing the region labelling (just from the edges). This relies on the richness of edges available in a typical frame, and the redundancy this provides. Usually, there is only one set of motions which can be found to fit these edges, and estimating these motions independently of the region labelling will approach the global maximum.<sup>10</sup> If desired, a global optimisation may be performed once an initial set of motions and region labelling has been found, and this is discussed in Chapter 7. Given this simplifying assumption, the expression to be maximised is

$$\underbrace{\arg \max_{\Theta} P(\Theta|\mathbf{D})}_a \underbrace{\arg \max_{\mathbf{RF}} P(\mathbf{RF}|\Theta\mathbf{D})}_b \quad (3.4)$$

where the value of  $\Theta$  used in term (b) is that which maximises term (a). The two components of (3.4) can be evaluated in turn: first (a), the motions, and then (b), the region labelling and layer ordering.

### 3.5.2 Estimating the motions $\Theta$

The first term in (3.4) estimates the motions between frames ( $\Theta$  encapsulates all the motions). Thus far this statistical formulation has not specified how the most likely motion is estimated, and neither are edges included. As explained earlier in this chapter, edges are robust features to track—they provide the only good motion

<sup>10</sup>Pathological cases are, of course, possible where there is not one obvious set of motions. For example, a sequence with predominantly horizontal edges where all motions are also horizontal.

information—and they provide a natural link to the final image segmentation, being fundamental to the segmentation process.

The edges must be introduced into the statistical model, where they are expressed by the random variable  $\mathbf{e}$  which gives the labelling of an edge—the motion that each edge obeys. This is a necessary variable, since in order to estimate the motion models from the edges it must be known which edges belong to which motion. However, simultaneously labelling the edges and fitting motions is a circular problem: the edge labelling is needed to estimate the motions, while a motion estimate is required to label the edges. One method of resolving this is by expressing it in terms of the classic Expectation-Maximisation (EM) algorithm [43], which iteratively estimates the motions, refining the current estimate  $\Theta_n$ :

$$\begin{cases} P(\mathbf{e}|\Theta_n\mathbf{D}) & \text{E-stage} \\ \arg \max_{\Theta_{n+1}} \sum_{\mathbf{e}} \log P(\mathbf{e}\mathbf{D}|\Theta_{n+1}) P(\mathbf{e}|\Theta_n\mathbf{D}) & \text{M-stage} \end{cases} \quad (3.5)$$

Starting with an initial guess of the motions, the expected edge labelling is estimated (the E-stage). This edge labelling can then be used to maximise the estimate of the motions (the M-stage), and the process iterates until convergence. The EM algorithm is described in more detail in Chapter 4, which describes an implementation of this framework.<sup>11</sup>

### 3.5.3 Estimating the labellings $\mathbf{R}$ and $\mathbf{F}$

Having obtained the most likely motions, the remaining parameters of the model  $\mathbf{M}$  can be maximised. These are the region labelling  $\mathbf{R}$  and the layer ordering  $\mathbf{F}$ , which provide the final segmentation. Once again, the edge labels are used as an intermediate step. Given the motions  $\Theta$ , the edge label probabilities are estimated, and from Section 3.3 the relationship between edges and regions is known. Term (3.4b) is augmented by the edge labelling  $\mathbf{e}$ , which must then be marginalised,

---

<sup>11</sup>As with all iterative schemes, the problem of local maxima is a concern with EM. However, the results in this dissertation show that under two motions this does not present a problem, and Chapter 7 describes a new initialisation scheme which ameliorates the problem for a greater number of motions.

giving<sup>12</sup>

$$\max_{\mathbf{R}, \mathbf{F}} P(\mathbf{R}, \mathbf{F} | \Theta, \mathbf{D}) = \max_{\mathbf{R}, \mathbf{F}} \sum_{\mathbf{e}} P(\mathbf{R}, \mathbf{F} | \mathbf{e}, \Theta, \mathbf{D}) P(\mathbf{e} | \Theta, \mathbf{D}) \quad (3.6)$$

$$= \max_{\mathbf{R}, \mathbf{F}} \sum_{\mathbf{e}} P(\mathbf{R}, \mathbf{F} | \mathbf{e}) P(\mathbf{e} | \Theta, \mathbf{D}) \quad (3.7)$$

where the first expression in (3.6) can be simplified since  $\mathbf{e}$  encapsulates all of the information from  $\Theta$  and  $\mathbf{D}$  that is relevant to determining the final segmentation  $\mathbf{R}$  and  $\mathbf{F}$ , as shown in earlier in this chapter.

The second term, the edge probabilities, can be extracted directly from the motion estimation stage—it is the result of the E-stage of the EM algorithm, (3.5). The first term in (3.7) is more difficult to estimate, and it is easier to recast this using Bayes' Rule [58, 84], giving

$$P(\mathbf{R}, \mathbf{F} | \mathbf{e}) = \frac{P(\mathbf{e} | \mathbf{R}, \mathbf{F}) P(\mathbf{R}, \mathbf{F})}{P(\mathbf{e})} \quad (3.8)$$

This decomposes the probability of the region labelling and depth ordering, given the edge labels, into the probability of the edges given  $\mathbf{R}$  and  $\mathbf{F}$ , and two prior probabilities. The prior probability of an edge labelling,  $P(\mathbf{e})$ , does not change over the maximisation (3.7), which is only over  $\mathbf{R}$  and  $\mathbf{F}$ . The joint prior of  $\mathbf{R}$  and  $\mathbf{F}$  may be separated (i.e. they are independent) since whether a particular layer is called 'motion 1' or 'motion 2' does not change its labelling. This leaves

$$P(\mathbf{R}, \mathbf{F} | \mathbf{e}) \propto P(\mathbf{e} | \mathbf{R}, \mathbf{F}) P(\mathbf{R}) P(\mathbf{F}) \quad (3.9)$$

Any foreground motion is equally likely, so  $P(\mathbf{F})$  is constant, but the middle term,  $P(\mathbf{R})$ , is not constant since some configurations of region labels *are* more likely than others (for example, regions belonging to one object are all expected to be adjacent to each other). This term must therefore be kept, and is used to encode likely labelling configurations. Substituting back into (3.7), this leaves the following

---

<sup>12</sup>The term  $P(\mathbf{e} | \Theta, \mathbf{D})$  in (3.6) could be expressed in terms of marginalising the joint distribution over  $\mathbf{R}$ ,  $\mathbf{F}$  and  $\mathbf{e}$ , i.e.

$$P(\mathbf{e} | \Theta, \mathbf{D}) = \sum_{\mathbf{R}} \sum_{\mathbf{F}} P(\mathbf{e}, \mathbf{R}, \mathbf{F} | \Theta, \mathbf{D})$$

However, this is not necessary as it is simpler to evaluate  $P(\mathbf{e} | \Theta, \mathbf{D})$  directly (see Chapter 4). Indeed, the decoupling of  $\{\mathbf{R}, \mathbf{F}\}$  and  $\{\Theta, \mathbf{D}\}$  by means of the intermediate variable  $\mathbf{e}$ , as expressed by (3.7), is the crucial stage which makes this problem tractable. The edges are this intermediary, and are fundamental to the solution.

expression to be evaluated:

$$\max_{\mathbf{R}\mathbf{F}} \sum_{\mathbf{e}} P(\mathbf{e}|\mathbf{R}\mathbf{F}) P(\mathbf{R}) P(\mathbf{e}|\Theta\mathbf{D}) \quad (3.10)$$

The  $P(\mathbf{e}|\mathbf{R}\mathbf{F})$  term is very useful. The edge labelling  $\mathbf{e}$  is only an intermediate variable, and is entirely defined by the region labelling  $\mathbf{R}$  and the foreground motion  $\mathbf{F}$  (via the Labelling Rule of Section 3.3.3). This probability therefore takes on a binary value—it is 1 if that edge labelling is consistent with the  $\mathbf{R}$  and  $\mathbf{F}$ , and 0 if it is not. The sum in (3.10) can thus be removed, and the  $\mathbf{e}$  in the final term replaced by the function  $\mathbf{e}(\mathbf{R}, \mathbf{F})$  which provides the correct edge labels for given values of  $\mathbf{R}$  and  $\mathbf{F}$ :

$$\max_{\mathbf{R}\mathbf{F}} \underbrace{P(\mathbf{e}(\mathbf{R}, \mathbf{F})|\Theta\mathbf{D})}_a \underbrace{P(\mathbf{R})}_b \quad (3.11)$$

The variable  $\mathbf{F}$  takes only a small, discrete set of values (for example, in the case of two layers, only two: either one motion is foreground, or the other). Equation (3.11) can therefore be maximised in two stages:  $\mathbf{F}$  can be fixed at one value and the expression maximised over  $\mathbf{R}$ , and the process then repeated with other values of  $\mathbf{F}$  and the global maximum taken. This is the same hypothesise-and-test process as outlined in Figure 3.6.

The maximisation over  $\mathbf{R}$  can be performed by hypothesising a complete region labelling and then testing the evidence (3.11a)—determining the implied edge labels and then calculating the probability of this edge labelling given the motions. Then this is combined with the prior (3.11b), calculating the likelihood of that particular labelling configuration. This should be attempted for each possible set of region labels, but an exhaustive search is impractical. In the implementation presented in Chapter 4, a search of likely region labellings is made using simulated annealing [59, 88].

### 3.6 Summary

Edges are fundamental to the process of motion segmentation. They are the only areas of the image which provide good motion information, and they also allow robust statistical techniques to be used, and an efficient implementation. A motion labelling of edges is sufficient for an entire labelling of the image, using regions from a static segmentation. The edges can be used both to label the image regions and determine relative depth ordering of the motion layers, up to unsolvable ambiguities.

Edge and region reasoning is also necessary for an accurate segmentation since some motion labelling decisions are non-local and non-symmetric.

A Bayesian formulation is presented within which the segmentation of a frame may be performed. The motions are first estimated and a probabilistic labelling of edges made. Region labellings and possible depth orderings are then hypothesised and tested against the edge label probabilities in order to find the most likely segmentation. The following chapters present implementations and evaluations of this framework.





# Implementation for two motions, two frames

---

## 4.1 Overview

In the previous chapter it was established that edges should be used for motion segmentation. The relationship between regions and edges was determined, and a Bayesian framework for edge-based motion segmentation was presented. This chapter presents an implementation of this framework for the case where there are two motions present (the background and one foreground object). This is a common case, and also the simplest motion segmentation scenario. Image data are used from the frame to be segmented, and one further frame.

The system progresses in two clear stages, as demonstrated in Figure 4.1. The first is to detect edges, find motions, and label the edges according to their probability of obeying each of the two motions (Figure 4.1(b)). These edge labels are sufficient to label the rest of the image.

In the second stage the frame is divided into regions of similar colour using these edges. The motion labels for these regions which best agree with the edge labelling is then determined according to the framework of Chapter 3. Table 4.1 gives an outline of this implementation.

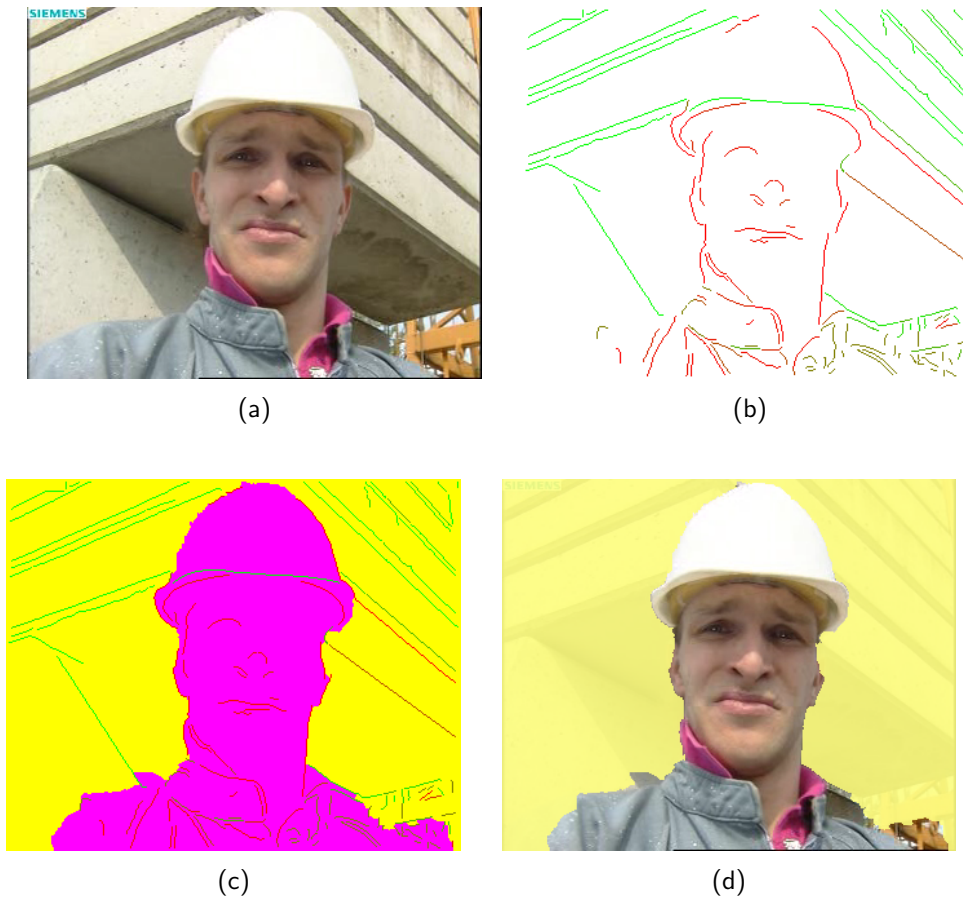


Figure 4.1: *Foreman segmentation from two frames.* (a) Frame 1; (b) Edges labelled by their motion. The foreman moves his head very slightly to the left between frames; (c) Maximum *a posteriori* region labelling; (d) Final foreground segmentation. (See also Table 4.1.)

- |  |
|--|
| <p>(a) <b>Find edges in image</b></p> <p>(b) <b>Find motions and label edges</b><br/>         Initialise motions <math>\Theta_0</math><br/>         Repeat (EM Loop)</p> <ul style="list-style-type: none"> <li>• Calculate edge probabilities <math>P(e \Theta_n D)</math></li> <li>• Estimate the motions <math>\Theta_n</math></li> </ul> <p>until convergence</p> <p>(c) <b>Find the best region labelling and layer ordering</b><br/>         Find regions<br/>         For each possible layer ordering:</p> <ul style="list-style-type: none"> <li>• Initialise region labelling</li> <li>• Refine by simulated annealing<br/> <i>Repeat</i> <ul style="list-style-type: none"> <li>– Try relabelling individual regions</li> <li>– Keep new labelling if total probability is greater</li> </ul> <i>until convergence</i></li> </ul> <p>Select most likely segmentation over all layer orderings</p> <p>(d) <b>Output final segmentation</b></p> |
|--|

Table 4.1: *System overview.* Summary of the edge-based motion segmentation scheme (see also Figure 4.1). As defined in Section 3.5,  $\Theta_n$  represents the set of motion parameters and  $P(e|\Theta_n D)$  the probabilities of the edge obeying each motion, given the motion and image data.

## 4.2 Finding edges

Edge detection is a subject which has received much study, due to the large number of vision applications which use edges and lines as primitives on the way to higher level goals. Most edge detection methods either find maxima in the first image derivative (introduced by Canny [33]), or find zero-crossings in the Laplacian of a Gaussian of the image, as proposed by Marr and Hildreth [93]. Convolution masks may also be used to evaluate the local image gradient, such as the Sobel filter [137]. Bouthemy [23] also used convolution masks, to determine *spatiotemporal* edges—determining both the location and motion of edges.<sup>1</sup> However, a combined approach

<sup>1</sup>A series of video frames may be stacked together to make a 3D volume  $(x, y, t)$ , i.e. two spatial axes and one temporal axis. An edge visible across all the frames (moving or otherwise) forms

such as this, while elegant, requires compromises to be made (in the case of [23], only straight edges undergoing translational motion are modelled). Separating the edge detection and motion estimation stages allows more sophisticated techniques to be used at each stage.

As the standard edge detector in use today, the Canny edge detector is used in the system presented here. Edge detection is performed in a grey-scale version of the input image. Other edge detection schemes would be equally applicable, and a colour edge detector (for example that used by Sinclair in [130]) would be a useful addition to the implementation presented here.

The Canny edge detector begins by applying Gaussian smoothing to the image (here using  $\sigma = 1$ ), and then computes image gradients. Non-maximum suppression is applied to pick out only the ridges in the gradient image and then hysteresis thresholding used to remove weak edges. Hysteresis thresholding uses two thresholds—edgels must be above the lower threshold in order to be considered, and connected chains of these edgels are detected, but a chain is only accepted if at some point along this chain there is an edgel which is above the higher threshold.<sup>2</sup> This reduces the fragmentation in the output edges that occurs if a single threshold is used.

In the implementation used here, it is important that each edge obeys the same motion along its length. Where there is a sudden change in edge direction, this might indicate a structural change, and a possible change in motion. To allow these different parts of the edge to be labelled differently, the candidate chains of edgels are split at points where the direction of the edge gradient changes too rapidly. Each of these split, thresholded chains is, in this implementation, an ‘edge’.

Various parameters must be set to achieve the most useful set of edges. The edge detector should extract as much as possible of the foreground object’s occluding boundary, and the edges due to structure. Edges due to texture and lighting effects (shadows and surface reflections) are either difficult to track or do not obey the motion assumptions and so are undesirable. These edges are usually less distinct than structural edges, and as a result, conservative thresholds are used which allow only the strong edges to be detected. The thresholds shown in Table 4.2 are found, empirically, to be suitable in almost all cases. Figure 4.1(b) shows a typical set of edges extracted using these parameters.

---

a surface in this space. Detecting this surface allows both the edge’s location and motion to be determined.

<sup>2</sup>Edgels are ‘edge elements’ i.e. pixels on an edge

Parameter	Value
Smoothing $\sigma$	1
Upper hysteresis threshold	30
Lower hysteresis threshold	10
Maximum direction change	20°

Table 4.2: *Parameters used for Canny edge detection.* Values based on an 8-bit greyscale image. Conservative values are used to avoid edges due to texture and lighting effects.

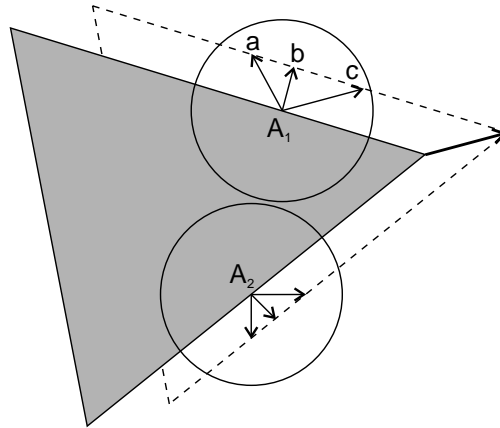


Figure 4.2: *The aperture problem.* It is impossible from local measurements (e.g. within the circle), to tell where either edge point  $A_1$  or  $A_2$  moves to. Point  $c$  is the correct match for  $A_1$ , but all that can be determined from edge measurements is the component of motion normal to the edge.

## 4.3 Estimating motions from edges

### 4.3.1 The aperture problem

Edges are perceived to provide a poor solution to the motion-estimation problem because of the *aperture problem* [94], demonstrated in Figure 4.2. Here the object moves up and to the right and an attempt is made to match an edgel  $A_1$  to its new location by seeking locally (within the circle). The correct match is at position  $c$ , but since an edge is only a one dimensional feature, the edgel could equally well find a match at positions  $a$ ,  $b$  or  $c$ . What all these points have in common is that they are the same perpendicular distance from the edge, so although the exact motion cannot be determined from an edgel, it *is* possible to determine the component of the motion normal to the edge. As pointed out by Buxton et al. [31], this is sufficient to determine a parametric motion. In this case, for a layered motion segmentation, this is all that is required.

The fact that the motion can be determined from only the normal component gives edges an advantage over corner features. Feature-based motion estimation

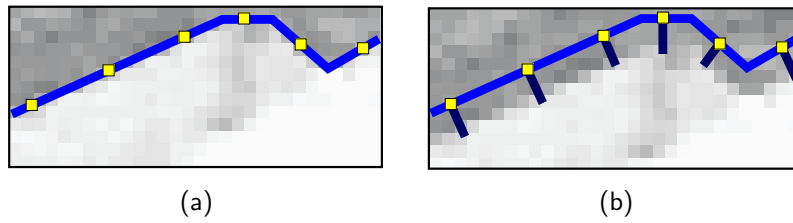


Figure 4.3: *Edge tracking example.* (a) Edge in initial frame, with sample points. (b) In the next frame, where the image edge has moved, a search is made along the edge normal from each sample point to find the new location. The best-fit motion is the one that minimises the squared distance error between the sample points and the image edge.

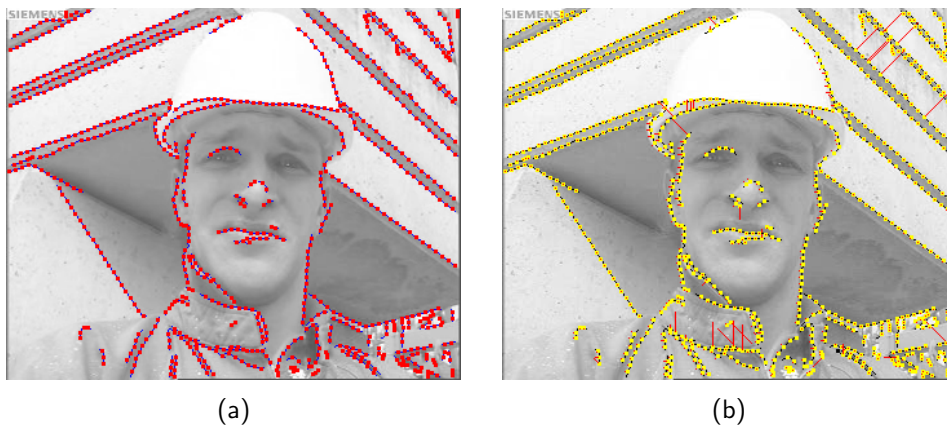


Figure 4.4: *Sample points in a frame.* (a) Sample points initialised every 5 pixels along the edges from Figure 4.1(b); (b) Matches found from by searching normal to each edge, showing each sample point's displacement in red. The background motion is approximately zero, while the head moves a few pixels to the left (see the left-hand brim of the hat). A few mismatches are also observed.

provides a speed advantage over pixel-based approaches, but in order to avoid the aperture problem many researchers turn to two-dimensional image features, ‘corners’, whose matches can be exactly determined. Using corners, however, slows the system again since, in order to find the exact match, a search is usually required over all the possible locations within a search window. Using edges *takes advantage* of the aperture problem by accepting that an exact match cannot be found, but that *any* point on the edge is an acceptable match if the minimisation then just uses the perpendicular distance. This means that only a one-dimensional search is needed to find the new edge location, which is much faster.

### 4.3.2 Finding a match

In order to reduce the calculation cost further, sample points are assigned at regular intervals along the edge, as in Figure 4.3(a), and the motion of these sample points is considered to be representative of the whole edge motion. In this implementation, sample points are placed one every five pixels along an edge, as shown in the example in Figure 4.4(a). This density of sample points is found to give good tracking performance while, clearly, being five times faster than tracking every edge point. Figure 4.4(a) shows 804 sample points, which is a typical number.

From each sample point a search is made to determine the motion of the edge (Figures 4.3(b) and 4.4(b)). As discussed above, the only measurement that is relevant is the motion normal to the edge. This may be determined by only searching normal to the edge but, as shown in Figure 4.2, one could equally well search in any reasonable direction and then project onto the edge normal. For speed and simplicity, therefore, the search is made along the closest compass direction to the normal i.e. in both directions along one of the vectors  $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ ,  $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$  or  $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ , as demonstrated for edge  $A_2$  in Figure 4.2. The search length  $\rho$  is chosen to be at the upper end of the observed inter-frame motion. In the sequences tested, this rarely exceeds 10 pixels, so a generous value of  $\rho = 20$  pixels is used.

To find a match, the colour image gradient at the original location is compared with that at each of the proposed new locations along the search track.<sup>3</sup> Comparisons are only made at integer locations in the pixel grid, giving a match to the nearest pixel; no sub-pixel interpolation has been found to be necessary. The gradient is evaluated independently in the red, green and blue components of the image using a  $5 \times 5$  convolution kernel calculated for this work:

$$\left\{ \begin{array}{l} \begin{bmatrix} -0.7358 & -0.5353 & 0.0000 & 0.5353 & 0.7358 \\ -1.0705 & -0.7788 & 0.0000 & 0.7788 & 1.0705 \\ -1.2131 & -0.8825 & 0.0000 & 0.8825 & 1.1231 \\ -1.0705 & -0.7788 & 0.0000 & 0.7788 & 1.0705 \\ -0.7358 & -0.5353 & 0.0000 & 0.5353 & 0.7358 \end{bmatrix} & x \text{ direction} \\ \begin{bmatrix} -0.7358 & -1.0705 & -1.2131 & -1.0705 & -0.7358 \\ -0.5353 & -0.7788 & -0.8825 & -0.7788 & -0.5353 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.5353 & 0.7788 & 0.8825 & 0.7788 & 0.5353 \\ 0.7358 & 1.0705 & 1.2131 & 1.0705 & 0.7358 \end{bmatrix} & y \text{ direction} \end{array} \right.$$

which as well as taking differences in the  $x$  or  $y$  direction, also provides some (truncated) Gaussian smoothing, using  $\sigma = 2$ . Smaller convolution kernels, such as that proposed by Sobel [137], were found to suffer from noise in some sequences.

The gradients are calculated at both the original location in the first image and

---

<sup>3</sup>A correlation of image intensities over a small window would also be appropriate, although not as invariant to changes in illumination.

the proposed location in the second. The match score  $S$  is taken to be the sum of the squared differences between these gradients, summed over the three colours, and over both the  $x$  and  $y$  directions. If a match is found above a threshold ( $S_{\max} = 100,000$  using the given kernels and 8-bit colour values), then the normal distance for this sample point,  $d^k$ , is taken to be the dot product between this vector and the unit edge normal. If  $S$  is smaller than the threshold at each proposed location (if, perhaps, the edge is occluded in the second frame), ‘no match’ is returned.

### 4.3.3 Motion models

To fully describe the observed image motion, a description of the three-dimensional motion of the objects would be required, together with the depth of all points in the scene and a model of the camera imaging process—in other words a full 3D reconstruction. This is completely general but highly complex and ill-conditioned, due to the vast number of unknowns. However, some attempts have been made in this direction, for example in [9] and [122], which begin with a 2D motion model and then build up to the local depth parameters.

A more practical approach is to use a 2D parametric transformation to describe the motion on the image plane. This 2D problem, with its small number of parameters, is highly overdetermined, efficient and numerically stable. Such models are valid when either the camera translation magnitude is small with respect to the depth of the objects, or where there is only a small amount of depth variation in the scene [9, 78]. In these cases the scene can be considered to be approximately planar. At least one of these situations can be reasonably assumed when considering the small motion between neighbouring frames of a video sequence, and this parametric approach works well in the system presented in this dissertation.

Many common transformations in two-dimensional projective space may be represented by a  $3 \times 3$  matrix operating on a two-dimensional homogeneous co-ordinate  $(x \ y \ 1)^T$ , with the convention that the third value in the co-ordinate is always scaled back to a value of one.<sup>4</sup> As a result, scaled versions of the matrix produce identical transformations and so there are  $9 - 1 = 8$  dimensions to this group. The eight independent modes of deformation are typically expressed as shown in Table 4.3, with the transformation matrices given in the  $M_i$  column.

This group of 2D projective transformations has a number of important subgroups, as shown in Table 4.4. A few pixel-based techniques (e.g. [30]) assume that, locally, the only motion is translation. Many approaches to parametric motion estimation use the 2D affine subgroup (e.g. [37, 107, 158]), while some use a full

---

<sup>4</sup>See [51, 64, 124] for introductions to homogeneous co-ordinates and projective geometry.




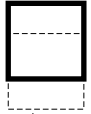
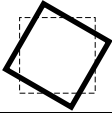
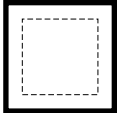

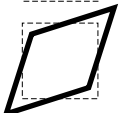


	Deformation	Example	$\mathbf{M}_i$	$\mathbf{G}_i$	$\mathbf{L}_i$
1	$x$ translation		$\begin{bmatrix} 1 & 0 & m \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$
2	$y$ translation		$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & m \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$
3	Rotation about origin		$\begin{bmatrix} \cos m & -\sin m & 0 \\ \sin m & \cos m & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{pmatrix} -y \\ x \end{pmatrix}$
4	Dilation about origin		$\begin{bmatrix} e^m & 0 & 0 \\ 0 & e^m & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{pmatrix} x \\ y \end{pmatrix}$
5	Pure shear		$\begin{bmatrix} e^m & 0 & 0 \\ 0 & e^{-m} & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{pmatrix} x \\ -y \end{pmatrix}$
6	Pure shear at $45^\circ$		$\begin{bmatrix} \cosh m & \sinh m & 0 \\ \sinh m & \cosh m & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{pmatrix} y \\ x \end{pmatrix}$
7	Finite $x$ vanishing point		$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ m & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$	$\begin{pmatrix} x^2 \\ xy \end{pmatrix}$
8	Finite $y$ vanishing point		$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & m & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$	$\begin{pmatrix} xy \\ y^2 \end{pmatrix}$

Table 4.3: *Planar transformations*. The eight planar transformations in the 2D projective group  $P(2)$ , and their corresponding transformation matrices  $\mathbf{M}_i$ , generators  $\mathbf{G}_i$  and vector fields  $\mathbf{L}_i$ .

Group	Modes (see Table 4.3)							
	1	2	3	4	5	6	7	8
Translation	✓	✓						
Euclidean	✓	✓	✓					
Similarity	✓	✓	✓	✓				
2D Affine (GA(2))	✓	✓	✓	✓	✓	✓		
2D Projective (P(2))	✓	✓	✓	✓	✓	✓	✓	✓

Table 4.4: *The hierarchy of two-dimensional transformations*. The subgroups of 2D projective transformations.

projective parameterisation (e.g. [1]). For further reading about this hierarchy in motion estimation, see Bergen et al. [9]. The implementation described in this dissertation allows the user to select any of these modes to be allowed in the model. In practice, the affine subgroup is found to work well in segmenting almost all of the sequences tested (see Appendix D).

#### 4.3.4 Lie group formulation

The 2D projective transformation group, and its subgroups (Table 4.4) are mathematical groups under matrix multiplication, i.e. they are closed and associative, and have inverses and the identity within the group (or subgroup). Each of these is also a Lie group.

A Lie group is a group which is also a smooth manifold (it locally has the topology of  $\mathbb{R}^n$  everywhere). Lie groups provide a useful way of describing the image transformations in a generic way by means of the vector fields that they generate in the image. A more complete discussion of Lie groups and algebras, with a more precise definition, is available in [156] or [119]. The application of Lie groups to edge tracking was introduced by Drummond and Cipolla in [44].

The transformation matrices  $\mathbf{M}_i$  in Table 4.3 each describe one-dimensional family of transformations on  $\mathbb{R}^2$ , parameterised by  $m$ , mapping a point  $(x, y)$  to the transformed point  $(x', y')$ . In each case, setting  $m$  to zero generates the identity transformation. Linearising about the identity (differentiating w.r.t.  $m$ ) creates a series of vector fields, which are the (linearised) motion due to each of the transformation modes:

$$\mathbf{L}_i = \left. \frac{d\mathbf{M}_i(m) \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}}{dm} \right|_{m=0} \quad (4.1)$$

These vector fields may be seen in the final column of Table 4.3. The fields for modes 7 and 8 are non-linear due to the transformation affecting the last element of the homogenous position vector (the projective component), which must then be normalised back to one.

The tangent space to the Lie group at the identity is fundamental to the study of Lie groups, and is known as the Lie algebra. The basis for this space is given by the matrices

$$\mathbf{G}_i = \left. \frac{d\mathbf{M}_i(m)}{dm} \right|_{m=0} \quad (4.2)$$

which are referred to as the generators of the Lie group. The generators for the group P(2) may be seen in Table 4.3. Since differentiation is linear, (4.1) may also

be written as

$$\mathbf{L}_i = \mathbf{G}_i \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (4.3)$$

A point  $\boldsymbol{\alpha} = (\alpha_1 \dots \alpha_n)^T$  in the Lie algebra (with basis  $\mathbf{G}_1 \dots \mathbf{G}_n$ ) is mapped into the Lie group by the exponential map [119, 156]

$$\mathbf{A} = e^{\sum_i \alpha_i \mathbf{G}_i} \quad (4.4)$$

but since  $e^{\mathbf{X}} = \mathbf{I} + \mathbf{X} + \frac{1}{2}\mathbf{X}^2 + \dots$ , for small transformations  $\boldsymbol{\alpha}$ , this can be approximated by a linear sum of the generators:

$$\mathbf{A} \approx \mathbf{I} + \sum_i \alpha_i \mathbf{G}_i \quad (4.5)$$

In the case of the projective  $P(2)$  group and the affine  $GA(2)$  subgroup, this approximation will still yield a matrix in the group.

An image point  $\mathbf{x}$  undergoing a transformation  $\mathbf{A}$  maps to the point  $\mathbf{x}'$  according to (4.5):

$$\begin{aligned} \mathbf{x}' &= \mathbf{A}\mathbf{x} \\ &\approx \mathbf{I}\mathbf{x} + \sum_i \alpha_i \mathbf{G}_i \mathbf{x} \\ &\approx \mathbf{x} + \sum_i \alpha_i \mathbf{L}_i \end{aligned} \quad (4.6)$$

In other words, the displacement of a feature location between two frames is a linear sum of the vector fields at that location.

The power of the Lie group formulation comes in its generality. As long as an independent mode of transformation can be expressed as a group generator, its weighting may be included as another linear term in expression (4.5). The intuitive parameterisations which follow from this formulation assist in the interpretation of the results, and make the application of prior constraints particularly straightforward (see later). In addition, by expressing each of the modes in the hierarchy of 2D projective transformations (Table 4.4) as an independent vector field, and the overall transformation as a linear sum of these (4.6), individual transformation modes may be included or discarded simply by selecting which  $\mathbf{L}_i$  to use. Other modes of deformation can also be added into the framework, or constraints may be placed between the motions of different objects, as described in [45].

### 4.3.5 Solution by re-weighted least squares

Given the Lie group formulation, the motion estimation task is one of estimating the weighting  $\alpha_i$  for each of these deformation modes ( $i = 1 \dots n_d$ ), which can be done by comparing the observed deformation with that predicted by (4.6). As discussed earlier, due to the aperture problem, only the motion normal to the edge can be determined, and so a measurement is taken at each sample point  $k$  to find the normal distance to the image edge,  $d^k$  (Figure 4.3(b)). The expression to be minimised at each sample point is the distance between this and the projection of the fields onto the unit normal  $\hat{\mathbf{n}}^k$ :

$$\text{Error} = d^k - \sum_j \alpha_j (\mathbf{L}_j^k \cdot \hat{\mathbf{n}}^k) \quad (4.7)$$

#### Least squares solution

Over the whole set of  $K$  sample points, the ensemble of errors (4.7) may be expressed in matrix-vector form, and the least squares estimate of  $\boldsymbol{\alpha}$  given by

$$\arg \min_{\boldsymbol{\alpha}} \|\mathbf{d} - \mathbf{N}\boldsymbol{\alpha}\|_2^2 \quad (4.8)$$

where

$$\mathbf{d} = \begin{pmatrix} d^1 \\ d^2 \\ \vdots \\ d^K \end{pmatrix} \quad \mathbf{N} = \begin{bmatrix} \mathbf{L}_1^1 \cdot \hat{\mathbf{n}}^1 & \mathbf{L}_2^1 \cdot \hat{\mathbf{n}}^1 & \dots & \mathbf{L}_n^1 \cdot \hat{\mathbf{n}}^1 \\ \mathbf{L}_1^2 \cdot \hat{\mathbf{n}}^2 & \mathbf{L}_2^2 \cdot \hat{\mathbf{n}}^2 & \dots & \mathbf{L}_n^2 \cdot \hat{\mathbf{n}}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{L}_1^K \cdot \hat{\mathbf{n}}^K & \mathbf{L}_2^K \cdot \hat{\mathbf{n}}^K & \dots & \mathbf{L}_n^K \cdot \hat{\mathbf{n}}^K \end{bmatrix} \quad \boldsymbol{\alpha} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{pmatrix} \quad (4.9)$$

for which the least squares solution is given by the pseudo-inverse

$$\boldsymbol{\alpha} = (\mathbf{N}^T \mathbf{N})^{-1} \mathbf{N}^T \mathbf{d} \quad (4.10)$$

This dissertation calculates  $\boldsymbol{\alpha}$  in the typical manner, defining  $\mathbf{M} = \mathbf{N}^T \mathbf{N}$  and  $\mathbf{v} = \mathbf{N}^T \mathbf{d}$  and solving

$$\boldsymbol{\alpha} = \mathbf{M}^{-1} \mathbf{v} \quad (4.11)$$

The elements of  $\mathbf{M}$  and  $\mathbf{v}$  are directly calculated from the measurements. See Appendix A for full details, or Table 4.5 for a summary.

#### M-estimators: Iterative re-weighted least squares

The least squares solution is the maximum likelihood estimator for sample points whose errors are independent and normally distributed. In this dissertation it is found that this is not the correct model and that the errors tail off more slowly,

resembling a Laplacian distribution (see Section 4.4.4, particularly Figure 4.6). Such distributions can be handled within the least squares framework by iterative re-weighting to implement an M-estimator [70, and Appendix A.3]. The re-weighting function used in this dissertation is

$$w(d^k) = \frac{1}{c + |d^k|} \quad (4.12)$$

with a value of  $c = 1$ . The measurements are multiplied by this factor, reducing the influence of gross outliers. This M-estimator is the maximum likelihood estimator for distributions which behave as a Laplacian for most values of  $d^k$ , but a Gaussian for small values of  $d^k$  (which avoids the discontinuity that would occur if a pure Laplacian were used). Since the motion estimation is iterative, as part of an Expectation-Maximisation loop (see Section 4.4.2), iterative re-weighted least squares is a natural solution.

### Translation prior: Regularisation

It is found from experience that the majority of motions in video sequences are translational. Rather than restrict the motions to the translation subgroup, it is found to be more useful merely to place a prior on the solution. This may be achieved by a *regularisation* of the solution (Appendix A.4). The term  $\lambda \mathbf{R}$  is added to the covariance matrix  $\mathbf{M} = \mathbf{N}^T \mathbf{N}$ , where  $\mathbf{R}$  is the diagonal matrix

$$\mathbf{R} = \begin{bmatrix} 0 & & & & \\ & 0 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix}_{n_d \times n_d} \quad (4.13)$$

and  $\lambda$  is selected to make the prior of similar magnitude to the data. This is achieved by using

$$\lambda = \frac{1}{n_d} \text{Tr}(\mathbf{M}) \quad (4.14)$$

where  $n_d$  is the number of vector fields (dimensions) and the function  $\text{Tr}()$  returns the trace of a matrix.  $\lambda$  is thus the mean of the diagonal elements.

### Solution by SVD with normalisation

Finally, the re-weighted least squares solution is calculated using Singular Value Decomposition (SVD) [112], with the matrix normalised to increase the conditioning

(Appendix A.5). The complete motion estimation subsystem is outlined in Table 4.5.

At each iteration of the motion estimation, a new search is made for the real edge starting from the sample point positions given by the current motion estimates. The distance  $d^k$  measured at each iteration is therefore itself a residual measurement—the error between the real location and the current motion estimate.<sup>5</sup> The system outlined in Table 4.5 calculates a correction term  $\alpha^{(k)}$  to minimise this residual further, where  $(k)$  is the iteration number. Clearly, these correction terms should tend to zero as the iteration process progresses. Once the iteration has converged, the maximum likelihood estimate for the motion is given by the sum of the  $\alpha^{(k)}$  terms over all the iterations:

$$\text{Motion estimate } \theta = \sum_{\text{iterations}} \alpha^{(k)} \quad (4.15)$$

## 4.4 Multiple motion estimation using EM

### 4.4.1 Dominant vs simultaneous multiple motion estimation

Motion estimation becomes a more difficult proposition where there are independent moving objects in a sequence. If it were known *a priori* which edges belonged to which object then the motion for each object could be estimated independently using just the correct edges, using the method described in Section 4.3 above. Of course, the edge labelling is not known *a priori*—the reason why the motions are being estimated is in order to then label the edges according to their motion.

The *dominant motion* technique is a popular method for solving this circular problem [5, 40, 75, 78, 107]. Under this scheme, one motion is robustly fitted to all the data, and any features or pixels which are identified as ‘conforming’ to this ‘dominant motion’ (according to some criteria) are labelled as part of that motion segment and are removed from the estimation process. The process is then repeated with the remaining features or pixels in order to find the other motions. While this technique works well in some cases, particularly when much of the frame does obey one motion (for example when a large area of the scene is background), it relies heavily on the ability of the motion estimation scheme to correctly extract one of the motions in the scene when presented with very noisy data (i.e. containing all motions). For reliable results, this is dependent on the use of *robust estimators*.

---

<sup>5</sup>Because  $d^k$  is already a residual measurement, this is the value used in calculating the re-weighting factor  $w^k$ .

- **Make measurements**

For each sample point  $k$

- Transform  $k$  according to current motion estimate
- Compute unit normal to edge,  $\hat{\mathbf{n}}^k$
- Search along closest compass direction for an edge
- Compute residual error

$$d^k = (\text{distance to edge}) \cdot \hat{\mathbf{n}}^k$$

- **Calculate weighted measurement matrices**

For  $i, j = 1 \dots n$

- Let  $M_{ij} = v_i = 0$
- For each sample point  $k$

$$w^k = \frac{1}{1 + |d^k|}$$

$$v_i = v_i + d^k (\mathbf{L}_i \cdot \hat{\mathbf{n}}^k) w^k$$

$$M_{ij} = M_{ij} + (\mathbf{L}_i \cdot \hat{\mathbf{n}}^k) (\mathbf{L}_j \cdot \hat{\mathbf{n}}^k) w^k$$

- **Regularise (translation prior)**

For  $i = 3 \dots n$

$$M_{ii} = M_{ii} + \frac{1}{n} \sum_{k=1}^n M_{kk}$$

- **Calculate normalisation factors**

For  $i = 1 \dots n$

$$S_i = \frac{1}{\sqrt{M_{ii}}}$$

- **Pre-normalise**

For  $i, j = 1 \dots n$

$$v'_i = v_i S_i$$

$$M'_{ij} = M_{ij} S_i S_j$$

- **Compute  $\alpha' = \mathbf{M}'^{-1} \mathbf{v}'$  using SVD**

- **Post-normalise**

For  $i = 1 \dots n$

$$\alpha_i = \alpha'_i S_i$$

Table 4.5: *Motion estimation algorithm*. See Section 4.3.5 and Appendix A for further details; the entire table is iterated until convergence. When used as part of the EM algorithm (see Section 4.4.5), the edge responsibilities must also be considered—see Table 4.7.

One robust approach is to use an M-estimator, as introduced in Section 4.3.5 and Appendix A [68, 107, 121]. However, while M-estimators are very effective at providing a maximum-likelihood estimation for non-Gaussian (but known) distributions, they fare much less well with gross outliers. In their survey paper on robust estimation [146], Torr and Murray found that M-estimators are poor for more than 20–25% outliers in the data set. In other words, unless over 75% of the data obey one motion (and are matched correctly), the M-estimator will not yield a good solution. It is unreasonable to expect this to be the case in all motion segmentation scenarios.

The most effective robust methods are based on random sampling: either Least Median of Squares (LMedS) [118], or RANSAC [54]. Under these schemes a number of trials are made, using random subsets of the data, in the hope that one of those subsets will contain no outliers and thus will yield good results. Torr and Murray [146] recommend LMedS and find that it works well for at least up to 50% outliers. While these schemes do work well (e.g. [5, 145]), one problem is selecting a reasonable subset. Ideally, this should contain the minimal set of points necessary (to reduce the probability that one of these is an outlier). This minimal set is difficult to define in the case of edge motions since the conditioning of the solution depends on the direction as well as the number of normal motions found. Even with point features, the solution can be ill-conditioned if the points are poorly selected. In [163], Zhang et al. suggest a bucketing technique whereby the image is divided into a number of bins and only one point can be taken from each bin. However, in the case of more than one motion this may make the problem worse, as each moving object may only be represented in a few bins. A random-sampling approach was tested for the edge-based motion segmentation scheme in this dissertation, but it was found that this problem of selecting a suitable subset was a major complication.

When it comes to feature or pixel labelling, dominant motion schemes tend to use a greedy approach. All features or pixels which match the first motion (up to a threshold) are labelled as that motion and are removed from the process even if, later, a motion is found which they would fit better. An obvious solution to this is to, at the end of the process, reassign features or pixels by comparing with all possible motions. And once this has been done, these motions could then be re-estimated to better fit their altered regions of support. However, this is then essentially *simultaneous* multiple motion estimation, with the initial greedy algorithm as an initialisation stage.

Simultaneous multiple-motion estimation avoids some of the problems of the dominant motion approach by modelling the complete system from the start. If it is desirable to segment the sequence into two motions then two motions are fitted



from the start rather than fitting first one single-motion model and then another. This brings its own problems as the number of motions must be determined *a priori*, and fitting a larger number of parameters will always be a more difficult problem. However, a simultaneous approach is selected for this dissertation, and selecting the correct number of motions is discussed in Chapter 7. The current chapter assumes that it is known that there are two motions to be fitted.

It is possible to place all the parameters (motion parameters and the feature/pixel labelling) into one vast minimisation scheme, but the classic solution to these circular labelling/estimation schemes is the Expectation-Maximisation (EM) algorithm [43] which alternately minimises the two sets of parameters. This is the approach followed by, for example, Jepson and Black in fitting Gaussian mixture models to dense motion fields [85], and by Sawhney and Ayer [121], amongst others, for motion segmentation. EM has also proved to be a good solution to the problem of multiple edge motions presented in this dissertation. Furthermore, the EM algorithm ties in naturally with the statistical framework developed in this dissertation since it produces, and makes use of, a labelling probability for each edge.

#### 4.4.2 The Expectation-Maximisation algorithm

Introduced by Dempster et al. in 1977, the Expectation-Maximisation algorithm [13, 43] is a general method for finding the maximum likelihood estimate of the parameters of a distribution when there is missing data (in this case the motion labels for each of the edges). For a distribution governed by a set of parameters  $\Theta$ , and a set of data  $Z = \{z_1, \dots, z_N\}$  drawn from this distribution, the likelihood of this data,  $\mathcal{L}[\Theta; Z]$ , is given by:

$$\mathcal{L}[\Theta; Z] = P(Z|\Theta) \quad (4.16)$$

$$= \prod_{i=1}^N P(z_i|\Theta) \quad \text{if the data are independent} \quad (4.17)$$

The case considered by Dempster et al. is when the data set is incomplete, i.e. only data  $X$  is observed, out of the complete data set  $Z = \{X, Y\}$ . This can occur either due to missing data or if it is not possible to observe  $Y$  (it is a hidden parameter). The likelihood then becomes

$$\mathcal{L}[\Theta; Z] = \mathcal{L}[\Theta; X, Y] = P(X, Y|\Theta) \quad (4.18)$$

and it is this expression which must be maximised for the maximum likelihood estimate of  $\Theta$ . It is important to realise that the value of function  $\mathcal{L}[\Theta; X, Y]$  is a

random variable since the missing information  $Y$  is unknown and random, and this likelihood function can be considered a function of  $X$  and  $\Theta$ , which *are* known. It is therefore necessary to consider the *expected value* of (4.18),  $\mathcal{E} [\mathcal{L} [\Theta; X, Y]]$ .

The EM algorithm casts this expectation as part of an iterative update scheme, considering at each iteration the function  $\mathcal{L} [\Theta; X, Y]$  given by the values  $\Theta^{(i-1)}$ , from the previous iteration, and  $X$ . This conditional expectation is defined as

$$Q (\Theta, \Theta^{(i-1)}) = \mathcal{E} [\log \mathcal{L} [\Theta; X, Y] | X, \Theta^{(i-1)}] \quad (4.19)$$

This is maximised to give an improved estimate for  $\Theta$ :

$$\Theta^{(i)} = \arg \max_{\Theta} Q (\Theta, \Theta^{(i-1)}) \quad (4.20)$$

If the unknown data  $Y$  is a series of discrete states  $y_j$  (as is the case with the labelling of edges), then the expectation can be expressed as the sum over the state probabilities:

$$Q (\Theta, \Theta^{(i-1)}) = \mathcal{E} [\log \mathcal{L} [\Theta; X, Y] | X, \Theta^{(i-1)}] \quad (4.21)$$

$$= \sum_{\mathbf{y}} \log \mathcal{L} [\Theta; X, \mathbf{y}] P (\mathbf{y} | X, \Theta^{(i-1)}) \quad (4.22)$$

In this form the iterative nature becomes clear: the labelling probabilities (also known as responsibilities),  $P (\mathbf{y} | X, \Theta^{(i-1)})$ , can first be calculated given the current parameters (this calculation of expectations is known as the E-stage). Then the expected likelihood (4.22) is maximised (the M-stage), given these values of  $\mathbf{y}$ , to give an updated estimate of  $\Theta$ .

In this dissertation, the known data are the edges and the sample point matches, the missing data are the motion labels for each edge, and the parameters to be estimated are those of the two motions. Using the notation of Chapter 3, the maximisation of (4.22) can be written as

$$\arg \max_{\Theta_{n+1}} \sum_{\mathbf{e}} \log P (\mathbf{e} \mathbf{D} | \Theta_{n+1}) P (\mathbf{e} | \Theta_n \mathbf{D}) \quad (4.23)$$

The final term represents the edge label probabilities, which are calculated in the E-stage by referring to the sample point errors (see Section 4.4.4). Given these, (4.23) is maximised using weighted least squares (Section 4.4.5). The process is outlined in Table 4.6.

- Initialise motions  $\Theta_0$
- Repeat (EM Loop)
  - **E-Stage**  
Calculate edge label probabilities  $P(e|\Theta_n D)$
  - **M-Stage**  
Estimate the motions  $\Theta_{n+1}$  given the current edge label probabilities
- until convergence

Table 4.6: *The EM algorithm for multiple motion estimation using edges.* See Table 4.1 for context.

### 4.4.3 Initialisation

The most difficult part of any iterative algorithm is the initialisation, and EM is no exception. Given the circular nature of the problem and the algorithm, it must be started by some guess, either of the parameters or the labelling. While the EM algorithm is guaranteed to improve the likelihood at each step [43], it can only improve upon this initial guess and if this is too far from the true maximum then there is a danger EM will converge to a local maxima.<sup>6</sup>

In developing the work for this dissertation, various heuristic initialisation techniques have been tried, for example using the null motion and the mean motion as the two initial motion guesses. However, the danger of using heuristics is that they must be appropriate to the task, and if used inappropriately they can be counter-productive. Consider the case, with a null/mean motion initialisation, where both foreground and background are moving with a similar, large motion. All edges will therefore obey the mean motion, and will continue to throughout the iterations—the two independent motions will never be detected. For the two motion case considered in this chapter it has been found that such heuristics are not necessary, and the easiest way to achieve a reasonable initial motion estimate is through a random initial edge labelling. Chapter 6 introduces a more sophisticated initialisation technique for the cases where there are more than two motions.

The random initialisation starts by measuring the motion of each sample point. Taking each of the sample points in frame 1, a match is found in frame 2 for each of the sample points by searching, from their initial location, for a distance  $\rho$  in each direction normal to the edge ( $\rho = 20$  pixels). The edges are then randomly divided

<sup>6</sup>In fact, while the edge-based system of this dissertation does find a maximum, it does not do so monotonically (see Section 4.4.6).

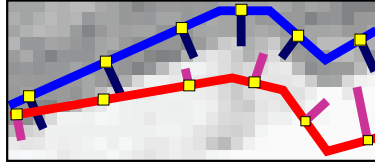


Figure 4.5: *Evaluating edge probabilities.* The edge is considered under both possible motion models and the error distances under each motion are used to estimate the probability of a good match.

into two groups, and the sample point residuals from the two groups are used to estimate two initial motions (according to Section 4.3). Then the EM can begin at the E stage, which estimates the probabilities that the edges obey these motions. The advantage of a random initialisation is that it provides, to a high probability, two motions which are plausible across the whole frame, giving all edges a chance to contribute an opinion on both motions.

#### 4.4.4 Expectation: Calculating edge probabilities.

The first stage in the EM iteration is to calculate the edge responsibilities,  $P(\mathbf{e}|\Theta_n\mathbf{D})$ . That is, for each edge, the probability of its assignment to each of the possible motions. The obvious data to use to estimate this are the sample point errors, already used to calculate the motions. An ensemble of small errors clearly indicates a more likely fit than a motion which has large residual errors.

Figure 4.5 demonstrates the process. The edge is transformed according to each motion model and a search is made from each sample point for a match. The set of distances under motion 1 (one residual per sample point) will be referred to as data  $\mathbf{D}_1$ , while the error distances under motion 2 are  $\mathbf{D}_2$ . If a sample point does not find a match under a motion then that is considered as a special distance code, which is included in the data set for that motion in the same way as if a match had been found.

It is a reasonable assumption that these two sets of sample point errors encapsulate all of the information from  $\Theta_n$  and  $\mathbf{D}$  that is necessary to label the edge, and hence the edge label probability can be written as

$$P(\mathbf{e}|\Theta_n\mathbf{D}) = P(\mathbf{e}|\mathbf{D}_1\mathbf{D}_2) \quad (4.24)$$

Calculating the probability that the edge fits one motion rather than another, i.e. evaluating (4.24), is a standard case for using Bayes' rule. This is used when comparing how well the observed data are modelled according to each of the possible hypotheses [58, 84]. In this case the observed data are both sets of sample point

errors, and the hypotheses are either that motion 1 is correct, or that motion 2 is correct. It is important to realise at this point that if, for example, motion 1 were correct, this should explain both the errors observed under motion 1 *and* those under motion 2, since the background distribution is not uniform. Denoting this first hypothesis by ‘ $e = 1$ ’, and so on, the probability that this is correct is given by

$$P(e = 1 | \mathbf{D}_1 \mathbf{D}_2) = \frac{P(\mathbf{D}_1 \mathbf{D}_2 | e = 1) P(e = 1)}{P(\mathbf{D}_1 \mathbf{D}_2)} \quad (4.25)$$

$$= \frac{P(\mathbf{D}_1 \mathbf{D}_2 | e = 1) P(e = 1)}{P(\mathbf{D}_1 \mathbf{D}_2 | e = 1) P(e = 1) + P(\mathbf{D}_1 \mathbf{D}_2 | e = 2) P(e = 2)} \quad (4.26)$$

The prior probabilities of the two motion labels are equal, since there is no particular meaning at this stage to each motion label—foreground and background labelling comes later. A modelling assumption is also made: that the two data sets may be treated as independent. There is in fact a small correlation between the data under each model, but the probabilities reported by assuming independence are not unreasonable.<sup>7</sup> By assuming independence, and given equal priors, the edge probability is given by

$$P(e = 1 | \mathbf{D}_1 \mathbf{D}_2) = \frac{P(\mathbf{D}_1 | e = 1) P(\mathbf{D}_2 | e = 1)}{P(\mathbf{D}_1 | e = 1) P(\mathbf{D}_2 | e = 1) + P(\mathbf{D}_1 | e = 2) P(\mathbf{D}_2 | e = 2)} \quad (4.27)$$

The four different terms present in (4.27) represent only two different scenarios, either the probability that data  $\mathbf{D}_i$  comes from an edge obeying motion  $i$  (i.e. it is the correct motion), or that the data are due to the incorrect motion. It is assumed that one distribution is sufficient to model the data under any incorrect motion, but not that this is a uniform distribution.

This latter point is critical to the understanding of (4.27) since intuition might indicate that the terms under the incorrect motion,  $P(\mathbf{D}_i | e \neq i)$ , are superfluous. However, this would only be the case if this distribution were uniform, and this is most certainly not the case in this application (see Figure 4.6(b)). Implicit in the ‘ $e = 1$ ’ hypothesis is that the errors in  $\mathbf{D}_2$  are drawn from the background distribution, since the edge can only be labelled with one motion. Since some errors are more likely than others in this distribution, this implicit assignment of  $\mathbf{D}_2$  to the  $P(\mathbf{D}_i | e \neq i)$  distribution must also have some impact on the hypothesis.

The two distributions,  $P(\mathbf{D}_i | e = i)$  and  $P(\mathbf{D}_i | e \neq i)$  are modelled using the sample point errors. As another simplifying assumption, it is assumed that these errors are independent along an edge. This means that the edge probability is the product

<sup>7</sup>Appendix C investigates this assumption in more detail.

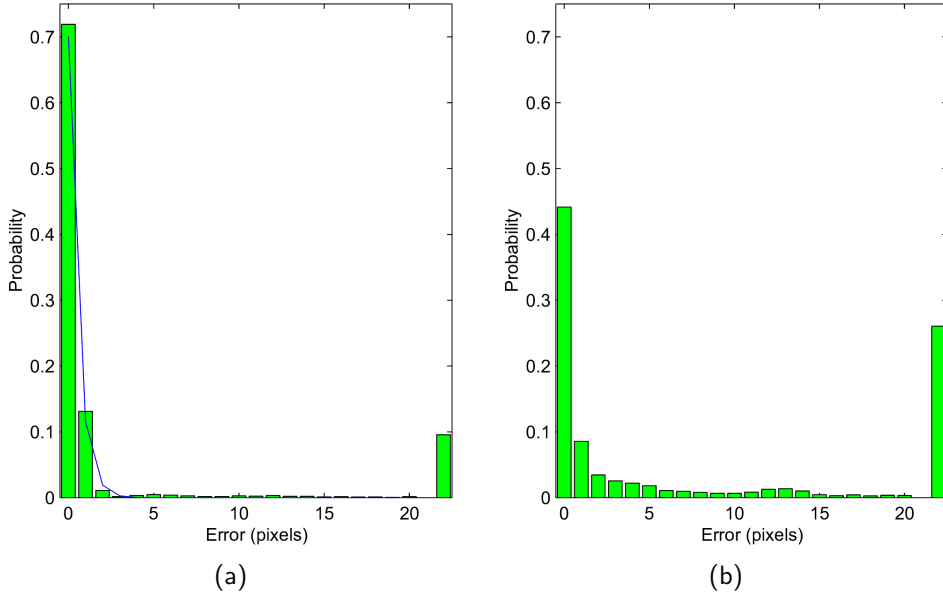


Figure 4.6: *Edge statistics*. Distribution of sample point measurement errors  $d_i^k$  under (a) the correct motion ( $P(d_i^k | e = i)$ ) and (b) the incorrect motion ( $P(d_i^k | e \neq i)$ ). A Laplacian distribution is overlaid on graph (a), showing a reasonable match.

of the individual sample point error probabilities:

$$P(\mathbf{D}_i | \mathbf{e} = i) = \prod_{k \in \mathbf{e}} P(d_i^k | \mathbf{e} = i) \quad (4.28)$$

$$P(\mathbf{D}_i | \mathbf{e} \neq i) = \prod_{k \in \mathbf{e}} P(d_i^k | \mathbf{e} \neq i) \quad (4.29)$$

where  $d_i^k$  is the error at sample point  $k$  under the  $i$ th motion. Assuming independence gives improved simplicity and flexibility, at the expense of some saturation of the probability estimates (see Appendix C for an investigation into these independence assumptions).<sup>8</sup>

The distributions of sample point errors  $d_i^k$  under the correct and incorrect motions have been estimated from the accumulated statistics of thirty different test sequences, a subset of those shown in Appendix D. Using earlier estimated statistics, EM was run to convergence to find the two motions, and the correct motion for each edge was then labelled by hand. The resulting distributions are shown in Figure 4.6.

<sup>8</sup>Appendix C, in particular, considers modelling the sample point errors along an edge as a first order Markov process along an edge [61]. It is found that, while the errors under the ‘correct motion’ hypothesis are largely independent, there is considerable structure to the errors under the incorrect motion.

Since matches are only made to the nearest pixel, the distributions are discrete and are given as a function of the number of steps made from the initial location, regardless of whether those steps were aligned with the pixel grid (N, E, S, W), or at 45° (NE, NW etc.). These could be stored as two different distributions, but the difference between them is found to be minimal, so they are combined as one.

Looking at the distribution under the correct motion (Figure 4.6(a)), it can be seen that the vast majority of sample points are matched with zero error. Very few sample points, 4%, find a match at a distance of more than 2 pixels. About 10% of all sample points fail to find a good match. Under the incorrect motion (Figure 4.6(b)) there are still a large number of sample points which find their best match with zero error. This is because in most sequences there are a number of edges which are along the line of both motions, and so the sample points provide a good match under both. Also, since the inter-frame motions tend to be small, most errors under the incorrect motion are also small. There are, however, a significantly greater number of failed matches. A Laplacian distribution has been overlaid on Figure 4.6(a) in blue and it can be seen that, as stated in Section 4.3.5, it provides a good fit, and justifies the use of the selected M-estimator.<sup>9</sup>

Returning to the edge probabilities of (4.27), and defining the likelihood ratio  $\mathcal{L}_{\mathcal{R}}$  as the ratio of the two distributions:

$$\mathcal{L}_{\mathcal{R}}(d_i^k) = \frac{P(d_i^k | \mathbf{e} = i)}{P(d_i^k | \mathbf{e} \neq i)} \quad (4.30)$$

equation (4.27) can be rewritten as:

$$P(\mathbf{e} = 1 | \mathbf{D}_1, \mathbf{D}_2) = \frac{\prod_{k \in \mathbf{e}} \mathcal{L}_{\mathcal{R}}(d_1^k)}{\prod_{k \in \mathbf{e}} \mathcal{L}_{\mathcal{R}}(d_1^k) + \prod_{k \in \mathbf{e}} \mathcal{L}_{\mathcal{R}}(d_2^k)} \quad (4.31)$$

Thus the probability that an edge is motion 1 is the product of the sample point likelihood ratios under that motion, normalised over all motions. Figure 4.7 shows the likelihood ratio derived from experiments. This discrete distribution could be directly used in the system, but to guarantee a maximum likelihood solution at an error of zero, it is better to smooth the values at larger errors and provide a model which increases monotonically towards zero.<sup>10</sup> The model used in this system is shown in in blue in Figure 4.7. This uses the raw values for the first two errors, and

<sup>9</sup>A Laplacian distribution is a double-sided exponential distribution. In this dissertation the sample point residual errors are double-sided (they are both positive and negative), but for clarity only the positive half of the distribution is shown in the figures in the chapter.

<sup>10</sup>There are few examples of the larger errors in the data set used, so these values are expected to be noisy.

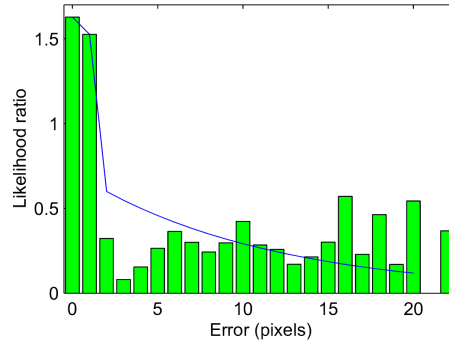


Figure 4.7: *Sample point likelihood ratio*. The likelihood ratio for each error  $d^k$ , given by the ratio of the number of good matches at that distance to bad matches (using the data of Figure 4.6). The blue line shows the smoothed, monotonic likelihood ratio used in this system.

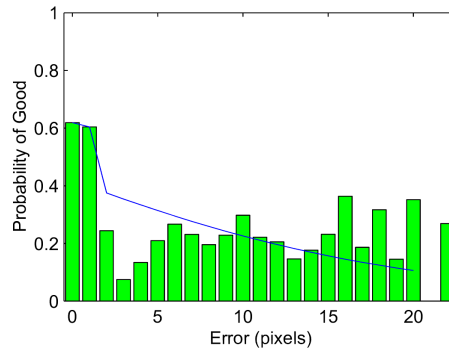


Figure 4.8: *Probability of a good match*. The probability that a sample point with error  $d^k$  comes from an edge obeying the correct motion. (Figure 4.6(a) normalised against the sum of Figures 4.6(a) and 4.6(b).) The modelled distribution (from the likelihood ratio) is shown in blue.

then an exponential decay.

The edge motion probabilities given by (4.31) are the responsibilities required for the E-stage of EM. Given two motions (either from the initialisation or from the M-stage), the edges are transformed under each motion and the residuals  $d_i^k$  found. For each edge the probability that it is motion 1 is calculated from (4.31); the probability that it is motion 2 is given by  $(1 - P(e = 1 | \mathbf{D}_1, \mathbf{D}_2))$ . As well as being the responsibilities used in the M-stage, these are also (once EM has converged) the final edge probabilities used in labelling regions (Section 4.6).

### Is this match correct?

As an aside, the likelihood ratio may also be used to calculate the probability that an edge is correctly matched under a motion (without any data from other motions). This probability is used in Chapter 6 to determine whether an edge is ‘trackable’ or



not. The probability is given by

$$P(\mathbf{e} = i | \mathbf{D}_i) = \frac{P(\mathbf{D}_i | \mathbf{e} = i) P(\mathbf{e} = i)}{P(\mathbf{D}_i)} \quad (4.32)$$

$$= \frac{P(\mathbf{D}_i | \mathbf{e} = i) P(\mathbf{e} = i)}{P(\mathbf{D}_i | \mathbf{e} = i) P(\mathbf{e} = i) + P(\mathbf{D}_i | \mathbf{e} \neq i) P(\mathbf{e} \neq i)} \quad (4.33)$$

and if the match is known to be correct for one of the two motions considered, then the priors are equal:

$$= \frac{P(\mathbf{D}_i | \mathbf{e} = i)}{P(\mathbf{D}_i | \mathbf{e} = i) + P(\mathbf{D}_i | \mathbf{e} \neq i)} \quad (4.34)$$

$$= \frac{\prod_{k \in \mathbf{e}} P(d_i^k | \mathbf{e} = i)}{\prod_{k \in \mathbf{e}} P(d_i^k | \mathbf{e} = i) + \prod_{k \in \mathbf{e}} P(d_i^k | \mathbf{e} \neq i)} \quad (4.35)$$

$$= \frac{\prod_{k \in \mathbf{e}} \mathcal{L}_{\mathcal{R}}(d_i^k)}{\prod_{k \in \mathbf{e}} \mathcal{L}_{\mathcal{R}}(d_i^k) + 1} \quad (4.36)$$

This probability is plotted in Figure 4.8, with the modelled distribution used in this implementation shown in blue. This is calculated from the likelihood ratio distribution according to (4.36). As expected, a sample point error of less than 2 implies that it is more likely that the point obeys the motion ( $P(e_i = i | \mathbf{D}_i) > 0.5$ ), whereas if the distance is larger it is likely to be incorrect.

#### 4.4.5 Maximisation: Calculating motions

The M-stage of EM calculates the most likely values of the motion parameters given the current edge responsibilities. According to the discrete version of EM (4.22) the expression to be maximised is the weighted sum of the edge likelihoods, where the weights used are the responsibilities from the E-stage. This may be performed using weighted least squares, as shown in Appendix B, performing one maximisation for each set of motion parameters.

In the case of two motions, this is achieved by applying the motion estimation algorithm of Section 4.3 twice, but now also weighting the measurements for each edge first by the responsibilities under motion 1,  $r_1(\mathbf{e}) = P(\mathbf{e} = 1 | \mathbf{D}_1, \mathbf{D}_2)$ , and then by those under motion 2. The modification required to the motion estimation algorithm of Table 4.5 (to estimate motion 1) is shown in Table 4.7.

It is here that the scheme diverts from standard EM practice in two ways. First, the motion estimation stage does not completely maximise the probability of the data, and instead only one iteration of the weighted least squares is performed.

	$\vdots$
– For each edge $\mathbf{e}$	
* For each sample point $k \in \mathbf{e}$	
	$w^k = \frac{1}{1 +  d^k }$
	$M_{ij} = M_{ij} + (\mathbf{L}_i \cdot \hat{\mathbf{n}}^k) (\mathbf{L}_j \cdot \hat{\mathbf{n}}^k) w^k r_1(\mathbf{e})$
	$v_i = v_i + d^k (\mathbf{L}_i \cdot \hat{\mathbf{n}}^k) w^k r_1(\mathbf{e})$
	$\vdots$

Table 4.7: *Multiple motion estimation.* Modification to Table 4.5 in order to calculate multiple motions via EM (here showing the calculation of motion 1) . The measurements are weighted by the edge responsibility (the probability that the edge obeys that motion):  $r_1(\mathbf{e}) = P(\mathbf{e} = 1 | \mathbf{D}_1, \mathbf{D}_2)$

This generates a solution which is refined in further iterations of EM. Second, the maximisation is not over all the data, but only over the data for one motion. The motion estimation merely tries to minimise the sample point errors under the assigned motion model, and not to also maximise the errors for the other motion. In fact, it is by no means clear that considering the second motion model is at all desirable. However, using the ‘incorrect’ motion model *is* useful in estimating the responsibilities and estimating the final edge probabilities.

#### 4.4.6 Convergence

The progress of the EM algorithm is monitored by considering the likelihood that is being maximised. Since sample points are assumed to be independent, this is the product over all sample points of the individual error probabilities, where the error probability for each sample point is taken to be the weighted average (according to the edge responsibility) over the different motions:

$$\text{Likelihood} = \prod_{\substack{\text{all edges} \\ \mathbf{e}}} \prod_{k \in \mathbf{e}} \sum_{m=1}^2 r_m(\mathbf{e}) P(d_m^k | \mathbf{e} = m) \quad (4.37)$$

Figure 4.9 shows how this likelihood changes over a typical run of EM as implemented in this system. It can be seen that the likelihood, in general, increases as the algorithm progresses and then levels out, but that there is also some noise in this process. EM is usually guaranteed to increase the likelihood with every iteration

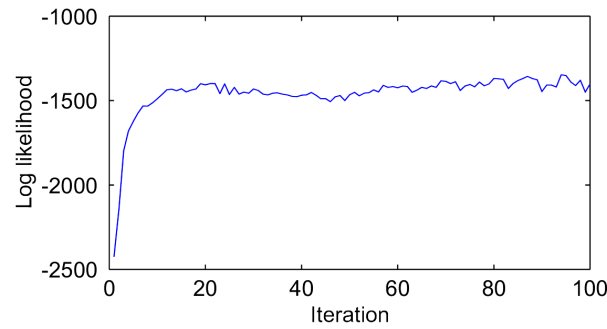


Figure 4.9: *EM convergence*. The log likelihood at each iteration of EM. In this system EM does not always improve the likelihood since the sample points find new matches at each iteration, which changes the data.

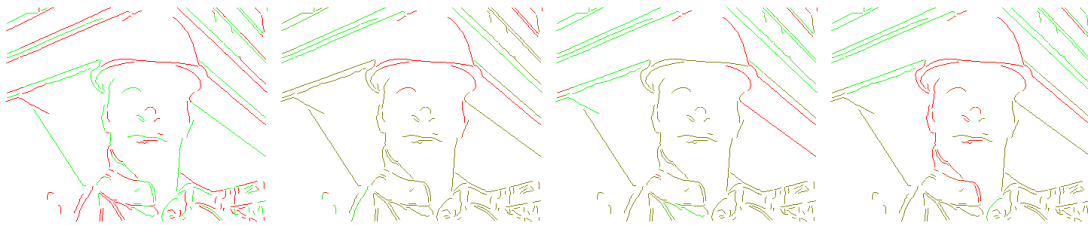


Figure 4.10: *Edge probabilities as EM converges*. Initial (random) edge labelling and then the first three iterations of EM. The edge colour blends from red (motion 1) to green (motion 2) according to the probability of each motion for that edge; consequently yellow indicates an edge with equal probability of either motion. Note that after very few iterations the solution is already very good. The final (converged) edge labelling can be seen in Figure 4.1(b).

but this is not the case here, for a number of reasons. For example, the probabilities are only partially maximised at each iteration, and the data are also different at each iteration (since the sample points are mapped under a revised motion and search along their new normal for a match). Quantisation also plays a small part, as matches are only found to the nearest pixel—small changes in the motion can therefore cause a jump of one pixel in the distance to the best match, and a related jump in the probabilities.

The qualitative solution (the edge probabilities) do not change significantly once the graph flattens out, and it can be seen from Figure 4.10 that the edge labelling appears reasonable after only very few iterations. It has been found sufficient to declare convergence when the likelihood has not risen above its current maximum for 10 iterations. Convergence is usually reached after 20–30 iterations, which for a typical image with about 1,000 sample points takes around 3 seconds on a 300MHz PC.

## 4.5 Finding regions

Having obtained the set of tracked edges, and labelled these according to their motions, it is necessary to build on these to label the rest of the pixels. Chapter 3 showed that the set of labelled edges were sufficient to label the complete image, when the rest of the image is divided up into regions of similar colour. The first task is therefore to divide the image into these regions.

### 4.5.1 Choice of segmentation scheme

There are an increasing number of motion segmentation schemes in the literature which use a static segmentation of the frame [12, 27, 68, 102, 110, 141, 153]. In each case, the authors point out that the choice of segmentation scheme does not restrict their proposed technique. The challenge in motion segmentation is not the static segmentation scheme, but the means by which regions are labelled with their motion.

All of the papers mentioned above do identify their favoured static segmentation scheme, the most popular approach being a watershed segmentation [157], which is a fast morphological region-growing approach (used, for example, in [12, 27, 110]). This technique grows regions from the minima of a gradient image until they meet, giving closed regions whose boundaries are at maxima in the gradient image. The watershed segmentation therefore does give region boundaries which are image edges, but only in an opportunistic manner, and it is worth considering schemes which make more explicit use of edges. The edges of the regions must agree with those edges which have been tracked and labelled according to their motions.

Edge completion schemes, which attempt to join up the known edges to make complete closed contours, are one possibility but, given that the aim is to combine pixels of similar colour into regions, a scheme which considers these pixels is more appropriate. Sinclair [130] has developed an image segmentation scheme which explicitly uses a set of provided edges in a region-growing approach. His segmentation scheme is currently applied to static image segmentation and indexing schemes [117].

The particular static segmentation scheme used is not a major consideration in this dissertation, and any scheme that enables the image edges to be used would be suitable. The Sinclair scheme is fast enough to be of practical use and produces pleasing segmentations, and so is the scheme selected. It would also be possible to modify watershed techniques to use the existing edges as hard boundaries.

### 4.5.2 Voronoi seeded image segmentation

Sinclair's segmentation scheme [130] uses morphological region growing, starting from points distant from edges and stopping when edges are encountered. The first stage, finding the seed points, uses a distance transform (Voronoi image) of the edge image. A distance transform assigns to each a pixel a value, in this case the distance to the closest edge. The distance transform only needs to be approximate, and an efficient calculation method is the chamfer technique popularised by Borgefors [22]. Figure 4.11(a) shows an edge image and Figure 4.11(b) the related distance transform (dark areas are furthest from edges). The peaks of the distance transform image, being the points furthest from edges, are taken to be seed points for region growing.

From each seed point a 'seed region' is grown, which consists of all the simply-connected pixels which are of very similar colour to the seed point (Figure 4.11(c)). Each region begins with just one pixel, the seed point, but then each pixel adjoining this is tested to see if it should be included in that region. There are two possible criteria for membership. If the colour difference between the candidate pixel and its neighbour (already in the region) is within the estimated standard deviation then it is included. This test considers the difference in each of the red, green and blue colour components independently and the criterion is met if the difference in each is smaller than 3 (assuming 24-bit images, and so a maximum of 255 in each component). If the difference in colour is larger than this then the pixel may still be included in the region if the colour difference between the candidate pixel and the current mean region colour is less than a second threshold (15 in each component). This process continues with pixels on the new boundary being considered for membership next. Pixels already labelled as belonging to another region, or which are original image edges, are not considered for membership. In this way the image edges act as hard barriers through which regions are not allowed to grow. The region growing stops when no boundary pixels satisfy the colour criteria.

Once all the seed regions have been established, blind region growing is performed simultaneously from each seed region. Any pixel adjoining a seed region which is not already assigned to a region is added to that region, regardless of colour. Once each region has absorbed one layer of pixels, the process is repeated to enlarge each region again, until all pixels have been labelled. In an improvement to the original scheme, which performed this last step continually until convergence, this dissertation presents a two-stage approach which prevents regions leaking through small gaps in a fragmented edge. In the first stage, pixels which are within  $\gamma$  pixels of an edge are not considered for merging, so that each region can grow until near an



Figure 4.11: *Example region segmentation.* From the initial edges (a), a distance transform image is calculated (b). The peaks of this gives seed points, which are expanded into regions of similar colour (c). A morphological operator is then applied which grows regions first until they are 3 pixels from edges (d), and then all the way to the edge (e). Finally, regions of similar colour are merged and edges are assigned to the region of closest colour (f).

edge, but not close enough to then bleed through gaps.<sup>11</sup> In this dissertation  $\gamma = 3$ , which has been determined empirically to be a reasonable value. The convergence of this first stage is demonstrated in Figure 4.11(d). Once this stage has concluded, the restraint is lifted and regions are allowed to grow the rest of the way to an edge (Figure 4.11(e)). The edges still act as hard boundaries to the region growing.

Given this set of proto-regions, any pair of neighbouring regions which abut each other at some point along their shared boundary (i.e. they are not completely separated by an original image edge) are then considered for merging. Following the lead of Sinclair, regions are merged if the difference in their mean colour is less than 10 in each colour component. This gives good results. Regions which are smaller than 10 pixels in size are also merged with the neighbour with the closest colour, as these regions will have very short boundaries and are unlikely to have a reliable edge labelling. These small regions also have very little visible impact on the final segmentation. Finally, the edge pixels themselves are assigned to the neighbouring region with the most similar colour. Figure 4.11(f) shows the final segmentation, with the regions coloured according to the mean pixel colour. The static segmentation takes about 2 seconds on a 300MHz PC for a typical image with  $352 \times 288$  pixels.

## 4.6 Labelling regions and finding the layer order

With the image edges labelled according to their motion probabilities, and the image pixels divided into regions along these edges, the complete motion segmentation of the image can now be produced. Determining the labelling of the image regions, and their relative depth ordering, is the second stage of the Bayesian framework introduced in Chapter 3. Together with the estimate of the motions, provided by the M-stage of the EM process (Section 4.4.2), this maximises the likelihood of the segmentation defined in (3.4).

The labelling of image regions, and the depth ordering of the motion layers, is completely determined by the edge labels. Following the approach described in Chapter 3, possible region solutions are hypothesised and tested against the edge label probabilities and also the prior probability of that labelling configuration.

---

<sup>11</sup>The pixels within a distance  $\gamma$  can be easily identified by reference to the distance transform image which gives, for each pixel, its distance from the closest edge.

### 4.6.1 Region probabilities from edge data

Given a hypothesised region labelling  $\mathbf{R}$  and the layer order  $\mathbf{F}$ , the edges can all be given a definite label by following the Labelling Rule from Section 3.3.3.<sup>12</sup> Under two motions, all edges which form the boundary of a region labelled with the foreground motion should move with that motion, and all other edges move with the background. The probability of the region labelling given the data (term (a) in (3.11)) is given by the probability of the edges having these implied edge labels.

A single image edge may form the boundary to several different segmented regions, but according to the theory developed in Chapter 3 a region should only consider the section of edge forming the region boundary. Fortunately this is easy since, by assuming that the edge sample points are independent, the motion probability for a section of an edge is simply the product of the sample points along that section.<sup>13</sup> This independence assumption makes this part of the region labelling trivial, since the evidence for a region is the product of the probabilities of the sample points on its boundary. Some regions may be bounded by no image edges, and hence have no sample points. In this case the labelling of the region is ambiguous, and will be entirely determined by the region prior. A complete hypothesised region labelling  $\mathbf{R}$  (given  $\mathbf{F}$ ) determines a labelling for all the sample points and so the evidence for this region labelling is thus the product of their probabilities under these implied motions.

### 4.6.2 Region prior

The labelled edges frequently contain a number of uncertain edges, or outright outliers, and relying on the edge labels alone produces a relatively poor segmentation (see Figure 4.12). The performance of region (and pixel) labelling algorithms can be greatly enhanced by remembering that not all labellings are equally likely. Particularly, objects are expected to have some spacial coherency—the regions or pixels belonging to an object are usually all together in one particular area of the image. Using a static segmentation enforces this to some extent, but regions with the same label are also, *a priori*, expected to be spatially coherent. This prior knowledge is encoded in Term (3.11b) of the treatment in Section 3.5.3.

The acknowledged means of modelling spatial coherency is with a Markov Random Field (MRF) [36, 59], which is often used in pixel-based motion segmentation methods [14, 40, 107, 160]. Here, the prior probability of a pixel's labelling depends

<sup>12</sup>The layer to which an edge belongs is that of the nearer of the two regions which it bounds'.

<sup>13</sup>The independence of sample points naturally also implies that the edge sections bounding a region are also independent.





Figure 4.12: *Region labelling solution with a flat prior.* If the region prior is constant (i.e. all configurations are equally likely), the regions are labelled from only the edge motion probabilities and the solution is fragmented.

on its immediate neighbours (either in a 4- or 8-connected sense). In the case of segmented image regions there is no regular grid as with pixels, and instead it is chosen here to consider neighbours in terms of the fractional boundary length,  $f_i$ . This is the length of the region's boundary which adjoins regions with the same labelling. When parameterised by this, the more of a region's boundary that adjoins a region of a given motion, the more likely the region is to also obey that motion.

The prior model has been estimated from thirty examples of correct (hand-labelled) region segmentations, a subset of the test sequences seen in Appendix D. Figure 4.13 shows the observed distribution. It is found that 77% of all regions are entirely surrounded by regions with the same labelling as themselves.

If the observed distribution is denoted by  $P(f_1|R=1)$  (where  $f_1$  is the boundary fraction adjoining motion 1 regions) the posterior probability that a region should be labelled motion 1, given  $f_1$ , is given by Bayes' Rule as

$$\begin{aligned} P(R=1|f_1) &= \frac{P(f_1|R=1)P(R=1)}{P(f_1|R=1)P(R=1) + P(f_1|R=2)P(R=2)} \\ &= \frac{P(f_1|R=1)}{P(f_1|R=1) + P(1-f_1|R=1)} \end{aligned} \quad (4.38)$$

assuming each model to be equally likely. This posterior probability is shown in Figure 4.14, and is well-modelled by a sigmoid with a function

$$P(R=i|f_i) = \frac{0.932}{1 + \exp(18(f_i - 0.5))} + 0.034. \quad (4.39)$$

as overlaid on Figure 4.14. There are special cases when  $f_i = 1$  or 0 i.e. when the region is completely surrounded or isolated. In these cases the posterior probabilities are chosen to be those given from the test data: 0.9992 and 0.0008 respectively.

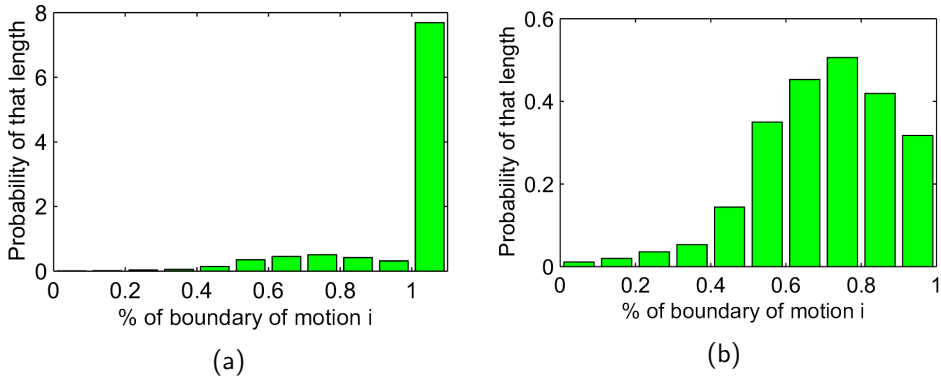


Figure 4.13: *Region statistics*. Distribution of regions with different boundary lengths  $f$ . (a) Whole probability density function; (b) a close-up of the cases where  $f < 1$ .

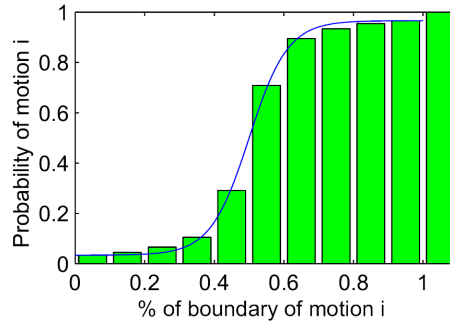


Figure 4.14: *Region prior*. Probability that a region is motion  $i$  given that a fraction  $f_i$  of the boundary that is also motion  $i$ .

Values of 1.0 and 0.0 were considered, which would disallow disconnected regions, or single-region holes in objects. However, these could interfere with the annealing process (described below), which requires the optimisation process to be able to pass through a lower (but non-zero) probability configuration.

### Non-symmetric priors

The region prior considered here is a very simple implementation, and it could be more sophisticated. It considers the labelling of both foreground and background regions equally but, as was pointed out in Chapter 3, the labelling of foreground and background regions is non-symmetric. Improved labelling performance could be obtained by using different priors for foreground and background regions. For example, it is very rare to find a hole in a foreground object, while there is almost always a ‘hole’ in the background, where it is occluded. The region label priors, assumed equal in (4.38), are also arguably different, since the foreground object is usually smaller (although this is no guarantee that it has fewer regions).

- For each possible layer ordering :
  - Initialise region labelling
  - Refine by simulated annealing (see Table 4.9)
- Select most likely segmentation over all layer orderings

Table 4.8: *Optimisation of region labelling and layer ordering.* Finding the most likely layer ordering  $\mathbf{F}$  by an exhaustive search, and the most likely region labelling  $\mathbf{R}$  (for each layer ordering) by simulated annealing.

This non-symmetric form of the priors has been tested for the case of two motions, but it is not obvious how it could be easily extended to an arbitrary number of motions (as is required for Chapter 7). The simpler prior described earlier does not give significant loss of performance in the majority of cases, and is applicable to a larger number of motions.

### 4.6.3 Solution by simulated annealing.

Finding the maximum likelihood labelling for the regions  $\mathbf{R}$  and the layer ordering  $\mathbf{F}$  is performed in two stages, outlined in Table 4.8. With two motions there are only two possible layer orderings and so an exhaustive search of these is possible: given a fixed  $\mathbf{F}$ , the region labelling  $\mathbf{R}$  may be maximised, and the most likely  $\mathbf{R}$  over all values of  $\mathbf{F}$  is the global maximum. Figure 4.15 shows the maximum likelihood region labelling given each of the two possible layer orderings, and in this case the posterior probability of (a) is much higher, indicating that this is the correct layer ordering and the best segmentation.

The maximisation of  $\mathbf{R}$  given  $\mathbf{F}$  (and the rest of the data) is a different matter, since the search space is combinatorial in the number of regions and there are no obvious polynomial solutions. Iterative schemes, performing a search of this space, are the usual solution in these cases. Starting with some initial guess, the solution is perturbed and is updated if the perturbation is an improvement, a process known as *stochastic relaxation* [59].

As with most iterative schemes, this suffers from the problem that it might find a local minimum. An improvement to this scheme is to provide the algorithm with the possibility of accepting a less favourable solution, subject to a small probability. In this way, given enough time, the algorithm will jump out of local minima and will find the global solution with a probability approaching one. To enforce convergence

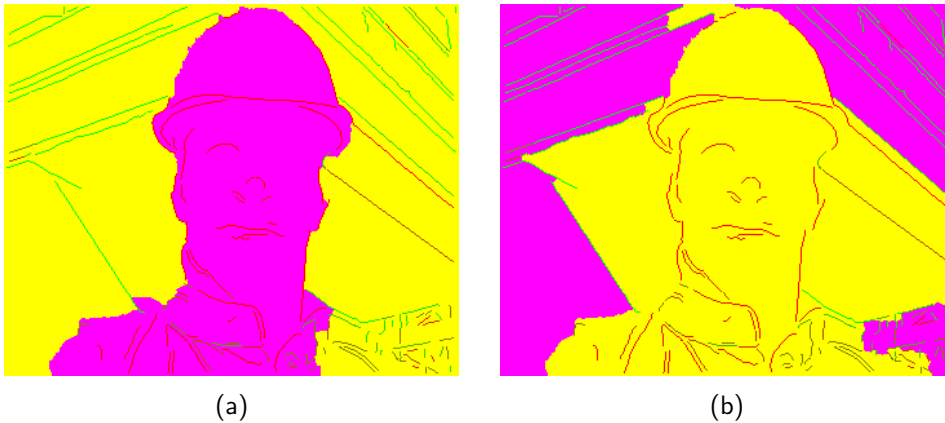


Figure 4.15: *Solutions under different layer orderings.* The most likely region labellings, showing the foreground as magenta and the background as yellow. (a) where red is the foreground motion; (b) where green is the foreground motion. Case (a) has a higher posterior probability, and so is the maximum likelihood segmentation over  $\mathbf{R}$  and  $\mathbf{F}$ .

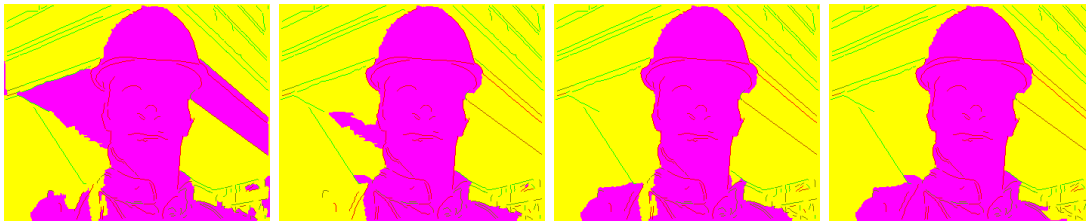


Figure 4.16: *Region labelling as simulated annealing converges.* Initial (heuristic) region labelling and then every fifth iteration of the annealing process. In this case, after fifteen iterations the labelling is a good as it will get.

the probability of a retrograde step is reduced over time. This is referred to as an *annealing schedule*, by analogy to the cooling of metals which occurs in a similar manner, and the approach is called *simulated annealing* [59, 88].

An initial guess for  $\mathbf{R}$  may be made by looking at the edge probabilities. Labelling each region according to the majority of its edge labellings is found to be a reasonable initialisation (see the first image in Figure 4.16). This solution must then be perturbed in an attempt to find a better labelling. The label for a particular region is only dependent on the local neighbourhood (the edges of the region and the neighbouring regions, via the MRF-style prior), and so the effect of perturbing a single region is only local. It is therefore possible to consider each region label in turn, maximising its own labelling probability given the current state of the other region labels.

The annealing simulation is achieved by considering the probability of a particular region label. When considering the relabelling of a region, the probability that it belongs to motion one may be calculated, and also for motion 2. This is the

- Initialise region labelling  $\mathbf{R}$
- For iteration  $n = 1 \dots 40$ 
  - Shuffle region order
  - For each region  $R_i$ 
    - \* Calculate (un-normalised) probability  $p_j$  of each labelling  $j$

$$p_j = P(R_i = j | f_j) \prod_{\mathbf{e}} P(\mathbf{e} = \text{Imp}(j, \mathbf{R}) | \mathbf{D}_1, \mathbf{D}_2) \quad (4.40)$$

- \* Cool probabilities

$$p'_j = p_j^{1+(n-1)^{0.07}} \quad (4.41)$$

- \* Normalise probabilities

$$P_1 = \frac{p'_1}{p'_1 + p'_2}$$

- \* Sample randomly  $u \sim \mathcal{U}_{[0,1]}$
- \* Set

$$R_i = \begin{cases} 1 & \text{if } u < P_1 \\ 2 & \text{otherwise} \end{cases}$$

Table 4.9: *Simulated annealing*. Optimisation of the region labelling  $\mathbf{R}$  in order to find the maximum likelihood labelling (given a particular layer ordering). The probability of a region's labelling is given by the product of an MRF-style prior and its edge probabilities. The edge probabilities are a function of the hypothesised labelling  $j$ , and the (fixed) labels of the other regions,  $\mathbf{R}$ , given by  $\text{Imp}(j, \mathbf{R})$ .

product of the MRF-style prior of the region having this labelling and the implied edge (sample point) probabilities. Given the labelling probabilities for this region, the region is assigned a definite label by a Monte Carlo approach, i.e. randomly according to the two probabilities. For the initial iterations this labelling is performed strictly according to the region probabilities, but as the iterations progress these probabilities are forced to saturate so that gradually the assignment will tend towards the most likely label, regardless of the actual probabilities. The saturation function, determined empirically, is

$$p'_j = p_j^{1+(n-1)^{0.07}} \quad (4.42)$$



Figure 4.17: *Probabilistic region labelling*. It is possible to label regions with probabilities, rather than a definite labelling. Due to various independence assumptions, saturation of region probabilities occurs, and the result is very similar to the deterministic labelling (compare with Figure 4.15(a)).

where  $n$  is the iteration number. This is applied to each of the estimated label probabilities as outlined in Table 4.9. The saturation function has been devised such that after around thirty iterations, all but the most balanced regions will be assigned their most likely label.

The annealing process continues for  $N$  iterations which, using  $N = 40$  is quick while being sufficient for a good solution to be reached. Each pass of the data tries flipping each region, but the search order is shuffled each time to avoid systematic errors. The entire maximisation over  $\mathbf{R}$  and  $\mathbf{F}$  (i.e. annealing twice), takes around two seconds on a 300MHz PC for a typical image with around 300 regions.

#### 4.6.4 A word on probabilistic region labelling

It is also possible to label regions with a probability, rather than a definite label, within the same simulated annealing framework. However, when it comes to calculating the new MRF-style prior for a region, this must consider each possible combination of the neighbouring regions, which is combinatorial in the number of neighbours. With some large regions having up to thirty neighbouring regions, it is infeasible to consider each of the  $2^{30}$  combinations and a linearising approximation must be made, using the *expected* fractional boundary length.<sup>14</sup> This, together with the independence assumptions already made earlier, leads to extensive saturation of the probability distributions. The process also takes much longer to converge, and so for efficiency it is initialised from the deterministic labelling. Figure 4.17 shows the probabilistic labelling, and it can be seen that, because of the saturation, it does

<sup>14</sup>*Loopy belief propagation* [104] may provide an efficient alternative approximation to this problem.

not diverge much from its initial labelling (compare with Figure 4.15(a)). A deterministic labelling, as already described, is fast and suitable for many applications.

## 4.7 Summary

This chapter has presented an implementation of the edge-based segmentation scheme outlined in Chapter 3. Edges are found using the Canny edge detector and tracked into the next frame. The motion of the edges is estimated by a group-based scheme which uses the motion along edge normals. Both motions are estimated simultaneously within an Expectation-Maximisation loop. This also provides the information fundamental to this scheme: the motion probability of each edge. A fast region-growing segmentation of the frame is performed based on these edges. Different labellings for these image regions are hypothesised in order to find the most likely labelling. Simulated annealing is used to quickly search an appropriate subset of the region possibilities. This maximisation is performed over all possible layer orderings to find the correct foreground motion.

The segmentation of a typical frame ( $352 \times 288$  pixels) into two motions, based on the motion to the next frame, takes around 8 seconds on a 300MHz PC. This implementation has been extensively tested, and evaluation of its performance is presented in the next chapter.





# Evaluation

---

## 5.1 Introduction

The implementation presented in Chapter 4 has been tested on a wide range of real video sequences, and the test results are presented here. In total, thirty-four test sequences are considered and the results from the complete set of sequences can be seen in Appendix D. This chapter presents detailed results for four of the test sequences, and then discusses the performance on the test set as a whole. Finally, the results of this novel framework are compared with results presented by other authors.

## 5.2 Test sequences

The development of this implementation has focussed on four sequences. In particular, examples from the **Foreman** sequence have been used throughout the body of this dissertation, and further results are presented here. Detailed results are also presented from two other common test sequences: **Coastguard** and **Tennis**. These three are among the sequences widely used by authors for testing video segmentation and coding applications. Several other test sequences have been recorded using a hand-held video camera. One of these, the **Car** sequence, has also been extensively used for testing because of the unusual features it exhibits, such as the background being visible through the window of the car. This is the fourth sequence considered

in detail.<sup>1</sup>

A large supply of other test sequences has kindly been provided for this work by AT&T Laboratories, Cambridge, derived from their AT&TV project [98]. This project maintains an archive of the past seven days' television across the four main UK terrestrial channels, and is used for investigations into large-scale information retrieval. Videos in the AT&TV system are stored in the MPEG-1 format and a selection of twenty-five MPEGs from this archive have been used for testing the two-motion implementation. Sequences were selected at random from programmes shown in February 2001, covering many different genres. In order to agree with the two-motion assumption, sections were chosen which, by eye, had only one foreground object.

### 5.3 Qualitative and quantitative results

Segmentation is a subjective procedure, and the desired results are often determined by the semantics of the scene, and the use to which the information will be put. For example, a person waving their arms may need to be segmented as one object (for background replacement), or the arms may need to be treated separately (for gesture recognition). As a result, there is no accepted method of assessing the quality of a segmentation.<sup>2</sup>

In the results presented here, the qualitative appearance of the segmentation is discussed. The quantitative segmentation performance is also measured by comparing it with a hand-labelling of the same static regions and edges. This gives some measures, such as the percentage of edges or pixels correctly labelled, which may be compared between segmentations. In Section 5.9, a qualitative comparison is made between the results of this dissertation and those from other authors.

All the segmentations, unless stated, use exactly the same parameter values, and the affine motion model is used throughout.

### 5.4 Foreman sequence

The Foreman sequence (Figure 5.1) is a sequence examined by many authors (e.g. [91, 103, 153]). Taken with a hand-held camera, it shows a man in a hard hat talking

<sup>1</sup>These test sequences, and a selection of others, are available for download from <http://www-svr.eng.cam.ac.uk/~pas1001/Publications/videos.html>.

<sup>2</sup>A recent paper by Martin et al. [95] presented a database of hand-segmented images. It was concluded that human segmentations are *consistent*, but that different observers choose to segment at different levels of granularity (of which the 'person' vs 'arms + head + body' is an example).



Figure 5.1: *Foreman sequence*. Frames 47–51 from the Foreman sequence. The foreman moves his head to the left during this part of the sequence.

animatedly to the camera. The section shown in Figure 5.1 is a rather less animated portion (and is the section considered by most authors); later parts of the sequence are very difficult to describe using only two affine motion models (see Chapter 7).

### Edge detection

Figure 5.2(a) shows the edges detected in the first frame of the sequence. It can be seen that almost all of the occluding boundary is included among these 153 edges, the only major absences being parts of the hat, and his right shoulder. The shoulder is missed because only a grey-scale edge detector is used and the intensities of the shoulder and the concrete background are very similar. Even with a colour edge detector, parts of the hat would still be missed as both are nearly white.

### Edge labelling

These edges are tracked into the next frame and labelled as motion 1 or motion 2 using the Expectation-Maximisation (EM) algorithm. The motion between the frames is negligible for the background and about two pixels for the head. This is well within the range of the search track, but is large enough for a clear differentiation to be made between the two motions. The EM stage reaches its convergence criterion after sixteen iterations (see Figure 4.10 for the first few iterations). This motion estimation/edge labelling stage takes about three seconds.<sup>3</sup>

The final edge probabilities are shown by the edge colours in Figure 5.2(a). In these figures, each edge is coloured as a blend between the two probabilities, where red is motion 1 and green is motion 2.<sup>4</sup> In this case, the edge probabilities are generally very good, with only a few errors or ambiguities. The edges on his left shoulder have equal probabilities under each motion (and so appear a dark yellow colour). In this area of the frame the two estimated motions are very similar, so it is impossible to determine a labelling from the motions (see Figure 3.2 for the

<sup>3</sup>All timings in this chapter are quoted for a conventional 300MHz PC.

<sup>4</sup>That is, for a 24-bit colour, the value (R,G,B) displayed is  $(255 \times P(\text{motion 1}), 255 \times P(\text{motion 2}), 0)$ .

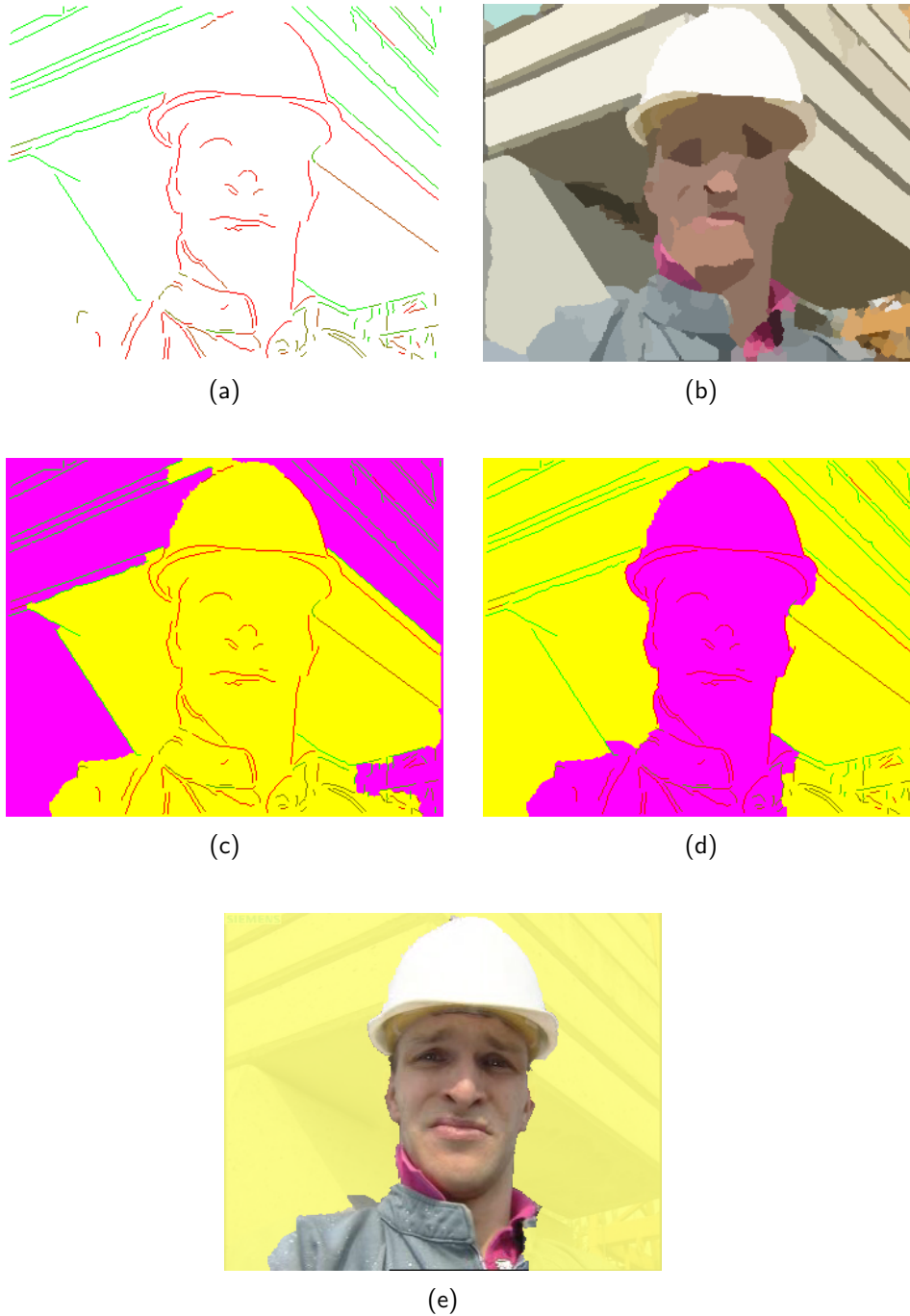


Figure 5.2: *Foreman segmentation from two frames.* (a) Edges labelled by their motion; (b) Region segmentation; (c), (d) Region labellings under alternative layer orderings. (d) is the most likely with a probability of over 99.99%; (e) Final foreground segmentation. Some images here were previously shown in Chapter 4.

full extent of the ambiguous region). The background edges above his left shoulder are mislabelled since in this part of the image, motion 1 (red) is along the line of these edges and so again they can fit either motion. A few slightly better matches under motion 1 leads to an erroneous decision in favour of this motion, with edge probabilities of around 80%. In total, 78% of edges are labelled correctly, compared with a hand-labelling of the same edges.

### Static segmentation

Figure 5.2(b) shows the static segmentation of this frame. It can be seen that, even though the edge of the hat is incomplete in the detected edges, it is correctly extracted in the static segmentation, which does detect enough difference in colour. His right shoulder is not fully segmented (it should extend all the way to the edge of the frame) because the colours are too similar. As a result, it will be impossible to accurately represent this shoulder in the final segmentation. Without an edge in the image, these problems can only be resolved by some higher-level modelling.

### Region labelling

The final stage is the labelling of the regions according to their motion. This process is performed twice, once for each possible layer ordering, using simulated annealing. Figure 5.2(c) shows the final solution assuming that motion 1 is foreground, and Figure 5.2(d) the solution if motion 2 were foreground. In each case, a region is coloured magenta if it is labelled as foreground, and yellow if background. It can be seen that, despite the occasional poor edge label, realistic segmentations are produced thanks to the use of the MRF-style prior on region labels. The two minimisations take a total of three seconds.

The two possible solutions have likelihoods (from the edge probabilities and MRF prior) of  $e^{-421}$  and  $e^{-411}$  respectively, giving a probability of over 99% that motion 1 is the foreground layer, and so the final segmentation is that shown in Figure 5.2(e). Of the 221 regions identified by the static segmentation, 212 are labelled correctly (compared with a hand-labelling). As a percentage of the pixels in the image, this is 97.6%, which is excellent. The complete segmentation process takes a total of eight seconds.

## 5.5 Tennis sequence

Figure 5.3 shows the second of the test sequences, another standard sequence (studied, for example, in [4, 47, 103]). In this part of the sequence, the player bounces the ball on his bat as he prepares to serve. The detected edges are shown in Figure 5.4(a), and it can be seen that the edge detection settings used throughout this work have a high enough threshold to avoid the textured background, but extract all of the occluding boundary. In total, sixty-seven independent edges are detected.

Figure 5.4(a) also shows the edge probabilities after the motion estimation stage, which are reasonable, with 88% of edges correct. The table and the bat have been identified as two distinct motions, but the labelling of the arm is uncertain. The upper arm is almost stationary, and the lower arm naturally obeys a motion part-way between that of the upper arm and the bat, so an uncertain labelling is somewhat justified. The motion of the ball is, of course, a genuine third independent motion. However, the ball's displacement between frames is quite large—about ten pixels—and with only one edge for that object, the measurements from any sample points which do find a match are swamped by the other motions. Consequently, the ball's motion is ignored and the ball's edge is labelled at close to 50% for each motion.

Despite having an almost complete occluding boundary, this does not guarantee a trouble-free region segmentation. The background is not uniform, but with few edges detected on the background only a small number of seed points are created for region growing. The initial region growing stage is controlled by colour, and with the textured background giving some colour variation, each seed region is only small. When the blind region growing stage is then performed, it takes a substantial number of steps to reach the edge of the arm and if there are any small gaps in the arm's edges, the regions inside the arm (which also try to grow) may bleed out into the background. It is for this reason that the two-stage region growing scheme outlined in Section 4.5.2 was introduced, where the first stage of growing stops a few pixels from the edge and then proceeds from there in a separate stage. This effectively blocks small gaps between edges and means that in this case a very good static segmentation is produced. This is shown in Figure 5.4(b), and contains seventy-two regions.

The two possible region labellings, one for each layer ordering, are shown in Figures 5.4(c) and 5.4(d). It can be seen that the brown background is clearly identified as background in both cases (it has edges of both motions), and the decision is between the table and the bat and arm being foreground. Even though there are no T-junctions between edges of the two motions in the detected edge



Figure 5.3: *Tennis sequence*. Frames 1–5 from the Tennis sequence. The table tennis player is bouncing the ball on his bat during this part of the sequence.

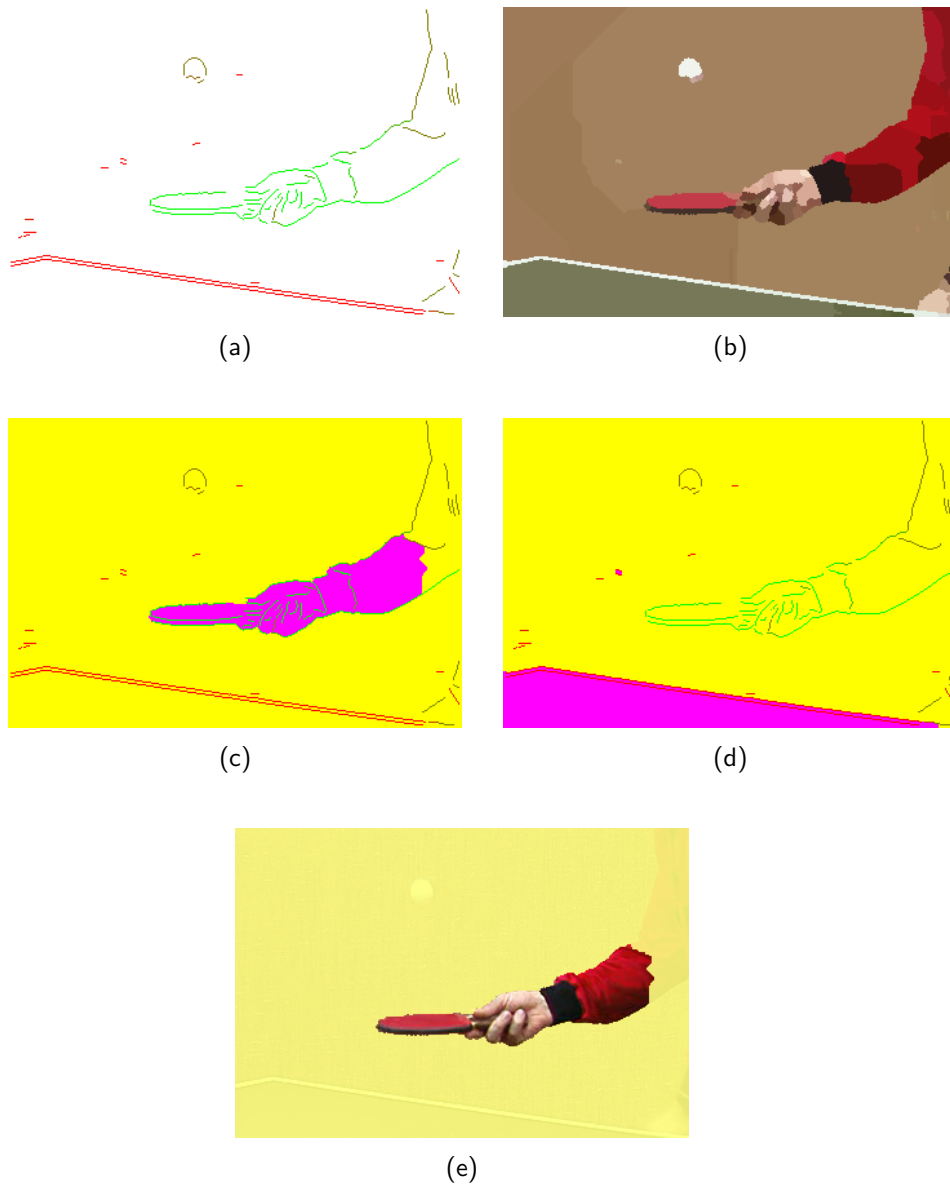


Figure 5.4: *Tennis segmentation from two frames*. (a) Edges labelled by their motion; (b) Region segmentation; (c), (d) Region labellings under alternative layer orderings. (c) is the most likely with a probability indistinguishable from 1; (e) Final foreground segmentation.

map, there *are* T-junctions between the region edges, and these have the same effect. In this case, it has already been noted that the brown regions must obey the background motion. The important difference between the two hypothesised labellings is the few edges on the background, which are labelled as obeying the red motion with a high probability. Unless red is the background motion, these will be mislabelled. As a result, green is correctly identified as the foreground, with a probability indistinguishable from 1. The final segmentation is clean and accurate, only missing the (admittedly stationary) upper arm.

With far fewer edges and regions than the previous **Foreman** case, the whole process is much faster. The complete segmentation takes about five seconds.

## 5.6 Coastguard sequence

The third standard test sequence presented here is the **Coastguard** sequence, shown in Figure 5.5, and also considered in [47, 110]. This is one of the more difficult sequences to segment: the movement of the water makes it difficult to track; the hull of the boat is a similar colour to the water, making the boundary difficult to identify; and there is also a significant amount of fine detail in the boat's mast and railings. The boat moves from left to right, and is tracked by the camera.

There are 316 edges detected in this frame, far more than in the previous two sequences, with quite a few short edges representing parts of the rocks along the side of the waterway, or waves (see Figure 5.6(a)). With the similarity of colour between the hull and the water, neither the prow nor stern of the boat is extracted as an image edge. This causes problems later when it comes to finding image regions and labelling them. In particular, the static segmenter merges the prow with the water, and the stern with the wake in Figure 5.6(b). The intricate mast is not segmented at all, but the railings at the bow are well segmented. A region-based approach is not a good technique if fine detail is required, and to preserve these a pixel-by-pixel refinement and labelling would be needed.

The EM stage converges quickly, in thirteen iterations taking two seconds, but gives a noisier solution than has been seen so far (Figure 5.6(a)). In this sequence there are many more background edges than foreground, and the foreground is concentrated in one small area of the frame. With this arrangement, it is quite possible for some background edges to agree with the foreground motion by chance, and the foreground motion can conform to these edges without significantly disturbing the labelling of the few foreground edges.





Figure 5.5: *Coastguard sequence*. Frames 256–260 from the Coastguard sequence. The camera tracks the boat as it sails from left to right.

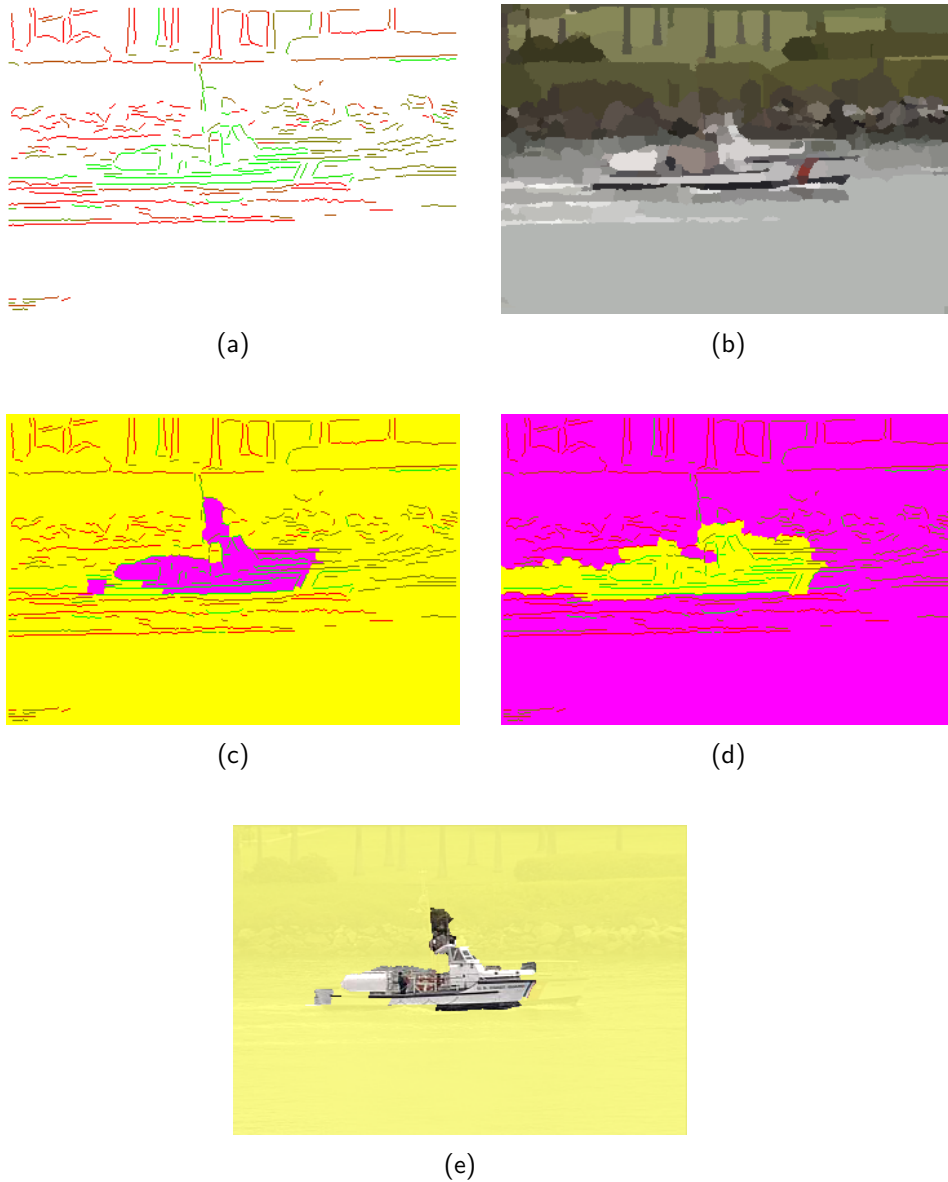


Figure 5.6: *Coastguard segmentation from two frames*. (a) Edges labelled by their motion; (b) Region segmentation; (c), (d) Region labellings under alternative layer orderings. (c) is the most likely with a probability of 99.18%; (e) Final foreground segmentation.

This example highlights the effectiveness of the MRF-style prior on the region labelling, as even with this noisy edge labelling both hypothesised region labellings are plausible and realistic. With more regions, 276, than either previous example, the simulated annealing process takes longer, converging on the two solutions in a total of three seconds. Looking at the two possible solutions, most of the erroneous edges are equally incorrect under either possible ordering (for example, the green edges on the tree trunks at the top of the frame). The layer ordering therefore depends only on the regions and edges around the occlusion boundary. The first solution, Figure 5.6(c), with motion 2 (green) as foreground is comfortably identified as the most likely.

The final solution, Figure 5.6(e), is a reasonable attempt given the static segmentation. A hand-labelling of the same image regions would only have labelled 18 (6%) of those differently, but in either case some quite substantial parts of the hull of the boat are missing; namely those which the static segmentation merges with the water. This problem is one which cannot be resolved without detecting or otherwise perceiving the edge of the boat. Refining the static segmentation to find these edges is a matter for future work, as discussed in Chapter 8. It should also be noted that this sequence violates the layer assumption—the water along the side of the boat, part of the ‘background’, is nearer the camera, and so the interface between the hull and the water is a background edge which occludes the foreground object. The *Coastguard* is a testing sequence for a segmentation scheme, and the edge-based scheme performs creditably.

## 5.7 Car sequence

Figure 5.7 shows the *Car* sequence, specifically filmed for this work using a hand-held MPEG-1 video recorder. It shows a close-up of a car being tracked as it drives to the left. This sequence exhibits several unusual features: the foreground object occupies the majority of the pixels (i.e. the ‘dominant motion’ is not necessarily the background); the background is visible through the car window; and there are reflections on the top and bonnet of the car. The system presented here deals admirably with the first two of these, although it falls down on the latter.

Figure 5.8(a) shows the detected edges, labelled according to their motions as before. In this case the EM algorithm used to estimate the motions takes more than 100 iterations (nine seconds) before it converges. The problem here is that the background motion is considerable (of the order of ten pixels), and is confined solely to the top of the frame. The two random motions are both initialised to give a



Figure 5.7: *Car sequence*. Frames 490–494 from the Car sequence. The camera tracks the car as it moves to the left.

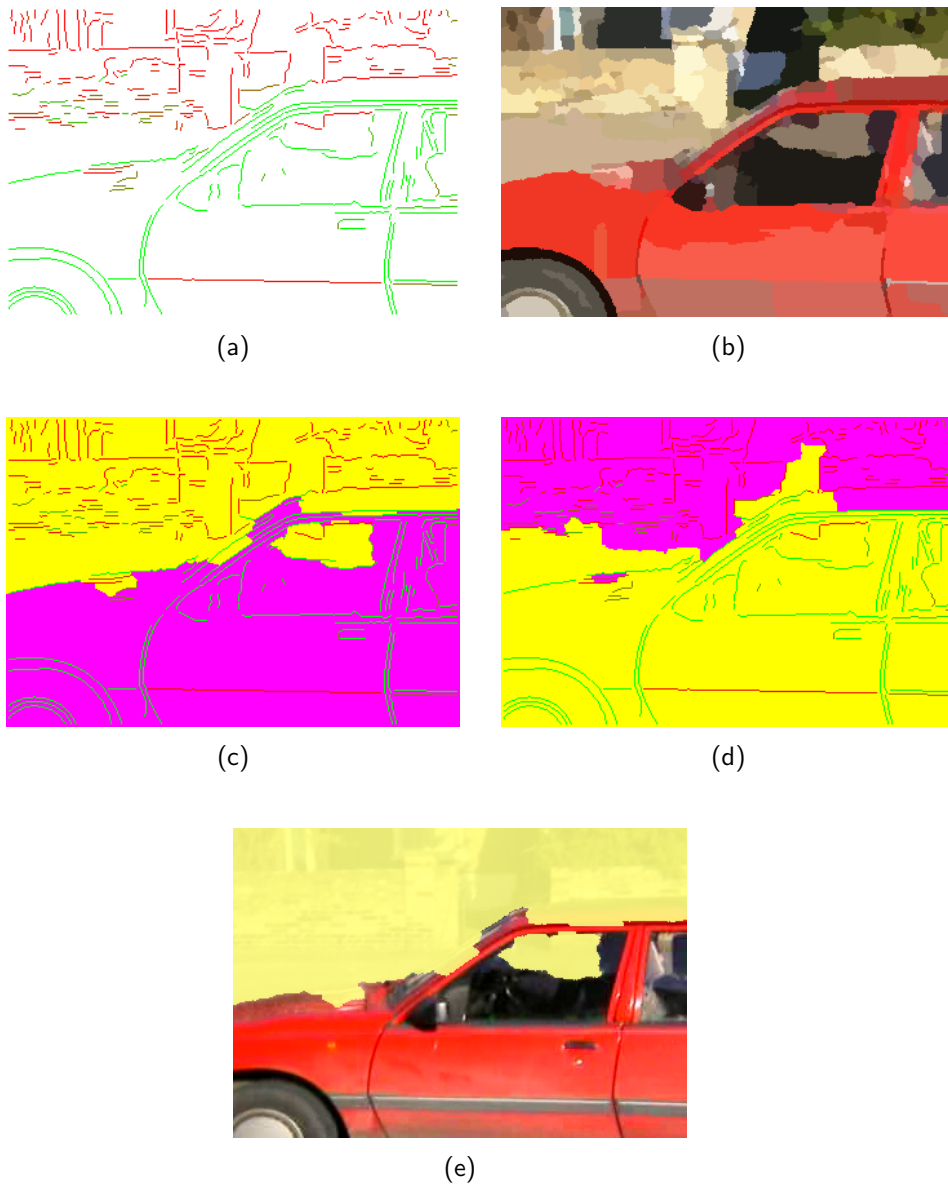


Figure 5.8: *Car segmentation from two frames*. (a) Edges labelled by their motion; (b) Region segmentation; (c), (d) Region labellings under alternative layer orderings. (c) is the most likely with a probability of 99.63%; (e) Final foreground segmentation.

reasonable fit to the entire of the frame, and so both contain significant shear terms (a deformation which is allowed under the affine model). These two initialisations only slowly diverge to give the two correct motions. Convergence is also particularly difficult since many edges are horizontal, and with both motions being horizontal the labelling of these edges is uncertain for much of the process, and frequently oscillates between the two possible motions as they are refined. These edges should be labelled with probabilities close to 50%, but unfortunately their edge statistics saturate and they are labelled strongly in favour of one or the other. The non-independence of sample points is investigated in Appendix C, and improved statistical models are an issue for future work, as discussed in Chapter 8. Nonetheless, 91% of edges are correctly labelled, with the only major errors being the top of the car and one edge along the side (which does not affect the region labelling). The error on the top of the car is the result of the background reflections on the roof, which naturally move with the background. Thus this edge in fact has the *correct* motion labelling, but the incorrect semantic labelling. The small edges on the car bonnet are also labelled according to background reflections. Without any higher-level processing, namely a model of a car's shape, this problem is difficult to resolve and is not discussed further here.

The static segmentation, Figure 5.8(b), is again good. Of the two layer ordering solutions, Figure 5.8(c) is the most likely, but with less certainty than in some sequences (although still high, because of the edge saturation). Despite the texture in parts of the background, there are in fact very few T-junctions, and the mislabelling of the occluding boundary on the top of the car also does not help the layer ordering choice. However, the correct layer ordering *has* been determined, in a case where (due to the relatively small area of background) a naïve ‘dominant motion’ approach might fail.

The final foreground segmentation is shown in Figure 5.8(e), where 96.2% of the pixels are correctly labelled. Because a background edge is visible through the window of the car, the window regions are correctly labelled as background by the logical constraints, which is particularly pleasing. The only errors are in the areas where reflections are present.

## 5.8 Ensemble results

The implementation of Chapter 4 has been tested on a total of thirty-four image sequences: the four sequences described above; two other standard sequences; three ‘home movies’; and twenty-five sequences from the AT&TV archive (see Figure 5.9



Figure 5.9: *Examples from the AT&TV sequences.* A selection of images from the sequences in the AT&TV archives in February 2001. Results from these sequences can be seen in Appendix D.

for some examples). Specific sequences from the test set are identified by name in this section, and the results for all the test sequences can be found (in alphabetical order) in Appendix D.

### Segmentation parameters

The system was left to segment each sequence automatically, using an affine motion model in each case. Apart from seven exceptions (21%), each segmentation used exactly the same parameters (those given in Chapter 4). The exceptions were

**Lower edge threshold (4 cases: Friends, ITN, Tennis2, Thunderbirds1)** In four sequences no edges were detected in the scene background using the standard thresholds. These thresholds are usually set deliberately high to avoid edges due to texture, but in these cases the absence of structure in the background meant that the texture had to be detected. A common cause of this is a small depth of field—with some lenses, particularly those with long focal lengths, the background is significantly out of focus and so sharp edges are not present. In these four cases the threshold was reduced by hand until edges were detected in the background.

**Frame subsampling (3 cases: Horizon1, ITN, News)** In three further sequences, all of which feature seated people talking to the camera (including two news-readers), the inter-frame motion was found to be too small for any independent motion to be detected. Edge matches are only found to the nearest pixel, and the motion in each case was less than this. To force a larger motion these sequences were subsampled, taking every 10th or every 20th frame. Subsampling was also required for the two sequences from the *FlashGordon* cartoon, since that had only been animated at 15 frames per second (fps) and so featured

repeated frames when broadcast at the UK standard of 25fps. Taking every second frame avoided these repetitions.

Both of these parameter corrections could, in a future system, be identified and made automatically. The edge detection process could benefit from an adaptive threshold which attempted to encourage a certain density of features. The second case (that of the motion between frames being too small) would be resolved if the number of motions was not constrained to be two between each frame, as frames with negligible motion would best modelled using only one motion. Alternatively, sample points with no motion could be flagged to wait to see if a motion does eventually occur.

### EM convergence

With these small parameter changes to a few sequences, each sequence was deemed to have sufficient edges, and motion, for the EM process to have a reasonable chance of success. In 82% of cases the EM did indeed reach a good solution. The cases where the EM does not satisfactorily converge are the result of:

**Non-affine motion (3 cases: Driven2, Horizon2, Nick)** In these cases the foreground motion cannot be modelled by an affine motion—either it is projective, or changing in 3D shape—and the deformation is too great for the sample points to give the correct statistics.

**Too many background edges (1 case: Buffy)** If there are far more background edges than foreground edges, the EM process (starting with a random labelling) can converge to a local maximum which includes some of the background edges with the foreground. (This was partially in evidence in the *Coastguard* sequence considered earlier in this chapter). In the severe case, both initial edge labellings will contain a predominance of background edges and EM will never converge correctly on the foreground motion.

**Too few background edges (1 case: ITN)** One sequence, even with a lower edge threshold, still had too few edges for a background motion to be estimated, for the same reasons as the reverse case described above.

**Motion too large (1 case: FlashGordon2)** One of the cartoons exhibits a foreground motion of 60 pixels, which is larger than the search track (and unrealistic in real video sequences).

The remaining twenty-seven cases produced reasonable motion estimates and edge probabilities. These cover a wide range of genres, and include some challenging subjects. In particular, the edge labelling for the running lion in **Cats1** is good, and that for the cat in the **Trin** sequence is also reasonable, both of which are examples of non-affine motion which were successfully tracked.

Of the cases where EM does not converge, the problem of large non-affine motions is the most significant. In one of those cases, **Horizon2**, the background motion could be modelled by a full 2D projective model, rather than the 2D affine used, but if this is tried it is found that the EM struggles to converge with these additional degrees of freedom. A multi-resolution approach, fitting first an affine and then a projective model might perform better here [9, 78]. The **Nick** sequence also features a projective deformation, where the subject tilts his head back violently, although this also suffers from having very few edges. In the other case (**Driven2**), the image motion is probably best explained by three motions, and this should be detected by a multi-motion approach, as introduced in Chapter 7.

Ensuring a suitable number of edges is something which could be accomplished by an adaptive approach, as described earlier, but it should also be remembered that in some sequences segmentation *won't* be possible. Likewise, if the motion is too large any tracking scheme will find difficulties.

### Layer ordering

Whenever EM converges to a reasonable solution, the region labelling is also good. This validates the fundamental assertion of Chapter 3, that an edge labelling is sufficient for a dense labelling of the frame. However, this is only true up to unresolvable ambiguities and, in particular, if the layer ordering is ambiguous the foreground layer can be incorrectly determined. Of the twenty-eight sequences where EM provides a good solution, the layer ordering is correctly determined in twenty-three cases (85%). The five sequences where the layer ordering is incorrect are:

**No T-junctions (3 cases: AHvid, Thunderbirds1, Tweenies)** Where the foreground regions do not interact with background edges, the layer ordering cannot be determined. Even where there are a few T-junctions, if the edges corresponding to these are poorly labelled then the layer ordering can be ambiguous. In one of these cases (**Thunderbirds1**), the edge labels are good, but there is genuinely no interaction between regions which can be labelled by these edges. In the other two cases, there are a few T-junctions, but the edge labelling is poor due to the foreground deforming significantly.

Ranking	Pixels correct	Frequency (%)	
Excellent	> 95%	11	(32%)
Good	85–95%	8	(24%)
Reasonable	75–85%	3	(9%)
Poor	50–75%	5	(15%)
Failure	0–50%	7	(21%)

Table 5.1: *Percentage of pixels correctly segmented using two frames.* Overview of segmentation performance over the thirty-four test sequences. Any figure over 85% is a good segmentation; those over 95% are almost flawless.

**Missing occluding boundary (2 cases: Food&Drink, Horizon1)** Where a substantial portion of an occluding boundary is missing, foreground regions can bleed into the background. This is in fact likely, since if there is no image edge, the foreground and background in this area have a similar intensity. A region spanning both foreground and background violates one of the assumptions (that each region obeys only one motion), and also means that a (partly) foreground region may be bounded by an edge obeying the background motion. As a result, the layer ordering is not well defined, and an erroneous ordering can occur if the edge labels are poor.

The first type of error is difficult to deal with over two frames—these cases truly are ambiguous unless more background edges can be detected or a better edge labelling determined. Both of these are possible if multiple frames are used. The second type can also sometimes be dealt with by a multi-frame approach which would eventually detect and maintain the missing occluding edges.

### Final segmentation

Table 5.1 summarises the segmentation results over the thirty-four test sequences. It shows the percentage of pixels correctly labelled compared with a hand-labelling of the same image regions. Any automatic segmentation which labels more than 95% of pixels correctly is almost indistinguishable from the ideal segmentation, and any figure over 85% is still very good. It can be seen from the table that over half of the test sequences fall into one of these top two categories, with virtually a third of sequences segmented almost flawlessly.

Of the twelve sequences which perform poorly (i.e. have less than 75% of pixels correct), ten have already been discussed: the EM stage failed on five sequences; and a further five had a reasonable edge labelling, but the wrong layer ordering. Two cases with a reasonable edge labelling and a consistent region labelling also scored poorly against the ideal labelling. In the **Friends** sequence only the actor’s



head moves, but the desired *semantic* segmentation is the whole actor. In the other case, the standard *FlowerGarden* sequence, the desired ‘clean’ segmentation is just that of the tree, but parts of the flowerbed move with a very similar motion and the automatic scheme picks these up as well. These two cases highlight the discrepancy that can occur between a pure *motion* segmentation and the semantic applications for which they are usually intended.

## 5.9 Comparative results

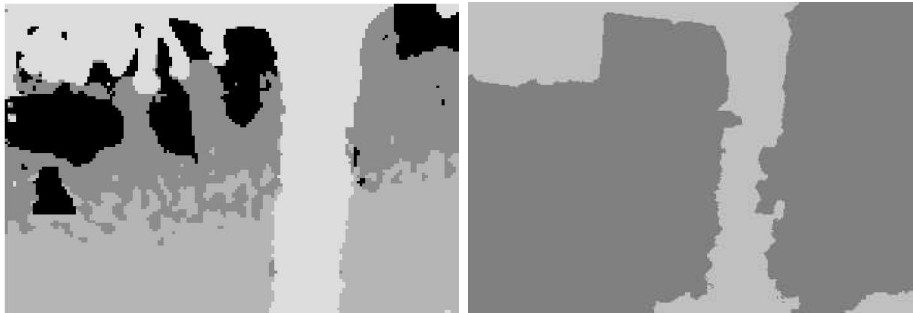
As demonstrated by the results shown in this chapter, motion segmentation is a difficult task. It is also difficult to assess, in quantitative terms, the accuracy of a segmentation, and many of the results presented here have been qualitative. It is therefore instructive to compare the results generated by this edge-based system with work published by other authors over recent years; this gives an indication of the relative success of the edge-based approach. Again, with no accepted quantitative measure of segmentation performance, a qualitative comparison is made between results.

This section presents a comparison with a number of authors who have analysed freely-available sequences. The results are extracted from their published papers, and comparable results from the implementation described in this dissertation are shown side-by-side. Each author displays their results differently and so, as far as possible, the results presented from the edge-based system have been generated so as to emulate each of their particular styles.

### 5.9.1 Pixel-based approaches

A popular test sequence amongst pixel-based authors is the *FlowerGarden* (see Appendix D), which is unsurprising as it contains a large amount of texture in the foreground objects. Pixel-based approaches rely on texture for an accurate motion estimation and pixel labelling. This sequence is unfortunately also one where the edge-based approach performs less well, because of the dominance of edges from one motion model (the flower bed), and the difficulty in extracting the edge of the tree against the flowers.

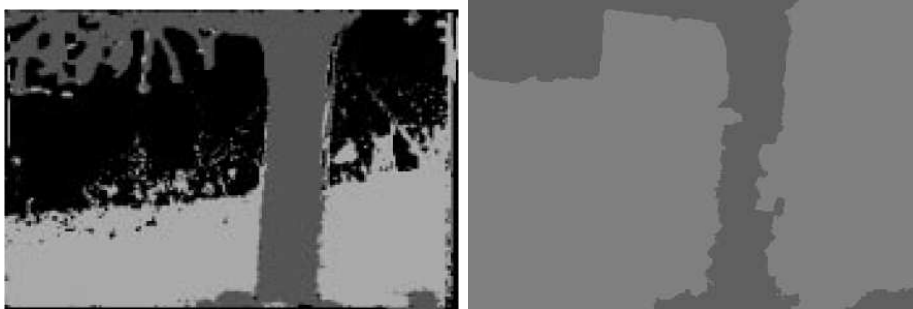
Wang and Adelson [159] presented results from this sequence in their paper introducing the layered representation, and Figure 5.10 shows a comparison with this. The edge-based approach extracts the tree’s edges more accurately along some of the trunk and main branch, but less well in other areas. The fine detail of the



Wang and Adelson (1994)

this dissertation

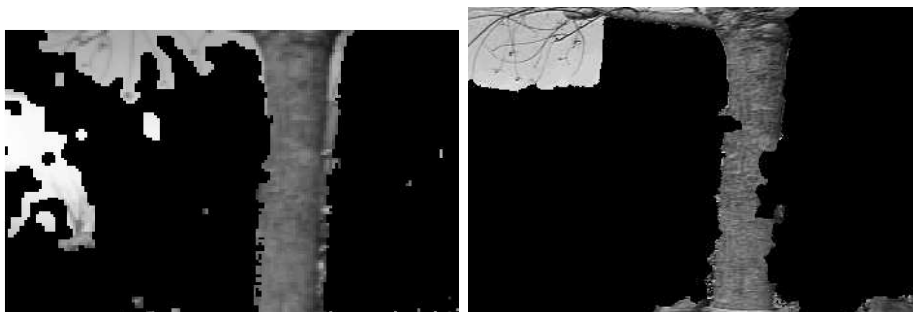
Figure 5.10: *Comparison with Wang and Adelson: FlowerGarden sequence.* A comparison with results presented by Wang and Adelson in [159]. The segmentation of the tree is comparable—Wang and Adelson estimate it to be too wide, while the edge-based approach misses a few sections.



Ayer and Sawhney (1995)

this dissertation

Figure 5.11: *Comparison with Ayer and Sawhney: FlowerGarden sequence.* A comparison with results presented by Ayer and Sawhney in [4]. Ayer and Sawhney's is a better outline, but there is more noise in the background.



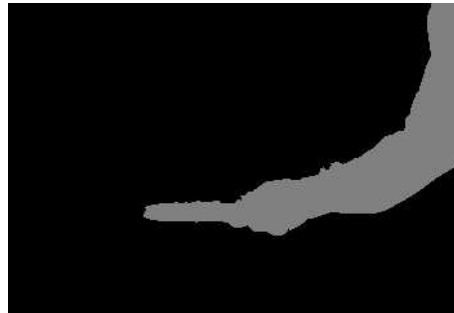
Weiss and Adelson (1996)

this dissertation

Figure 5.12: *Comparison with Weiss and Adelson: FlowerGarden sequence.* A comparison with results presented by Weiss and Adelson in [160]. The results are similar, but the edge-based approach is cleaner.



Ayer and Sawhney (1995)



this dissertation

Figure 5.13: *Comparison with Ayer and Sawhney: Tennis sequence.* A comparison with results presented by Ayer and Sawhney in [4]. Ayer and Sawhney's is much worse, with poor boundary localisation and large amounts of noise.

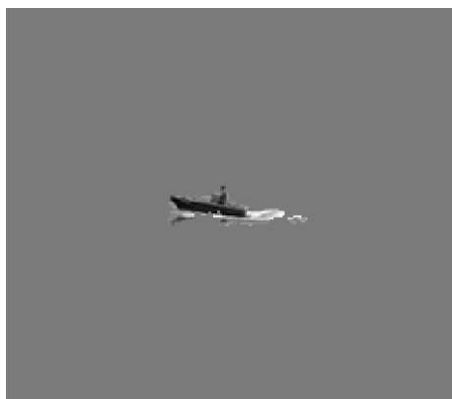


Elias (1998)



this dissertation

Figure 5.14: *Comparison with Elias: Tennis sequence.* A comparison with results presented by Elias in [47]. Both segmentations are excellent.



Elias (1998)



this dissertation

Figure 5.15: *Comparison with Elias: Coastguard sequence.* A comparison with results presented by Elias in [47]. Both segmentations are excellent, although Elias's approach finds a little more fine detail than is detected by the edge-based scheme.

small branches cannot be well represented by image regions, and these are segmented poorly. Comparisons with Ayer and Sawhney [4] and Weiss and Adelson [160] are also presented for this sequence (Figures 5.11 and 5.12 respectively). Both of these authors' results show some outlying pixels or regions which are absent in the edge-based approach, which gives the system presented in this dissertation a more pleasing appearance. The same is true, to a much larger extent, for Ayer and Sawhney's segmentation of the *Tennis* sequence (Figure 5.13), which contains a considerable number of erroneous pixels as well as giving the arm too large an extent. The edge-based approach is both more accurate and cleaner than their work.

Some of the best pixel-based work is that of Elias [47], who uses a multi-frame EM approach, also modelling pixel occlusion. Both segmentations of the *Tennis* sequence—Elias's and the edge-based approach—are almost flawless (Figure 5.14); they only disagree about the ambiguous upper arm. The finer detail available to pixel-based methods means that his segmentation of the *Coastguard* sequence (Figure 5.15) is slightly better, but the edge-based approach also performs very well on this difficult subject. Results are therefore comparable but, importantly, his segmentations take about 1 minute to perform, compared with a few seconds for the edge-based approach (on roughly comparable machines).<sup>5</sup>

### 5.9.2 Region-based approaches

Region-based approaches avoid the problem of occasional misassigned pixels, seen in some of the previous examples, by only labelling homogenous regions of the image. The segmentations produced by this approach are thus naturally cleaner, but depend to some extent on the quality of the original region segmentation.

The segmentation scheme used in this dissertation is clearly superior to that used by Moscheni and Dufaux in [103], which still somehow manages to give outlying pixels (Figures 5.16 and 5.17). The labelling of regions is also superior, particularly in the *Foreman* case, where their system has performed a number of incorrect merges. Although that particular pair of frames are difficult to segment (both schemes include some background), the edge-based system gives a more appropriate result.

Other work by Dufaux et al. [46] also performs some strange merges, and the edge-based approach here also appears superior since the complete bat and hand is segmented in the *Tennis* sequence (Figure 5.18). The edge-based approach decides that the upper arm is stationary between these two frames and so belongs to the

---

<sup>5</sup>Elias's figures are quoted for a 100MHz Sun SPARCstation, and the figures in this dissertation for a 300MHz PC.

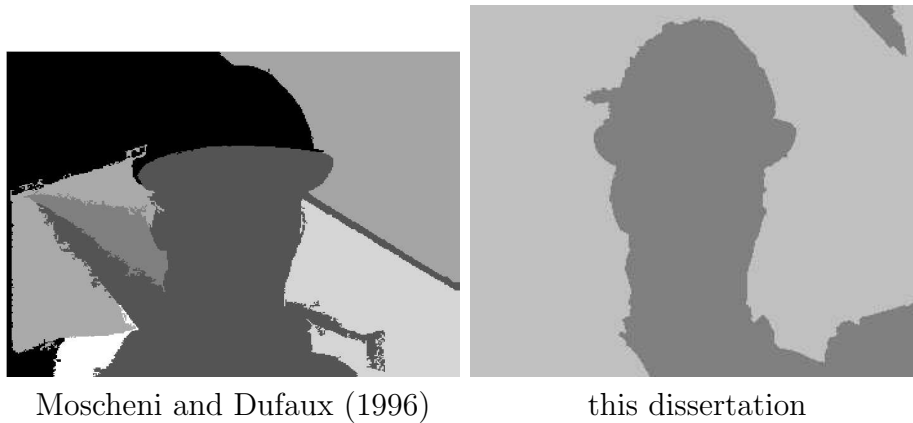


Figure 5.16: *Comparison with Moscheni and Dufaux: Foreman sequence.* A comparison with results presented by Moscheni and Dufaux in [103]. Both approaches merge some erroneous regions with the foreground, but the edge-based approach is considerably better.

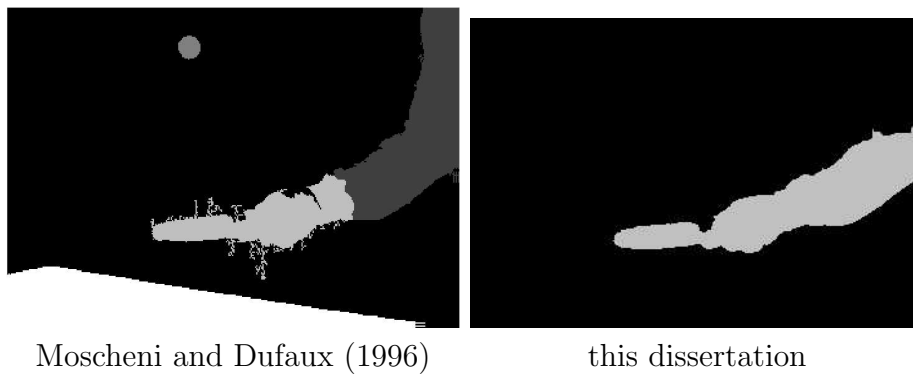


Figure 5.17: *Comparison with Moscheni and Dufaux: Tennis sequence.* A comparison with results presented by Moscheni and Dufaux in [103]. The edge-based approach does not detect all of the arm, but more accurately detects the outline.

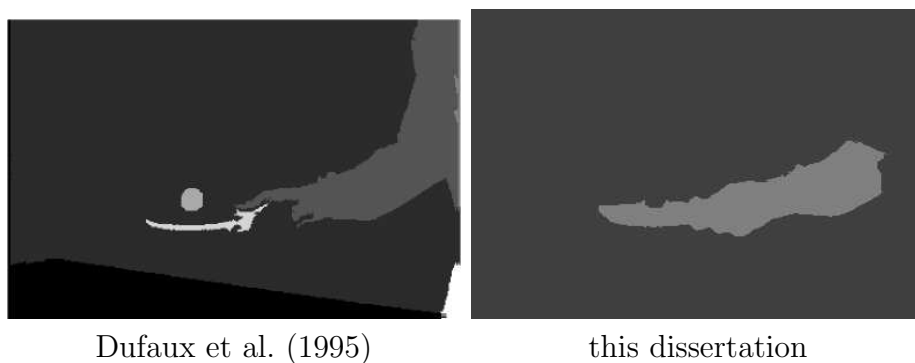


Figure 5.18: *Comparison with Dufaux et al. : Tennis sequence.* A comparison with results presented by Dufaux et al. in [46]. Part of the arm is considered to be stationary by the edge-based approach (which is perhaps reasonable), but all of the hand and bat is extracted as foreground.

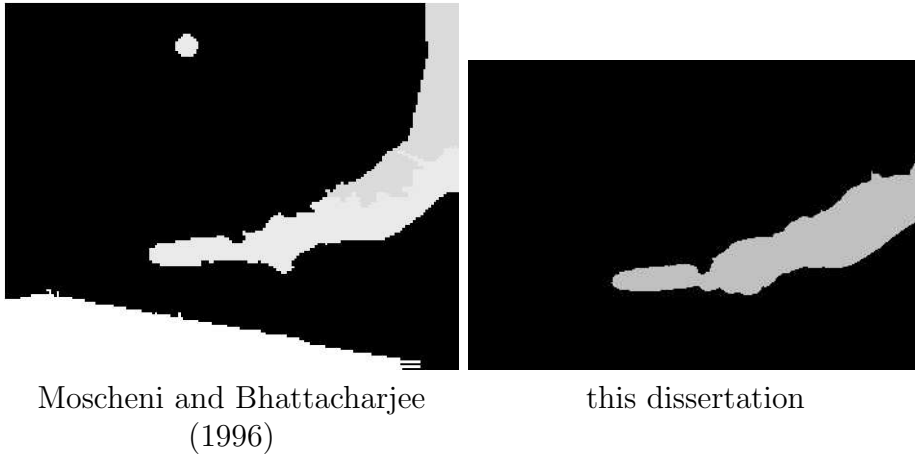


Figure 5.19: *Comparison with Moscheni and Bhattacharjee: Tennis sequence.* A comparison with results presented by Moscheni and Bhattacharjee in [101]. The edge-based approach gives a slightly more accurate object boundary.

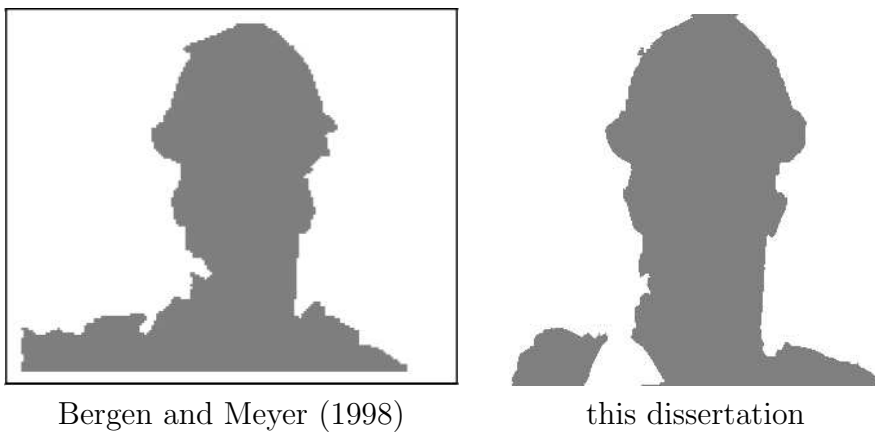


Figure 5.20: *Comparison with Bergen and Meyer: Foreman sequence.* A comparison with results presented by Bergen and Meyer in [12]. The static segmentation used by Bergen and Meyer is inferior to the one used in this dissertation, giving a less accurate boundary.

background. Better results by Moscheni were presented in [101] and, looking at Figure 5.19, are comparable with the edge-based approach.

Bergen and Meyer [12] use a morphological scheme for their static segmentation, but on the evidence of Figure 5.20 (and their static segmentation shown in Figure 2.4) the edge-based approach of Sinclair, adopted in this dissertation, provides considerably more accurate boundaries.

## 5.10 Summary

The edge-based motion segmentation scheme generates fast, accurate segmentations of many of the sequences tested. The results validate the theory of Chapter 3: if the edges in the frame are labelled according to their motion, a complete segmentation can be produced. The main limitation of the implementation presented here is in the edge labelling. When the parametric motion model is inappropriate, or the EM process fails to converge to the global maximum, a poor segmentation is produced. There are also some sequences when the depth ordering of the layers is ambiguous, and some of these are mislabelled. However, in the majority of the sequences tested, a good edge labelling is produced, the correct layer ordering is determined, and an excellent segmentation is the result. The edge-based approach compares very favourably with previous work, both pixel- and region-based, and outperforms a number of existing schemes, particularly on computation speed and the accuracy of the object boundary.

The next two chapters will consider extensions of the implementation presented in Chapter 4. Many of the problems, both the edge labelling and the layer ordering, can be resolved by observing the sequence over more than two frames, and Chapter 6 describes this segmentation of multiple frames. The edge-based framework is also equally applicable to more than two motions, and Chapter 7 considers segmenting sequences containing an arbitrary number of motions.





---

## Extension to multiple frames

---

### 6.1 Introduction

The previous chapters have presented an implementation and evaluation of the edge-based motion segmentation framework, which segmented a frame into two parts using the motion between that frame and the next. While successful in many cases, there can be problems when labelling edges using just two frames, and it is only with a reasonable edge labelling that a complete, accurate segmentation can be produced. Edge labelling errors occur due to noise, or where the motion is ambiguous. This chapter will show that observing the same edges through further frames reduces errors due to noise, and the motion of ambiguous edges can become clearer. As well as clarifying motion information, these additional frames can themselves also be segmented once their motion has been found. The segmentation of multiple consecutive frames from a sequence is essential if an analysis of the motion in a sequence is to be performed, as is required for many applications.

This chapter extends the implementation of Chapter 4 to include information from further frames. The concept of a *cumulative* edge probability is introduced—this is the probability of an edge having the same motion label over several frames. Cumulative edge probabilities provide a more robust edge labelling and lead to an improved segmentation. This chapter also considers some of the problems of tracking extended sequences: edge occlusion and non-parametric deformation, and techniques are presented to cope with these. Finally, results are presented, considering the extended segmentation of the thirty-four sequences already considered in Chapter 5, as well as presenting examples of image mosaicing.

## 6.2 Accumulating evidence: Continued tracking

This section considers using multiple frames to improve the edge labelling, and resolve ambiguities. The starting point for this is the two-frame segmentation of Chapter 4. The task is one improving the labelling of the *frame 1* edges, and thus its segmentation, by tracking these frame 1 edges into frame 3, and further frames. The larger motion between frames 1 and 3, or 1 and 4, and also over the larger time period involved, should allow a more certain edge labelling. In the sequences tested, only a few further frames are typically needed to provide this disambiguation, and greatly improve the edge labelling (see Section 6.6).

The approach followed is the same as for the two-frame algorithm, whereby the EM algorithm is used to estimate the motion and the edge probabilities, only this time between frames 1 and  $K$  (where  $K > 2$ ). It is assumed that the motion between these two separated frames is still approximately described by a projective transformation (or one of its subgroups). It is important to remember that the mapping does not need to be exact—the edge must simply match *better* under one motion than the other.

### 6.2.1 Initialisation

The EM process between frames 1 and 3 can be initialised using the results from the two-frame segmentation. This provides a prior probabilistic labelling for the edges, and an estimate of the motion. In general, the results from frame  $K$  can be used to initialise frame  $K + 1$ , as outlined in Table 6.1. First the edges from frame 1 are transformed into the correct area of the image by extrapolating from each of the previous motions. This is necessary since the search for the edge location in the new frame is only made over a short search track  $\rho$  each side of the edge (usually  $\rho = 20$  pixels), and only normal to the edge, so the search must begin close to the correct location and orientation.

Having transformed the edges to the appropriate region of the frame for each motion, each sample point makes a search to find the most similar pixel in the new image.<sup>1</sup> Given these error distances, a refined initial estimate is made which minimises these errors. Here, the edge probabilities from the previous frame are used to indicate which motion error each edge should be minimising. First each measurement is weighted by the probability that they are motion 1, and these residual errors are summed and minimised, and then the same for motion 2, as described in Section 4.4.5. This gives a good initialisation from which EM may then begin (at

<sup>1</sup>As in Section 4.3.2, the match is based on the squared error in colour image gradients between the pixel on the edge in frame 1, and the candidate pixel in the frame in question.

- Motion initialisation (frame  $K + 1$ )
  - Predict motions by velocity prediction from previous frame
 
$$\Theta_{-1}^{K+1} = \Theta^K + (\Theta^K - \Theta^{K-1}) \quad (6.1)$$
  - Transform tracking nodes under each motion and search for best match
  - Estimate motions  $\Theta_0^{K+1}$  given frame  $K$  edge probabilities
- Repeat (EM Loop)
  - As in Table 4.6

Table 6.1: *EM initialisation for frames after the first two.* To initialise tracking in frame  $K + 1$ , the motion between the previous two frames ( $\Theta^K - \Theta^{K-1}$ ) is used to estimate the new location. The previous edge labels are also used as a bootstrap.

the E-stage), and frequently only a few iterations are required to reach convergence, giving the edge probabilities and motion in this new frame.

### 6.2.2 Occlusion

As the foreground object moves, it occludes edges and sample points on the background layer. Over two frames the problem of occlusion has been ignored as the effects are minimal. However, when tracking over multiple frames, significant numbers of sample points become occluded. These sample points either fail to find a match or, worse, find a spurious match on the foreground layer; this can lead to a poor motion estimation and edge labelling.

The foreground/background labelling for edges and regions from the previous frame's segmentation enable this problem to be overcome, as background edges can be tested to see if they are occluded by any foreground regions in the next frame. The region labelling provides an implied edge labelling, and every edge which is implied to be background has its sample points tested.

Figure 6.1 illustrates the occlusion test. Each sample point to be tested, i.e. those on background edges (red in this case), is transformed under the current estimate of the background motion,  $\theta_B$  to find its new location in the frame in question. This location is then tested to see whether it is occupied by a foreground object in this frame. To do this, the sample point's new location is transformed under the inverse

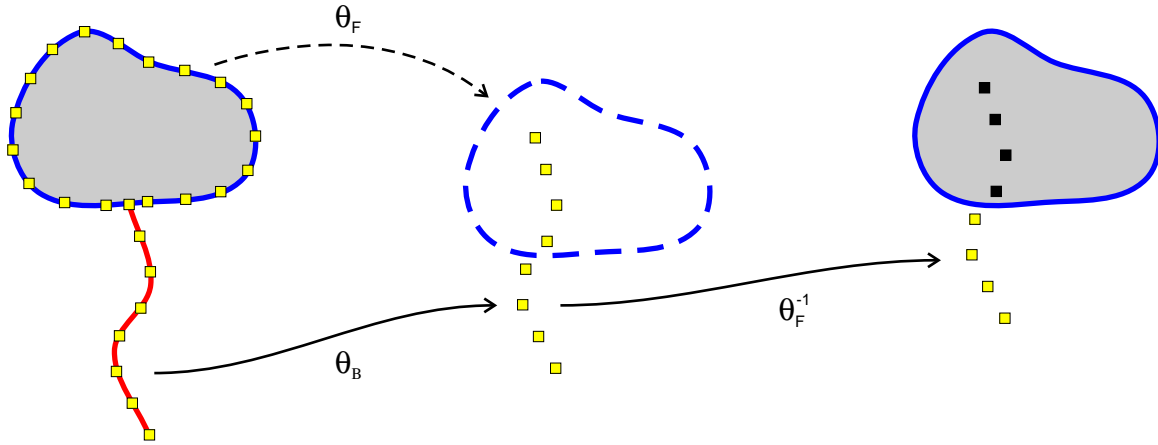


Figure 6.1: *Detection of sample point occlusion.* Sample points on edges previously labelled as background (red) are transformed according to the current background motion,  $\theta_B$ , and then under the inverse of the foreground motion,  $\theta_F^{-1}$ . If they fall within regions previously labelled as foreground (blue), they must have been occluded.

of the foreground motion,  $\theta_F^{-1}$ . If the point falls into a region labelled as foreground in the previous frame then this point is now occupied by a foreground region and it must be occluded. Each occluded sample point is marked as such, and does not contribute to the tracking for that edge. All sample points are also tested to see if they project outside the frame under the current motion and, if so, they are also ignored.

This occlusion test should be performed before the EM loop commences, to prevent these points from affecting the solution, but it requires estimates of the current motions  $\theta_F$  and  $\theta_B$ . The occlusion test therefore uses the approximate motions from the initialisation stage (outlined in Section 6.1), and is performed just before the EM loop.

### 6.2.3 Combining statistics

Tracking the edges between frame 1 and 3 provides an estimate of the total motion between those pair of frames and also an estimate for each edge of the probability that it obeys each of those motions. The two-frame algorithm also provides edge motion probabilities, this time for the motion between frames 1 and 2. Each edge must obey the same motion across all frames—an edge either remains foreground or remains background, it cannot change label part-way through a sequence.<sup>2</sup>

The probability that an edge obeys a particular motion over a sequence is the probability that it obeyed that motion between *each* of the frames in the sequence.

<sup>2</sup>The case of a foreground object moving and then stopping is ignored here, as it is unlikely to happen in the space of the few frames considered to refine the statistics. However, Section 6.6 shows some cases where this occurs as part of a longer sequence.

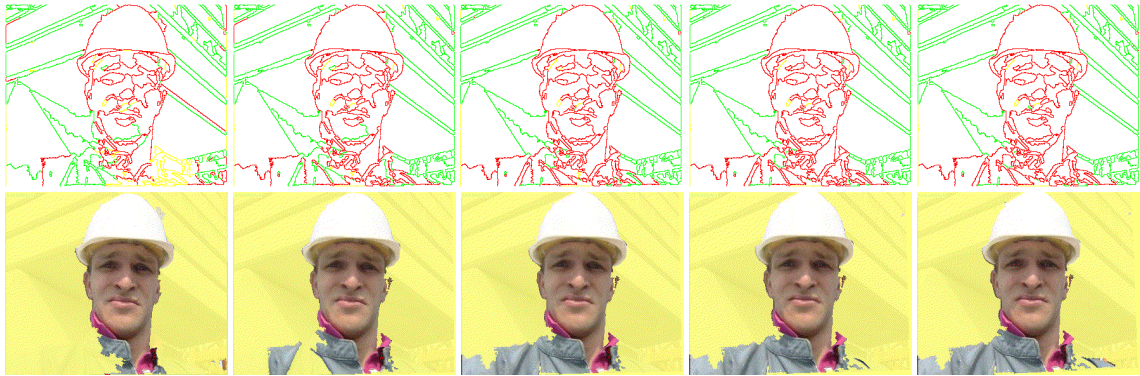


Figure 6.2: *Foreman sequence: Cumulative statistics.* The edge probabilities and region labels as evidence is accumulated over 5 consecutive frames. Beginning with a two-frame segmentation, the edge probabilities become more certain as more frames are used, and the region segmentation improves.

Since the sample point matches found in each frame are independent, the edge probabilities are also independent. As a result, the probability that an edge obeys motion 1 is the product of the probabilities that the edge obeyed motion 1 for each of the frames considered. The probability for motion 2 may be calculated similarly. Each edge must obey one of these hypotheses, so they may be normalised to give a *cumulative edge probability* of each motion.

### 6.3 Using cumulative statistics to segment a frame

The segmentation of a frame begins with a two-frame segmentation, as in Chapter 4. After this is completed, further frames are considered. For each frame an independent EM maximisation is performed, using only the edge probabilities and the motion between the frame in question and frame 1. The initialisation of EM is bootstrapped by the results of the previous frame. After convergence the final probabilities are multiplied together with the probabilities from the previous frames to give the cumulative edge statistics. The region and foreground labelling can then be performed in the same way as earlier (Section 4.6), but using the cumulative edge statistics instead of the edge responsibilities from the EM algorithm. Since this is still the segmentation of frame 1, the static segmentation of the frame need only be performed once.

Figure 6.2 presents an example of the accumulation of statistics. It clearly indicates the benefit of considering more frames, which improves the edge probabilities and motion segmentation of the original frame. Cumulative statistics are considered further later in this chapter, as part of a deformable multi-frame segmentation. First, however, a simpler form of multi-frame segmentation is considered.

## 6.4 Templated segmentation of a sequence

Once the (parametric) motion between each frame of a sequence is known, and the segmentation of frame 1 has been performed, this may be used to provide a rudimentary segmentation of the sequence.

The estimate of the foreground motion between each frame describes how the pixels belonging to the foreground object transform between frames. The segmentation of frame 1 may be used as a *template*, or a mask, which specifies the foreground pixels. Transforming this template according to the foreground motion provides the location of the foreground objects in the new frame, if it is assumed that the image motion of the object agrees with the parametric motion model (i.e. there are no other deformations). Over a short sequence this is commonly an adequate approximation. A short sequence can thus be segmented by using this foreground template, transformed by the motion, to cut out the foreground object in each frame. This form of segmentation also provides a useful test of the accuracy of the foreground motion estimation.

Figures 6.3–6.5 show examples of sequence segmentations using this approach. Each figure shows the segmentation of the original frame, and then the templated segmentation of a number of subsequent frames. In each case the quality of the segmentation of the additional frames is almost indistinguishable from that of the original frame. This shows, firstly, that the 2D affine motion model (used here) is capable of suitably modelling the inter-frame motions seen here, and even the motion between more widely spaced frames. Secondly, it shows that this parameterised motion is well estimated by the EM process.

The segmentation after the first few frames shows no obvious alignment errors, and the segmentations are good. It is only in the later frames of **Foreman** or **Tennis** sequences (Figures 6.3 and 6.4 respectively) that small errors can be observed. The modelling of the Foreman's head motion shows an error of 1–2 pixels in some areas by frame 7, with some of the background visible to the left of his hat brim. The Tennis sequence similarly has the occasional error by frame 7, with a little of the background visible under the player's shirt cuff. In both cases, however, the error is not due to any particular misestimation of the motion, but because over this larger period the object's image motion cannot be accurately described by the affine parameterisation. The **Car** sequence continues to be accurate for longer—in this case the moving object is (obviously) substantially more rigid and so exhibits less image deformation. Over short sequences of rigid objects, this is an easy and appropriate method of segmentation.

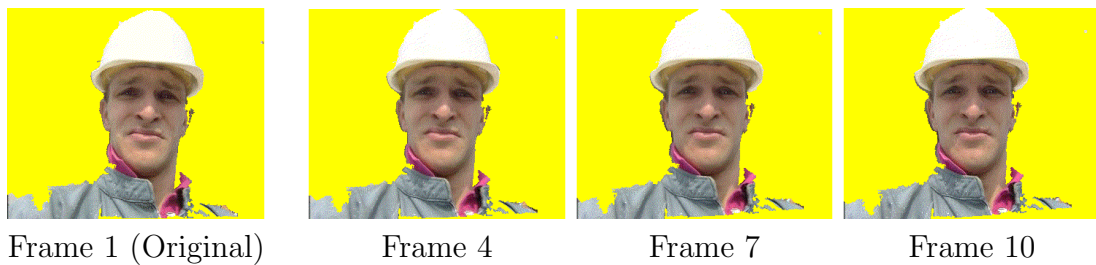


Figure 6.3: *Templated segmentation of the Foreman sequence.* The foreground segmentation for the original frame is transformed under the foreground motion model and used as a template to segment subsequent frames. As the frames progress, a small amount of the background can be seen to the left of the hat-brim (the dark pixels are not present in frame 1).

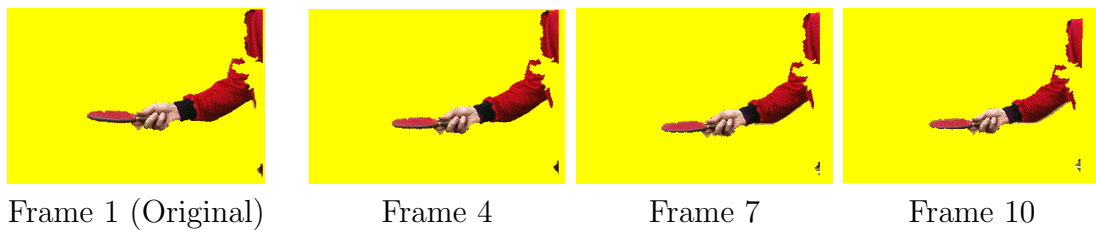


Figure 6.4: *Templated segmentation of the Tennis sequence.* The foreground segmentation for the original frame is transformed under the foreground motion model and used as a template to segment subsequent frames. In frame 10 a strip of background pixels can be seen below the player's lower arm, which are not visible in frame 1.

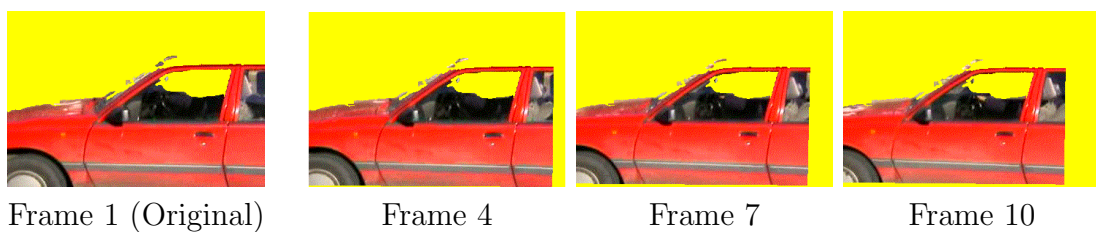


Figure 6.5: *Templated segmentation of the Car sequence.* The foreground segmentation for the original frame is transformed under the foreground motion model and used as a template to segment subsequent frames.

## 6.5 Deformable segmentation

To segment a general sequence, a more sophisticated multi-frame segmentation technique is needed—over a longer sequence, or with non-rigid objects, the image motion will not obey a simple 2D parametric model. In this case it is not sufficient to rely on one static segmentation, and instead it is necessary to segment each frame anew each time, using the local image edges. What is then needed is some means to continue to accumulate statistics across these new edges and segmentations. This process will enable a robust segmentation which adjusts to the changing object—a *deformable* segmentation.

### 6.5.1 Segmenting a new frame: Propagating edges

As introduced in Section 4.2, the image edges are found using the Canny edge detector, and conservative thresholds are set to avoid edges due to shadows or texture. This means that valuable edges may sometimes be missed, and this can be particularly problematic if these missed edges belong to occluding boundary. These boundary edges help the region segmentation produce an accurate representation, and increase the chances of finding T-junctions which determine the layer ordering. The Canny edges are also used to guide and constrain the static segmentation and an accurate segmentation of the objects is best achieved when the occluding boundary exists in the edge map and can act as a hard constraint to the region growing. Figure 6.6(a) shows the edges detected in frame 1 of the **Foreman** sequence, and in Figure 6.6(c) the edges detected in frame 2. In this case it can be seen that part of the boundary on the hat is missing in the second frame. These edge detection errors can be corrected by propagating edges from the previous frame.

To determine candidates for propagating, each edge from the previous frame is transformed according to each of the motions between the frames, and tested to see whether it finds a match. Testing for a match again uses the sample points, so in each new location the edge's sample points find their best matches. If the product of an edge's sample point likelihoods under the 'correct motion' distribution is greater than that under the 'incorrect motion' (Section 4.4.4, particularly (4.36)) then the edge is deemed to have found a match, and is 'trackable'.

Figure 6.6 shows the process. The edges from frame 1 are tested under each motion and the trackable edges are marked in red in Figure 6.6(b). It can be seen that (as should be expected), almost all edges are only trackable under one of the two candidate motions. Any sections of trackable edges which are not already present



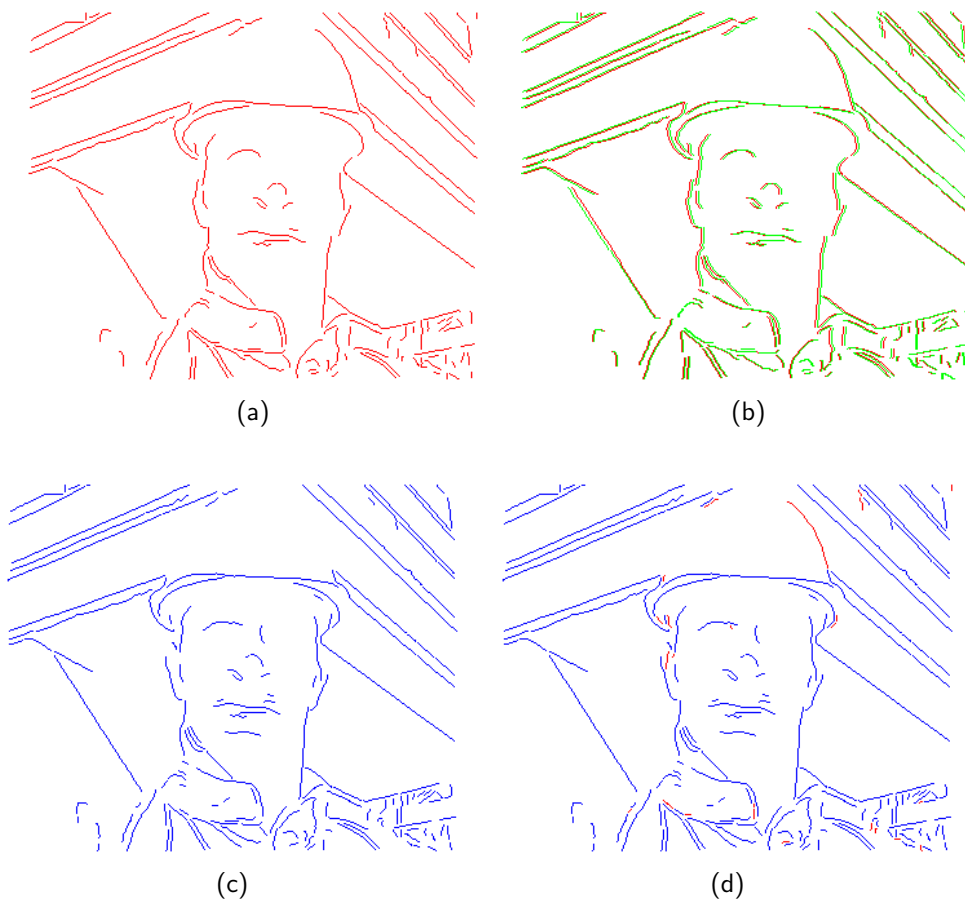


Figure 6.6: *Propagation of edges to the next frame.* (a) Edges detected in frame 1; (b) Frame 1 edges transformed under each motion. These are marked as red if they find a match in the next frame (i.e. they are 'trackable'); (c) Edges detected in frame 2; (d) Missing edges in frame 2 (such as parts of the hat) are filled in with trackable edges from frame 1.

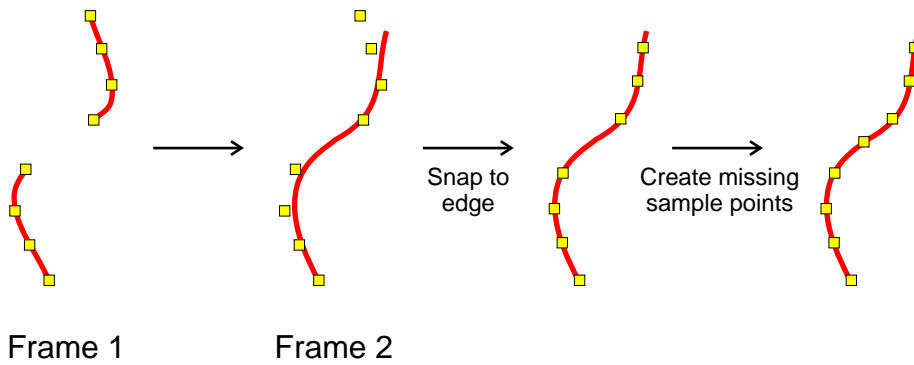


Figure 6.7: *Propagation of sample points between frames.* Sample points on an edge in frame 1 are mapped into frame 2 according to the motion parameters if that edge finds a match. They are then allowed to move up to 2 pixels to lock onto any new edge detected in the frame. Sections of new edges still without sample points then have new sample points created.

in frame 2 are added into the edge map. Figure 6.6(d) shows the augmented edge map.

This propagation of edges ensures that useful edges which are less clear in the new frame continue to be included and tracked, assisting both the motion estimation and the region segmentation. However, by relying as much as possible on edges detected in this new frame this new edge map also represents any non-parametric deformations of the objects that may have occurred between the frames.

### 6.5.2 Accumulating evidence: Propagating sample points

Although the static segmentation of the new frame starts afresh, with newly-detected edges (albeit augmented with previous edges), the previous edge probabilities should be exploited to make use of the cumulative edge statistics. Since edges are detected anew in each frame, persistence is difficult to ensure on a per-edge basis. An image contour detected as three separate edge sections in one frame may be detected as two different sections in the next frame, and it is unclear how the probabilities for one can be satisfactorily combined with the other. However, along an edge it is the sample points which are used to estimate the motions and edge probabilities, and these can be easily propagated from frame to frame. A new edge's probability can be estimated from whatever sample points are assigned to it.

Figure 6.7 demonstrates the sample point propagation scheme. For each edge which found a match in the next frame (again, based on whether the sample points had a higher probability under the 'correct' or the 'incorrect' motion hypothesis), the sample points are mapped into the new frame according to the edge's motion. It is known that the parametric model provides a reasonable match, but that the real

image edge may be one or two pixels away (in Section 4.4.4 it was seen that 96% of all good matches are at most at a distance of two pixels). Therefore, if the sample point does not map exactly onto an edge in the new frame, it is allowed to move a distance of up to two pixels to move onto the nearest real edge location in the new frame. This sample point propagation allows non-rigid objects to be tracked by this system while still retaining the simplicity of a 2D parametric motion model. Edges still lacking sample points have new sample points created.

### 6.5.3 Accumulating edge probabilities

Having found the new set of edges and assigned sample points, the edges may be tracked into the next frame (i.e. one further on from the ‘new’ frame) and labelled by EM as before. Initialisation for EM can proceed as described in Section 6.2.1: an initial set of motions is given by a velocity estimate from the previous frame, while the prior probability for each edge can be determined from the sample points. Each sample point maintains a record of the match probability *for that sample point* across all previous frames. As with edges, each sample point (which is simply part of an edge) must obey a single motion across all frames. Thus, for this new frame, the prior probability that an edge obeyed motion 1 is the probability that each sample point assigned to that edge obeyed motion 1 across all previous frames. From the previous independence assumptions, this is simply the product over all that edge’s sample points, over all the frames for which they have existed. Having determined the initial edge probabilities, EM can be initialised (as outlined earlier, in Table 6.1) and run to convergence.

This EM process is independent of that from the previous frame and the edge probabilities after convergence merely represent those of the current edges under the motion from the current frame to the next. However, those sample points which were propagated from the previous frame also store a record of their probabilities from the previous frame. As a result a cumulative probability may be calculated for each sample point (and, from this, also for an edge)—it is the probability that the sample point obeyed the same motion in all its frames. Since the matches found in each frame are independent, this is given by the product of all the inter-frame motion probabilities under that motion.

As an example, Figure 6.8(a) shows the edge labels in the **Foreman** sequence after the first frame, and Figure 6.8(b) the results of EM between frames 2 and 3. Both contain a few incorrectly labelled edges. After accumulating the sample point statistics, the cumulative edge probabilities are calculated, and these give more

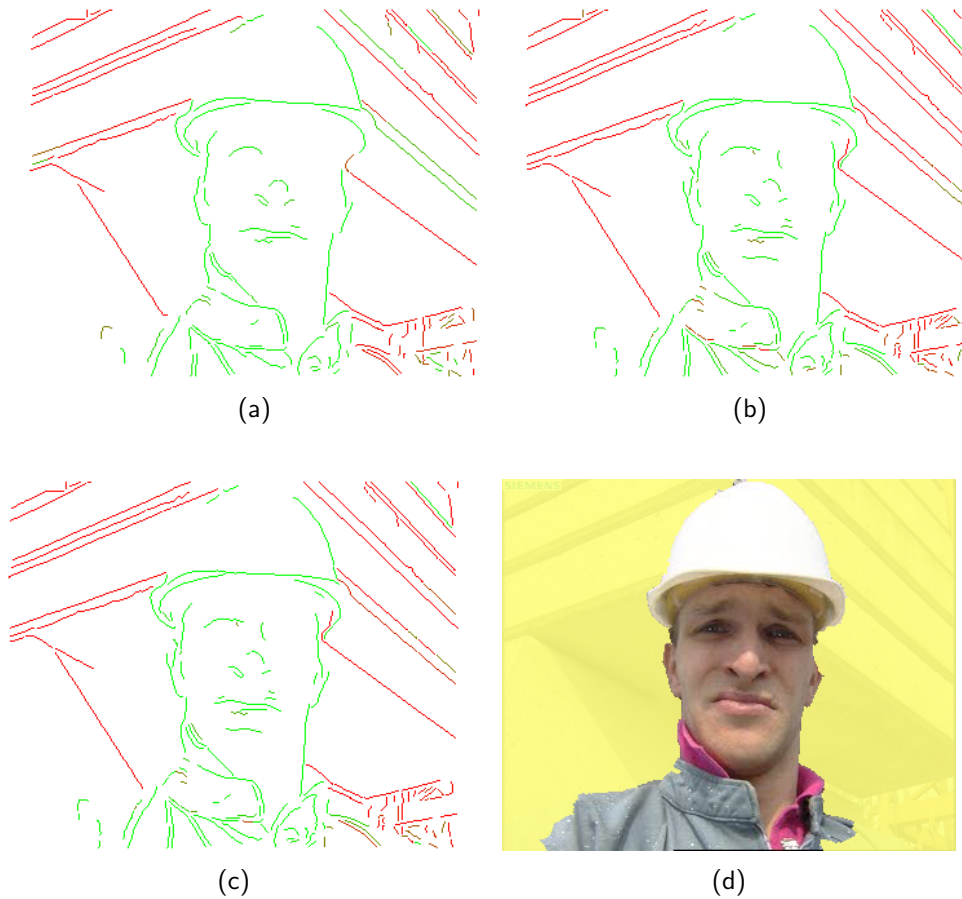


Figure 6.8: *Cumulative statistics for propagated edges.* (a) Edges in frame 1, labelled according to their inter-frame motion probabilities. Note that some of the diagonal edges to the right of the head are incorrectly labelled; (b) Edges in frame 2, labelled according to their inter-frame motion probabilities. Here, note the errors around the collar; (c) Edges in frame 2, labelled according to the cumulative probabilities over the 3 frames. Most of the errors present in one of the previous frames are corrected by this accumulation; (d) Labelled region segmentation of frame 2.

- Find edges
- Transfer trackable edges from previous frame as required
- Transfer sample points from previous frame
- Allow sample points to snap to nearest edge
- Track and label edges between this frame and the next
- Accumulate probabilities for edges and sample points
- Segment frame into regions
- Label regions using cumulative edge probabilities

Table 6.2: *Deformable segmentation of a new frame.* Overview of the propagation stages and segmentation.

robust edge labels. Figure 6.8(c) shows the improved edge labelling given by the cumulative statistics, and Figure 6.8(d) the improved segmentation.

#### 6.5.4 Continued deformable segmentation of a sequence

The techniques developed in this section enable an accurate segmentation of further frames to be performed even when the objects undergo additional deformation which is not described by the parametric model. The standard two-frame segmentation is performed between the first two frames and this is used to initialise further frames. Each new frame then uses newly-detected image edges, but propagates additional edges as required and, most importantly, also propagates the sample point statistics. This allows cumulative edge probabilities to be used with even the new edges. Table 6.2 gives an overview of the segmentation of these further frames, and this process can be repeated for each frame to enable a complete sequence to be segmented.

## 6.6 Evaluation

The ‘deformable segmentation’ approach has been tested on the corpus of sequences highlighted in Appendix D. As in Chapter 5, four sequences are considered in detail (Foreman, Tennis, Coastguard and Car), and then the performance over the complete set of thirty-four sequences is discussed.

### 6.6.1 Foreman sequence

#### Edge propagation

The first stage in the segmentation of the second frame in a sequence is to find the new edges, and fill any gaps in the edge map with edges from the previous frame. The complete edge map is shown in Figure 6.9(a), with the propagated edge sections shown in red. In this case most of the edges are found by the edge detector in the new frame and so do not need to be propagated. However, a few, such as the edge of the hat, are usefully added by this process.

#### Sample point propagation

Most of the sample points for the edges in the second frame are provided by propagating those from the previous frame, moving them a short distance onto the new edge if necessary. Of the 804 sample points in the first frame, 721 are successfully propagated, and Figure 6.9(b) shows the new configuration of sample points. Any gaps, such as on the edges on his left shoulder, are filled in with newly created sample points. The colour of the sample points in Figure 6.9(b) indicates their motion probability in the previous frame. This will be combined with the probability in this new frame to give the cumulative edge probability.

#### Cumulative edge probabilities

The motion between the second and third frames is estimated by EM, as before, and the resulting edge probabilities are shown in Figure 6.11(c). In this case it takes twenty iterations to converge. There are a number of errors in some of the small edges on his collar, but, with these exceptions, the probabilities appear to be, as for the previous pair of frames, very plausible (c.f. Figure 5.2(a)). The background edges on the right of his head, which were mislabelled in the previous frame, are more certain on this occasion.

Taking the product of the two edge labellings (and re-normalising) gives the cumulative labelling, encompassing the evidence from both frames. Figure 6.9(d) shows this cumulative labelling and it can be seen that, now, almost all of the edges are labelled correctly. The edges on his collar are closer to the correct labelling and the edges to the right of his head are, while still not perfect, better than in the previous frame. In this cumulative labelling, 89% of the edges are correctly labelled, compared with a hand-labelling. In the first frame only 78% of edges were correct, so this is a significant improvement.

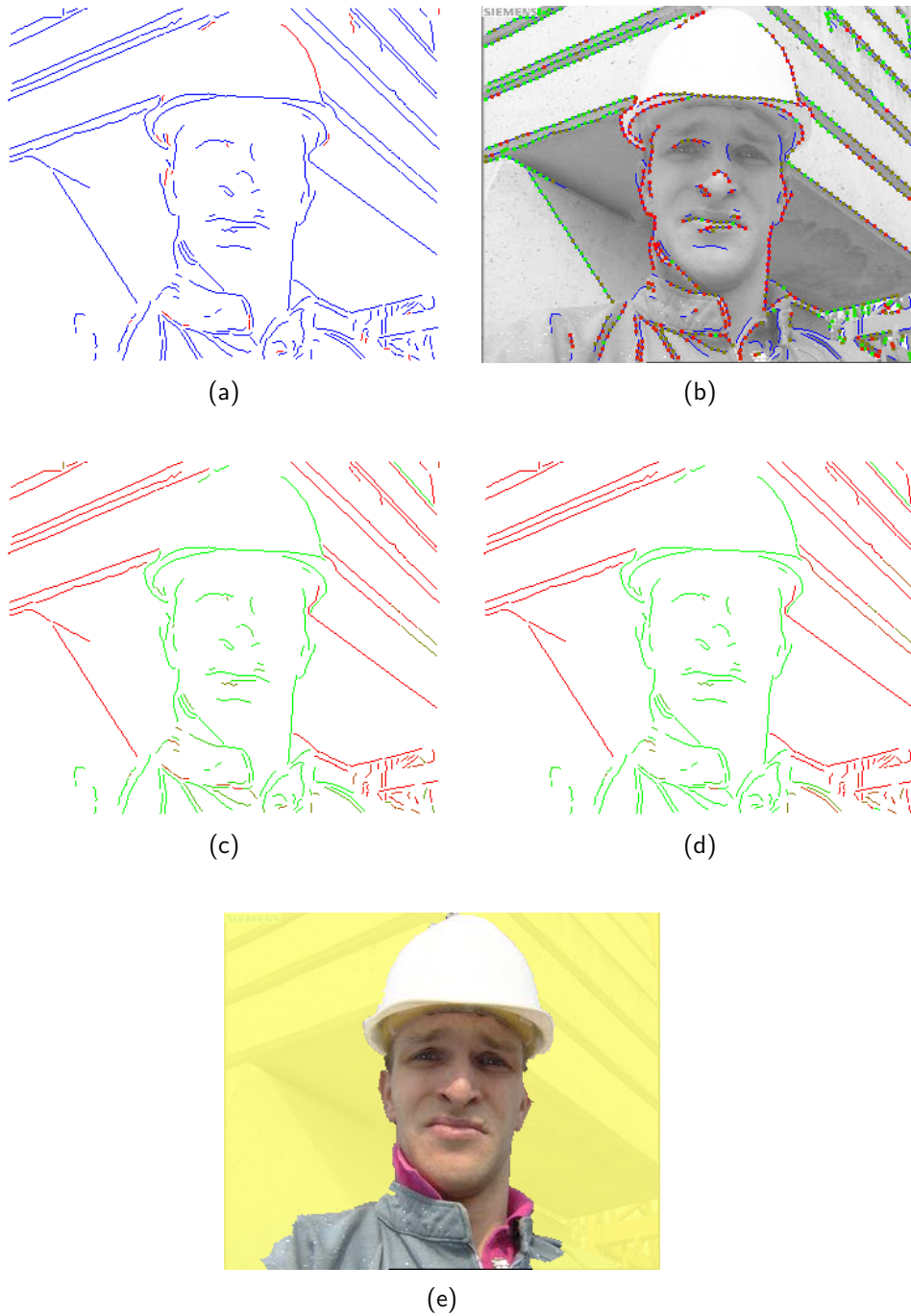


Figure 6.9: *Foreman segmentation of the next frame.* (a) Detected edges (blue) augmented by propagated edges (red); (b) Sample points propagated from previous frame. New sample points are created to fill any gaps; (c) Edge motion probabilities between the second and third frames; (d) Cumulative edge probabilities over both frames; (e) Segmentation of second frame.

### Final segmentation

Figure 6.9(e) shows the final region labelling, following a static segmentation of this second frame and the simulated annealing process (using the cumulative edge probabilities). This yields an even better solution than using only two frames, with 99% of pixels labelled correctly, compared with a hand-labelling.

### A longer sequence

Figure 6.10 shows the results of continuing this process over a larger number of frames and it can be seen that over later frames the segmentation is problematic. The diagonal edges still cause occasional problems, but major errors only begin to occur about nine frames in. The first problems occur when the man stops moving his head for a few frames. The EM process still attempts to fit two motions during these frames (the third row of Figure 6.10), and the motions end up converging on two solutions which only differ significantly due to noise. Unfortunately, even with these similar motions, the edge probabilities tend to saturate and a near-random edge labelling results. When these are accumulated with the earlier probabilities over a number of frames, the cumulative edge probabilities are diluted to the extent that a poor segmentation results. This problem should be resolved by selecting the best number of motions on a per-frame basis—where there is no foreground motion, only one motion will be fitted and no dilution of the edge probabilities will occur.

The second problem is that, between frames 15–28, the foreman throws his head back and opens his mouth. This rapid motion cannot be parameterised by the affine motion model, and the sample point errors are much larger than the mean distance (for a correct match) of 1.3 pixels. With these large motions, the sample points are also not propagated. As a result, edges such as the top of his hat cannot be fitted and the edge probabilities on these edges are again governed by noise and saturation. This problem cannot be resolved with the current system, but fortunately motions such as this are rare in the sequences considered. After this violent motion, the edge labels settle down and are again reasonable. Unfortunately, after passing through an almost random labelling during the previous few frames, the motion which converges on the head during the last frames is that which in earlier frames modelled the background. This is highlighted in this example because the layer ordering is assumed to be constant across frames. If the layer ordering were allowed to swap mid-way through a sequence, a correct segmentation would also be obtained for the last few frames.



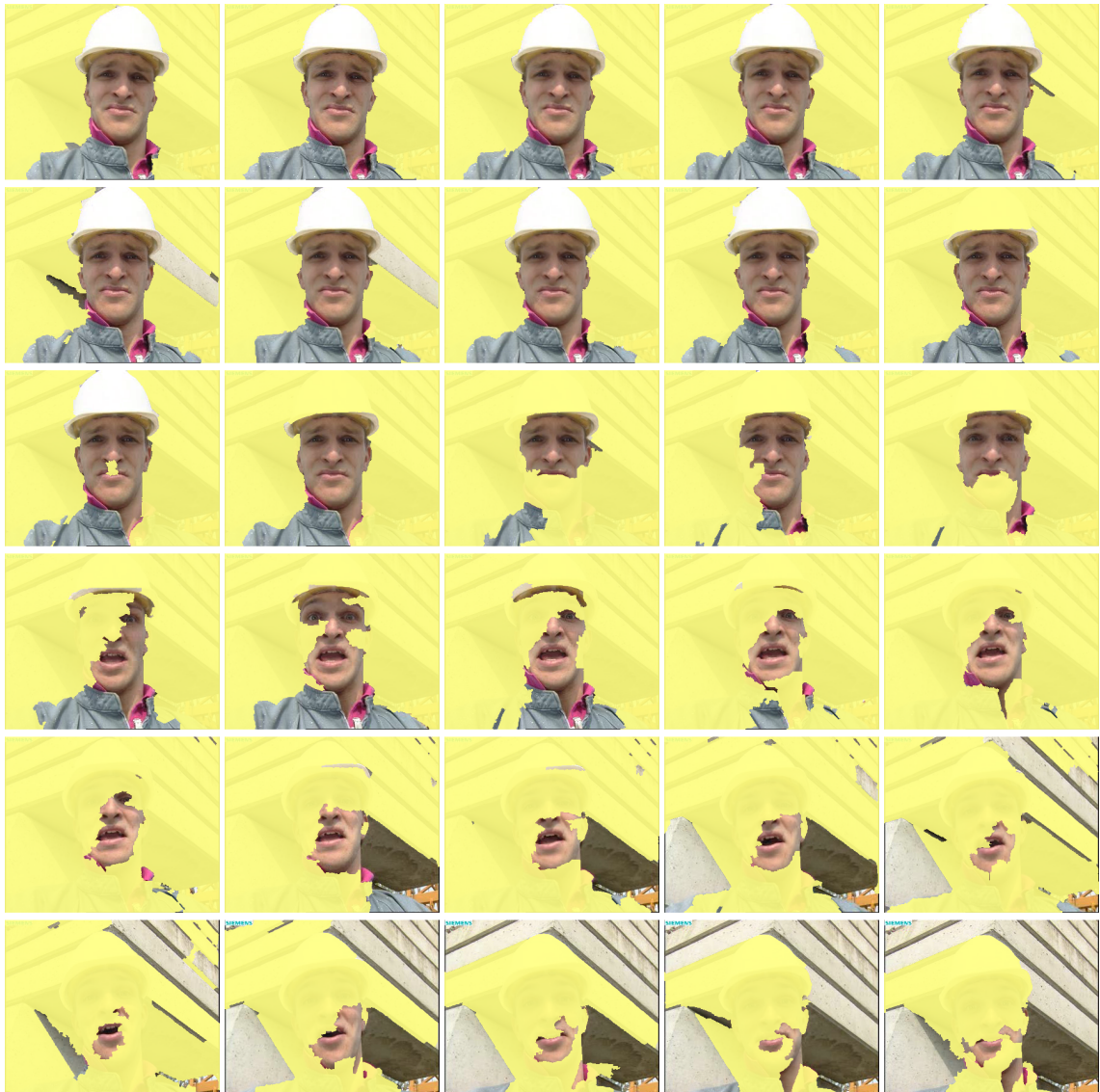


Figure 6.10: *Segmentation of the Foreman sequence.* Segmentation of thirty consecutive frames.

### 6.6.2 Tennis sequence

Figure 6.11(a) shows (in blue) the newly-detected edges in the second frame of the Tennis test sequence. As with the previous sequence, a valuable edge is added from the previous frame—in this case part of the top of the bat was not detected using the conservative edge detection thresholds. Figure 6.11(b) shows the propagated sample points, and it is apparent that, even without any motion analysis, the prior labelling will be very good. The player's upper arm was not well tracked between the first two frames (it was ambiguous). As a result its sample points were not propagated and new ones must be created along those edges. Between frames two and three, the upper arm more obviously obeys the green motion, as can be seen in Figure 6.9(c). When the results are combined with those from the previous frame, the edge labelling is still very good, with 96% of edges labelled correctly (Figure 6.9(d)). With this excellent edge labelling, the final region labelling (Figure 6.9(e)) cannot be faulted, segmenting the frame as accurately as could be done by hand.

The segmentation of the first thirty frames of this sequence is shown in Figure 6.12. The segmentations continue to be excellent, with only a few frames exhibiting unwanted behaviour, all of which may be described as differences in interpretation rather than errors:

**Ball segmented with foreground** In frames 4–6, the ball has reached the top of its flight and begins to fall. For this short period it has a very similar motion to that of the arm, and is segmented as foreground. This may, of course, be the desired segmentation in some cases. It is certainly the correct ‘motion’ segmentation.

**Missing upper arm** Between some frames the player's lower arm, hand and bat move much more than the upper arm, which at times is almost stationary. As a result the upper arm sometimes appears to have a motion more similar to that of the background motion, and is labelled as such. This again is the correct ‘motion’ segmentation, even if it is not the desired solution.

**Background regions glued to the arm** The background in each frame is segmented as several different regions. These are grown from different seed points and, unless the mean colour is very similar, they will continue to be distinct regions (it is not wise to merge regions before the motion segmentation stage unless absolutely certain that they are part of the same object). The boundaries of these regions do not necessarily have to include one of the detected Canny edges—they can simply be the points where regions have met during

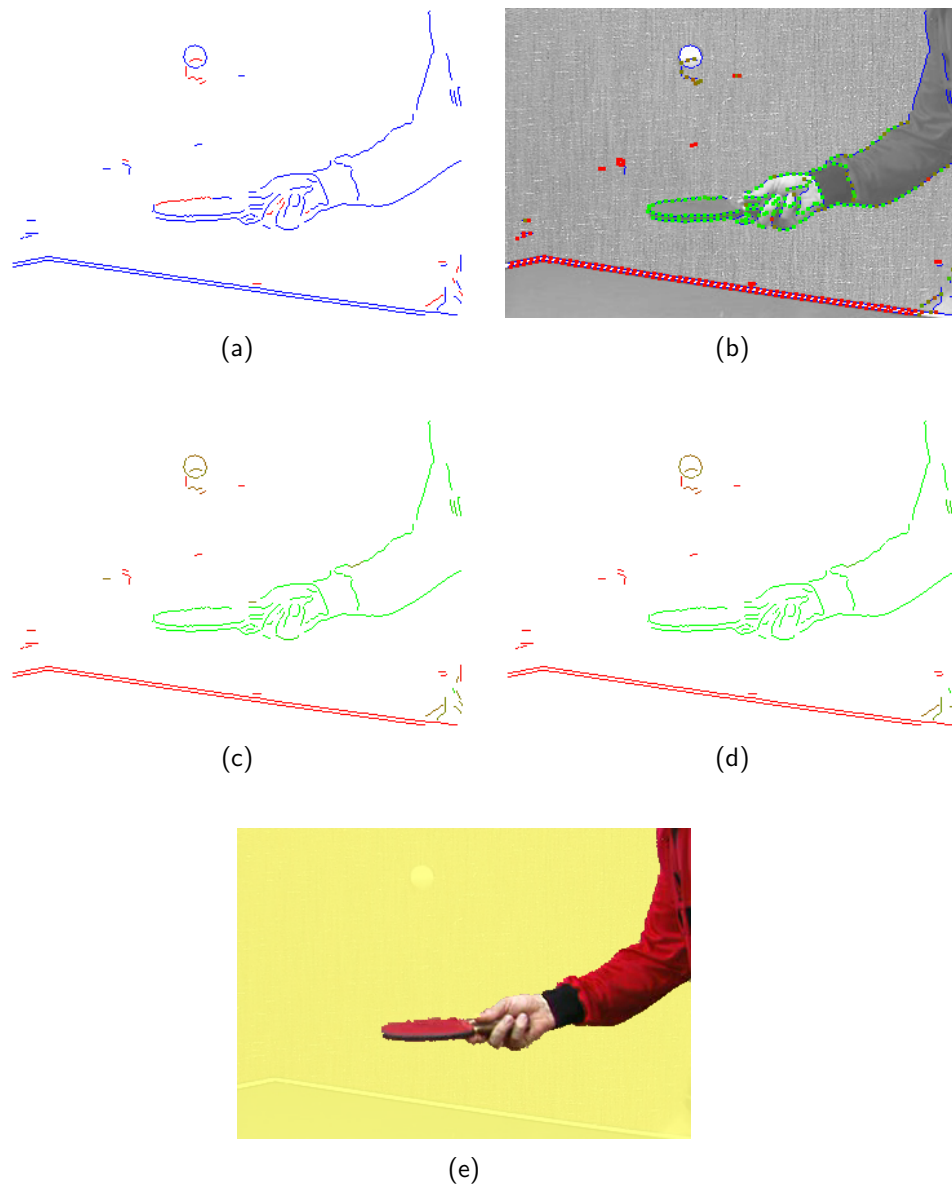


Figure 6.11: *Tennis segmentation of the next frame.* (a) Detected edges (blue) augmented by propagated edges (red); (b) Sample points propagated from previous frame. New sample points are created to fill any gaps; (c) Edge motion probabilities between the second and third frames; (d) Cumulative edge probabilities over both frames; (e) Segmentation of second frame.



Figure 6.12: *Segmentation of the Tennis sequence.* Segmentation of thirty consecutive frames

the region growing process. In the frames which exhibit erroneous background regions, the regions in question do not have, as part of their boundary, any detected edge labelled with the background motion, and their only labelled edge is foreground. This means that the labelling of this region is ambiguous—it would agree with the labelled edges if it were background *or* if it were foreground. However, since the region labelling stage combines both the edge evidence and a MRF-style prior, a decision can be made. In these cases this prior has forced the most contiguous (but incorrect) solution. Given the current edges and the region segmentation, the motion of these regions truly is ambiguous and a correct labelling of these regions cannot be guaranteed without more labelled edges.

### 6.6.3 Coastguard sequence

The third of the standard test sequences considered, the **Coastguard**, performed less well using only two frames (see Figure 5.5), and it is interesting to see whether using any more frames can improve the edge labelling, or resolve the errors in the static segmentation. Figure 6.13(a) shows the augmented edges in the second frame, but it can be seen that the stern and prow of the boat are still missing (there is still no obvious intensity difference between the colour of the hull and the water). Only a few inconsequential edges are propagated. Almost all of the sample points are propagated, but it can be seen from Figure 6.13(b) that many of them are incorrect or ambiguous, highlighting the difficult nature of this sequence, with its highly textured areas.

The edge labelling for the inter-frame motion from frames two to three (Figure 6.13(c)) is, as before, rather noisy, with many of the small background edges uncertain of their labelling. Comparing this with the labelling in the previous frame, Figure 5.6(a), it can be seen that, as expected for errors due to noise, there is no consistency in the mislabelled edges. As a result, when the edge probabilities are combined over the two frames, as shown in Figure 6.13(d), the edge labels are improved—most edges do have at least one good labelling, which dominates. This process continues to improve the edge labelling over the whole sequence (the edge labelling for Frame 10 may be seen in Appendix D). The only edges which continue to be occasionally labelled incorrectly are the horizontal edges, since both motions are horizontal.

The resulting foreground segmentation, Figure 6.13(e), is an improvement over the previous frame, this time only missing a small amount of the bow and stern.

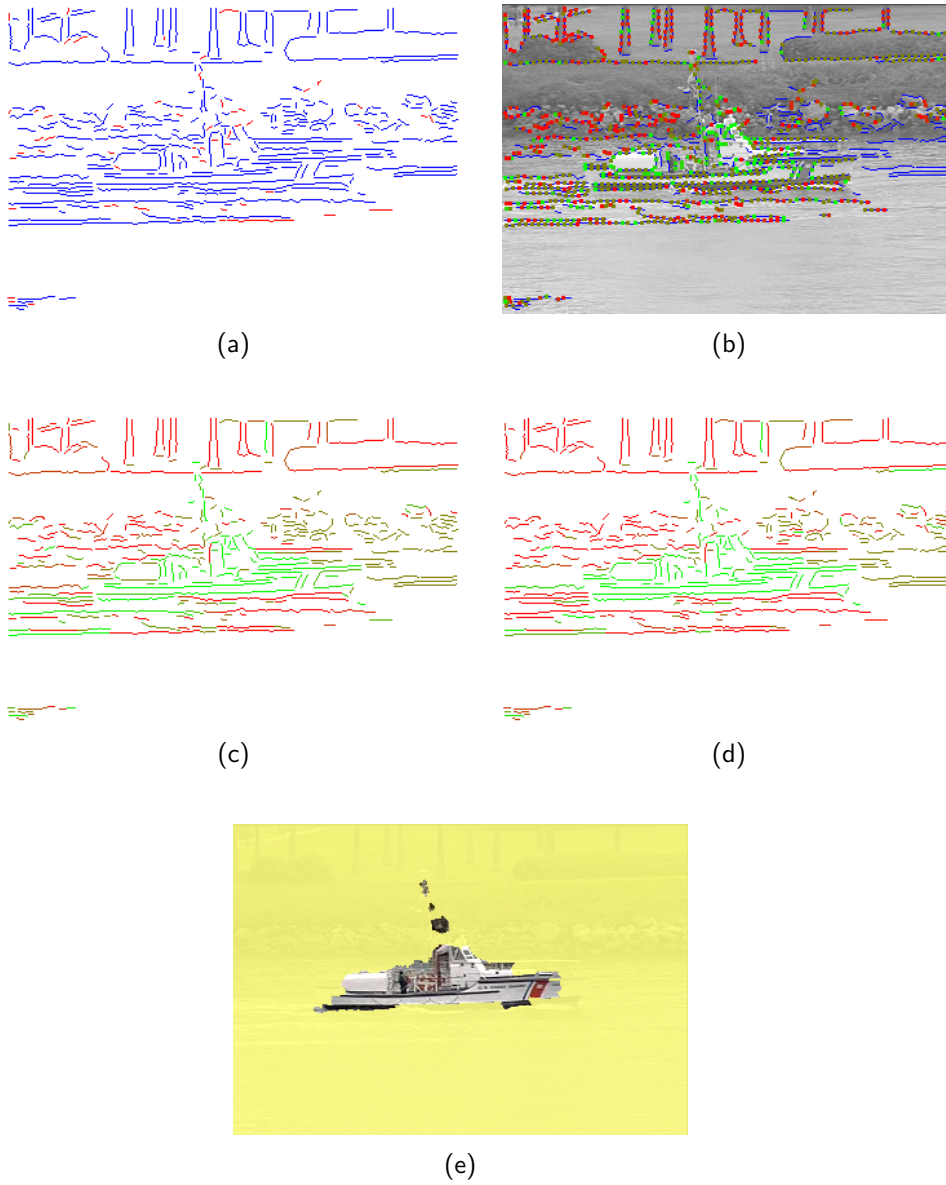


Figure 6.13: *Coastguard segmentation of the next frame.* (a) Detected edges (blue) augmented by propagated edges (red); (b) Sample points propagated from previous frame. New sample points are created to fill any gaps; (c) Edge motion probabilities between the second and third frames; (d) Cumulative edge probabilities over both frames; (e) Segmentation of second frame.



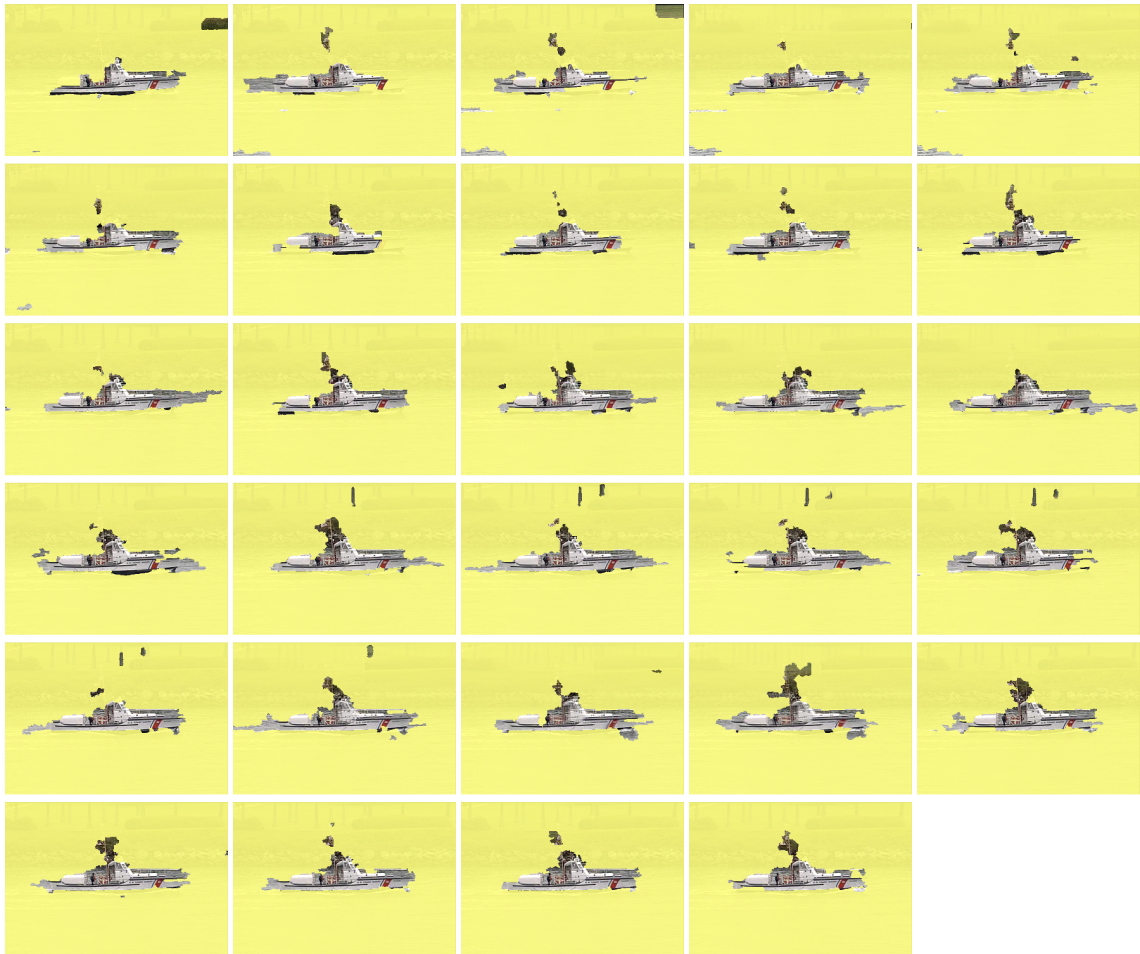


Figure 6.14: *Segmentation of the Coastguard sequence.* Segmentation of twenty-nine consecutive frames. (A software bug prevented further frames being processed in this case.)

As before, this occurs because the static segmentation merges these parts with the water. In this frame some of the edges belonging to the boat's mast have also been extracted and correctly labelled. However, most objects as small as the mast are not included in the static segmentation, so only a coarse attempt can be made to represent these in the final motion segmentation.

Figure 6.14 shows that the segmentation is stable over a longer sequence. The occasional background region is included in the segmentation, due to a noisy labelling of some edges, but the boat is well segmented throughout. The prow of the boat is finally represented in the region segmentation by the twenty-first frame, and better segmentations of the front of the boat are then performed, but the stern always looks too similar to the wake.

These results, however, show that the framework and implementation still work reasonably when occluding edges are missing. Most of the segmentation is correct, and if occluding edges are missing then the segmentation will be a *subset* of the desired segmentation. Resolving this problem of a missing occlusion boundary is discussed as part of the further work in Chapter 8.

#### 6.6.4 Car sequence

The final test sequence considered here in detail is the **Car** sequence. Figure 6.15(a) shows the detected and propagated edges, and again some useful edges are added from the first frame. Most of the sample points are propagated and these provide a good prior edge labelling. In this sequence, with the large foreground motion, occlusion is a major concern, particularly over an extended sequence, but as explained in Section 6.2.2, the sample points on background edges are tested for occlusion. Figure 6.16 shows those sample points identified as occluded in the next frame (marked in red). These points are well identified, and it is particularly pleasing to note that the background sample points which are visible through the window are correctly treated. Also marked (in blue) in this figure are those sample points which project off the image in the next frame, due to either motion. As with points occluded by the foreground motion, these are not used for motion estimation or calculating edge probabilities.

The edge probabilities after EM (Figure 6.15(c)) are as good as before, with the only major ambiguities being in the horizontal edges, and the only major errors in the region of the reflections. The cumulative edge probabilities of Figure 6.15(d) reinforce these, and the final segmentation, Figure 6.15(e), is very similar to that of the previous frame.



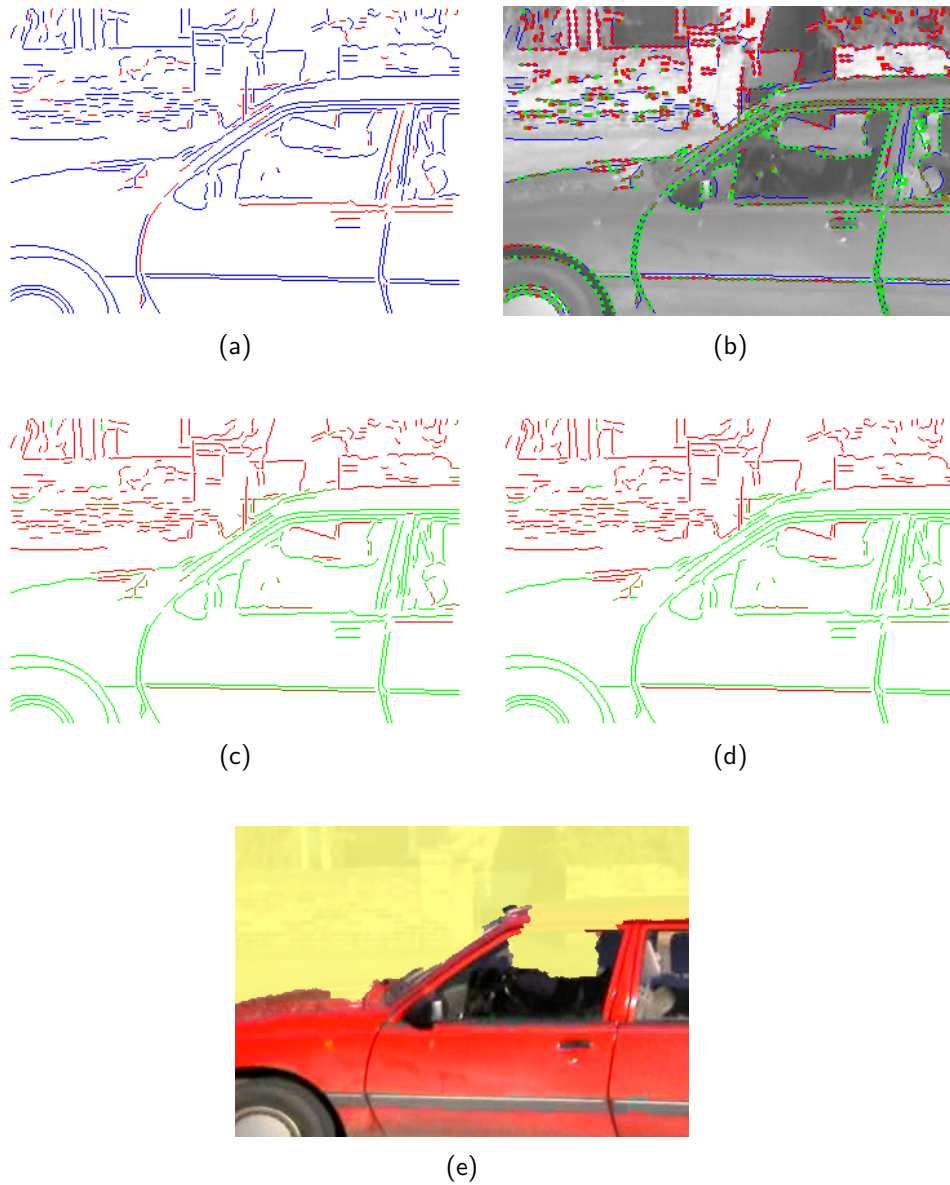


Figure 6.15: *Car segmentation of the next frame.* (a) Detected edges (blue) augmented by propagated edges (red); (b) Sample points propagated from previous frame. New sample points are created to fill any gaps; (c) Edge motion probabilities between the second and third frames; (d) Cumulative edge probabilities over both frames; (e) Segmentation of second frame.

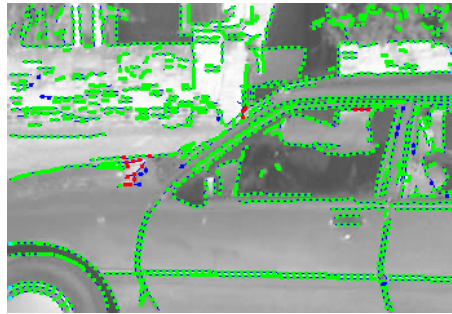


Figure 6.16: *Occluded sample points in the Car sequence.* Sample points in the second frame, some identified as occluded by the foreground (red) and some as off the image in the next frame (blue).

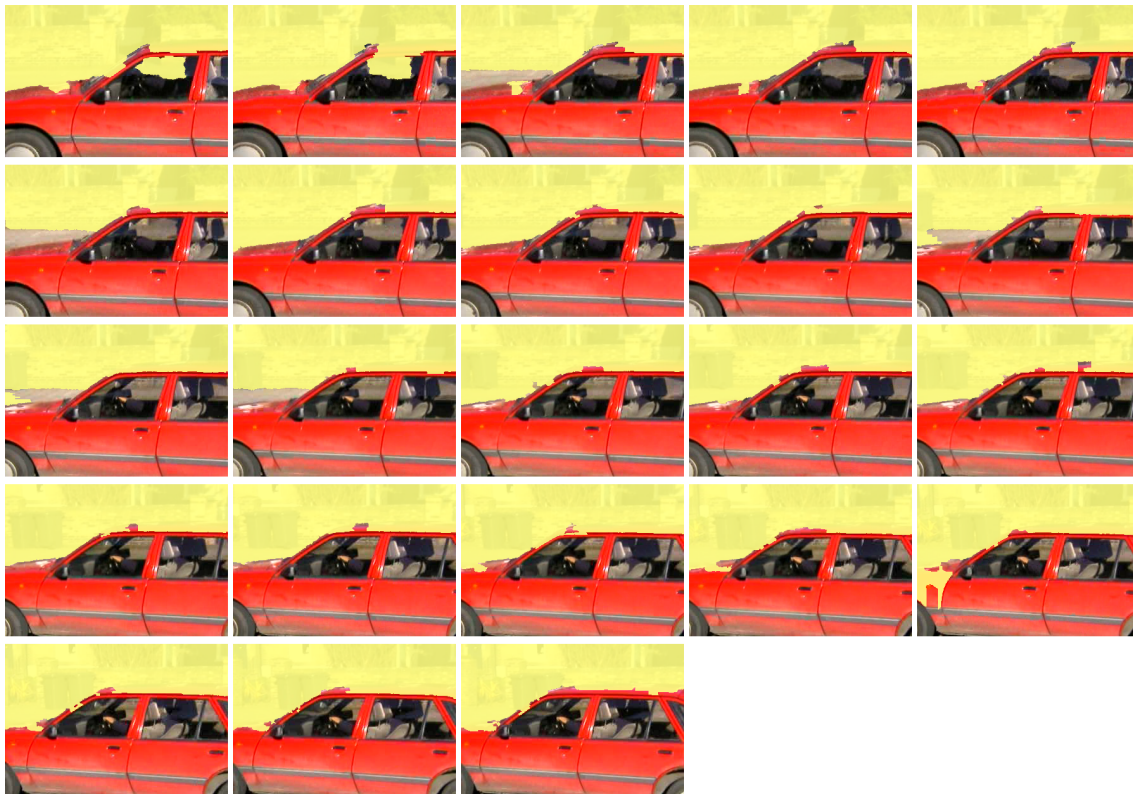


Figure 6.17: *Segmentation of the Car sequence.* Segmentation of twenty-three consecutive frames. (A software bug prevented further frames being processed in this case.)

Over the sequence, as shown in Figure 6.17, the segmentation continues to be very good, with the only consistent error being the roof of the car which, with its reflections, continues to be assigned to the background. In the occasional frame, the area of pavement above the car bonnet is also segmented with the car. Since this is only usually identified as background thanks to one horizontal edge, it will be mislabelled if that edge is labelled with the car’s motion, as occasionally happens. Also of note is the fact that as edge features are lost from the view through the car’s window, the regions there can no longer be identified as background. In general, the car’s image motion does not present any major problems to the affine motion model, and the only errors are due to the ambiguities of labelling the horizontal edges and the reflections. An excellent segmentation is produced throughout.

### 6.6.5 Ensemble results

The multiple-frame deformable segmentation has been tested on each of the thirty-four image sequences shown in Appendix D. The segmentation using only two frames has already been discussed in Section 5.8; here the segmentations after three and ten frames are considered for each sequence.

#### Segmentation of a second frame

The segmentation of the second frame in each sequence is performed in the same manner to that of the first frame, except that the edge probabilities used are those accumulated over both frames. It is therefore to be expected that sequences which performed well in the two-frame case also perform well for the next frame, and this is indeed what is observed. All of the sequences for which EM gave good solutions in the two-frame case also gave good solutions for the next frame. Over the thirty-four sequences, all but eight have a greater number of edges labelled correctly when using a third frame and, even including these, there is a mean increase of 5% in the number of correct edges. Those sequences which do show a drop in the edge labelling have only a small decrease (these are sequences where the labelling is uncertain throughout).

The **Nick** sequence shows a startling improvement between frames.<sup>3</sup> In the two-frame case, EM failed to converge because of the small number of edges and the non-affine motion, but between the next two frames the motion is well modelled, and a good labelling is found. The **Driven2** sequence again failed to converge well between either pair of frames, but does edge a little closer to a good labelling as

<sup>3</sup>See Appendix D for the results from the second and tenth frame for each sequence.

Ranking	Pixels correct	Frequency (%) at frame number		
		1	2	10
Excellent	> 95%	11 (32%)	14 (41%)	11 (32%)
Good	85–95%	8 (24%)	10 (29%)	12 (35%)
Reasonable	75–85%	3 (9%)	4 (12%)	6 (18%)
Poor	50–75%	5 (15%)	4 (12%)	2 (6%)
Failure	0–50%	7 (21%)	2 (6%)	3 (9%)

Table 6.3: *Percentage of pixels correctly segmented over multiple frames.* Overview of segmentation performance over the thirty-four test sequences. Any figure over 85% is a good segmentation; those over 95% are almost flawless.

the sequence progresses. The only sequence to perform significantly worse is the **Horizon2** sequence, with its large projective motion.

The use of an extra frame generally increases the confidence in the layer ordering, and in very few sequences is the opinion of the correct layer ordering changed from that in the first frame. Of the thirty-four sequences here, six sequences changed layer ordering—all from the incorrect to the correct layer ordering

Table 6.3 extends the earlier table to include the results from frames 2 and 10. It can be seen that with the use of the extra frame, the segmentation of frame 2 is generally improved. Twenty-four of the sequences (71%) have a higher number of pixels correct when using this extra frame and the same number have good, or excellent segmentations. These are excellent results, and the segmentation only genuinely performs poorly in four cases: **FlashGordon2** features a very large motion, **Horizon1** and **Tweenies** have significantly non-affine motions, which this current system does not pretend to be able to model, and **ITN** has no edge features in the background.

### Segmentation of the tenth frame

The final set of results in Appendix D are those of the tenth frame of each sequence. By this point in many sequences the motion has changed significantly from any original affine assumption. In addition, while sample points will have had ten frames to gather evidence, there is no guarantee that many of the original sample points will by now be visible, or will have been successfully propagated.

Looking at the results presented in Table 6.3 it can be seen that the results continue to be very good, with the only significant change (compared with the performance after the second frame) being that a few sequences slip from ‘excellent’ to merely ‘good’. This slight down-turn in the high-end results is due to the motion problems highlighted in the **Foreman** case: over a longer sequence, foreground objects can either cease to move, or can begin moving in a highly non-affine manner. The

first case would be dealt with by selecting the best number of motions in each frame. However, the manner in which events such as this ‘sleeping person’ problem [150] should be treated depends on the semantics of the scene and the correct treatment of these cases requires some higher-level understanding. The second case, of sudden changes in the direction or type of motion requires more sophisticated tracking technology. Once the boundary of the object has been identified (as is frequently the case within a few frames), the problem essentially becomes one of tracking this boundary. Schemes based on the CONDENSATION algorithm [20] have proved to be highly successful at tracking even vigorous motion.

The majority of sequences, however, continue to move according to the image motion assumptions, i.e. affine with perhaps a few pixels of non-parametric deformation. These perform excellently under the edge-based approach. By the tenth frame, a large number of edges have been collected and propagated, and typically around 80% of these would be labelled correctly according to their cumulative detected motion. Even difficult subjects such as the lion in *Cats1* or the boat in the *Coastguard* sequence have an excellent edge labelling by the tenth frame, and one which is vastly improved over that of earlier frames.

## 6.7 An application: Background mosaicing

Image mosaicing is the process of piecing together various different images of a scene to produce one large-scale image. This is a process which has long been performed, for example, in aerial photography [41]. There has also been a recent flurry of interest in image mosaicing for *image-based rendering* [87], to give a computer user the impression of being immersed in a 3D environment by using images to create a continuous field of view around a given point.

In motion analysis applications a number of authors advocate a mosaic-based approach to video coding and motion description. In motion description applications, a mosaic can be used to display a ‘visual summary’ of a sequence [57, 73]. A single mosaiced image is created of the backdrop to the sequence and then the motion of the foreground objects is overlaid on this as a series of tracks. A mosaic gives a common frame of reference within which the foreground motion may be described and analysed. For video compression the single background image may be transmitted once and then each individual frame can be described by the position of the current frame in the mosaic, plus the foreground objects and any correction terms [75, 121]. This process is part of the MPEG-4 standard [82, 129].

Any of these applications could be built on top of the video segmentation scheme



Figure 6.18: *Mosaic of the background to the Car sequence.* The background segmentations of frames 490-507 (shown at the top of the figure), transformed to a common co-ordinate frame using the background motion estimates. The red patches are areas of the car which, due to reflections, were mislabelled as background.

presented in this dissertation. This section presents a few sample mosaics generated automatically by the multi-frame implementation introduced in this chapter, as a demonstration of the background segmentations possible. It also provides a qualitative test of the background motion estimation, since without an accurate background motion estimation, the images cannot be stitched together correctly.

### 6.7.1 Implementation

A simple mosaicing implementation is demonstrated here. As each frame in a sequence is segmented, the pixels identified as background are extracted. These are transformed into the mosaic image, in the coordinate frame of the first frame, using the estimated (affine) motion between the current frame and the first. Over a sequence of frames, this is simply the matrix product of the inter-frame motions. For simplicity, this is only performed to pixel accuracy.

It is likely that each pixel in the mosaic image will receive contributions from several different frames. The displayed pixel colour in these cases is taken to be the *median* colour for that pixel, independently in red, green and blue.

### 6.7.2 Examples

#### Car sequence

Figure 6.18 shows the backdrop to the *Car* sequence over nineteen frames. As seen earlier, the top of the car is not usually segmented as foreground due to the reflec-





Figure 6.19: *Mosaic of the background to the Simpsons sequence.* The background segmentations of frames 77–94 (shown at the top of the figure), transformed to a common co-ordinate frame using the background motion estimates.

tions, and the same is sometimes true of part of the bonnet, so these are unfortunately included in the mosaic. The mosaic appears to be accurate, apart from a slight ‘tearing’ visible across the white lines at the top. These may be due to only using pixel accuracy for the mosaic, but it is also known that the motion estimation in this sequence does not converge particularly well—the EM process takes a very long time to converge, continually making small adjustments. However, on the evidence of this mosaic the error is at most one pixel, and only in some parts of the frame.<sup>4</sup> It is also interesting to note the difference in colour of the background pixels which were only seen through the tinted glass of the car window.

### Simpsons sequence

The background mosaic for the **Simpsons** sequence is shown in Figure 6.19 (see Appendix D for some of the individual segmentations). Of interest in this mosaic are the ghosts of the edges of the foreground, particularly visible above the garage.

<sup>4</sup>It should be remembered that an *accurate* motion estimate (i.e. sub-pixel) is not required for this motion segmentation scheme. The correct motion must just fit *better* than the other choices.

These indicate that, even with cartoons, it is difficult to extract the boundary on only a whole-pixel basis. The boundary in the world is unlikely to be imaged exactly at a pixel boundary, particularly in the  $320 \times 288$  MPEGs considered here. Consequently, the ‘boundary’ pixel will be a blend of both the foreground and the background colour. It is these pixels which are observed in this mosaic. However, as far as the motion is concerned, the mosaic is again good, with errors of at most one pixel. As mentioned earlier, these may be due to the mosaic generation process rather than the estimated motion.

## 6.8 Summary

In this chapter techniques have been developed which look at a sequence of frames. The tracking of edges, or sample points, across a series of frames enables evidence to be accumulated and a more accurate edge labelling to be performed, reducing ambiguity. Once the motion between each frame has been determined, a segmentation of each frame in the sequence may also be performed. This chapter introduced a *templated* segmentation, which is appropriate for short sequences, and also a *deformable* segmentation. The latter has been evaluated on a range of test sequences, with excellent results.

The techniques developed here consider only a causal improvement of labellings—using previous frames to assist the current one. The segmentation may also proceed in a non-causal manner by using future frames as well, either directly from a recorded sequence or by slightly delaying a real-time sequence. It is these segmented sequences which would be particularly useful for higher-level motion analysis applications. This and other possible improvements are suggested in Chapter 8.

The use of multiple frames resolves many of the problems identified in the evaluation of the two-frame algorithm (Chapter 5). However, thus far only two motions have been considered: the background and one foreground object. In this chapter, correctly modelling the number of motions in the frame has been identified as a problem in some test sequences. The next chapter considers extending the edge-based framework to segment an arbitrary number of motions.



---

## Extension to multiple motions

---

### 7.1 Introduction

While a wide number of useful video sequences feature only one moving object, a truly general video segmentation system must be able to detect and segment as many different moving objects as there are in the scene (and identify when there is no motion). The edge-based framework developed in Chapter 3 is applicable to any number of motion layers, and this chapter presents a preliminary investigation into extending the implementation to the multiple-motion case. In particular, new algorithms are developed to improve the edge labelling accuracy. Experimental results are presented for two three-motion sequences.

Segmenting multiple motions is a far more difficult problem than the two-motion case. With more motions spread throughout the frame, there are fewer edges with which to estimate each motion. With more motions to choose between, edges can be assigned to a particular model with less certainty. Both of these provide the Expectation-Maximisation algorithm with a far harder task in estimating the motions and edge labels, and the EM stage is found to have a large number of local maxima. Avoiding these maximum has required the development of a new EM initialisation scheme. Connected with the EM optimisation is the question of how many motion models should be fitted to the data. The approach adopted here considers solutions with different numbers of motions and selects the most plausible; this is done using the Minimum Description Length principle [116], as described later.

With more uncertainty in the edge labels, the labelling of the regions becomes more difficult, as does identifying the correct layer ordering. Although more difficult, it will be seen that this region labelling stage still performs reasonably, due to the spatial coherency enforced by the Markov Random Field approach. This region labelling can be used to improve the edge labelling, using the spatial reasoning of the region labelling to constrain the possible edge labels in an Expectation-Maximisation-Constrain loop. The region segmentation is also improved by enforcing the constraint that each foreground object should be represented by a contiguous group of regions. Each of these modifications is also presented in this chapter.

## 7.2 Recursive Splitting EM

The EM algorithm is guaranteed to converge to a maximum, but there is no guarantee that this will be the global maximum [43]. This local maximum problem is common in many iterative schemes: by only taking local measurements and making a small step in a favourable direction, large-scale features are missed. If the iteration is initialised at some distance from the global maximum and there are other, smaller, maxima on the route to this maximum, the iteration can easily ascend one of these local maxima and become trapped there, finding no local improvements. The best solution to these local maxima problems remains an open question.

One suggested solution to this problem is to attempt to remove the local maxima. For local maxima problems, Blake and Zisserman [21] proposed the Graduated Non-Convexity (GNC) algorithm, which approximates the function to be minimised by a smoothed version with a guaranteed single maximum (i.e. it is convex). The maximum of this will be close to the global maximum for the original function, and this then is a good starting point for a slightly less smoothed version of the function. The smoothing is gradually removed, calculating the new maximum at each point, so that eventually the real function is maximised. By starting at a point close to the global maximum of each function, the maximisation should not be caught in any local maxima. Ueda and Nakano [154] proposed this form of solution for EM, in their Deterministic Annealed EM algorithm (DAEM), which performs the smoothing by increasing the variance of each model. The variance is slowly restored to its true value as the EM progresses in order to give the final solution. While this scheme is effective at avoiding local maxima in the vicinity of the global maximum, it does not solve all local maximum problems.

A common problem occurs in multiple-model situations when too many models are initialised in one part of the space and too few in another. It would be desirable

for these to be redistributed optimally, but changes by local steps are not usually possible as this would involve passing through positions with a lower likelihood. Even with the smoothing of the DAEM algorithm, this is not usually possible.

To resolve this, Ueda et al. [155] introduced a Split and Merge EM algorithm. This initialised a pre-determined number of models and allowed them to converge to an initial solution. This solution was then analysed to determine whether any two models could be merged (if they were similar), and whether any model was under-fitting, and so should be split into two. If so, these motions were merged and split respectively (so that the total number of models remained constant), and EM again run from the new locations.

This section of the dissertation presents a similar approach, developed jointly with Tom Drummond and Rob Fergus (see also [52]). This scheme differs from that of Ueda et al. in that only splitting is performed—it begins with only one model and recursively adds more models. This enables it to be integrated with a scheme to select the best number of models, since splitting continues as long as it improves the interpretation of the data.

The Recursive Splitting EM (RSEM) algorithm presented here can be considered to be akin to the multi-resolution techniques common in image matching algorithms (e.g. [9, 78]). First the gross arrangement is estimated by fitting a small number of models and then these are split to see if there is any finer detail that can be fitted (see Figure 7.1). By proceeding in this fashion, it is guaranteed that all the models that are being fitted are initialised in sensible locations.

### 7.2.1 Initialising an extra model

In order to determine how to add an extra model to the data, it is worth considering what happens if too few models are fitted. For example, Figure 7.2(a) shows a set of data best explained by three models. If only two models are fitted then there are two likely outcomes (depending on the initialisation):

1. One (or both) of the models adjusts to absorb data which should belong to a third model (Figure 7.2(b)).
2. If the models are fitted with a robust estimator [70, and Appendix A], then the data belonging to the third model could be discarded as outliers and the other two motions fitted well (Figure 7.2(c)).

Given these likely cases, it is clear that there are a number of possibilities for initialising an  $(n+1)$ -motion solution given an  $n$ -motion solution. One of the models

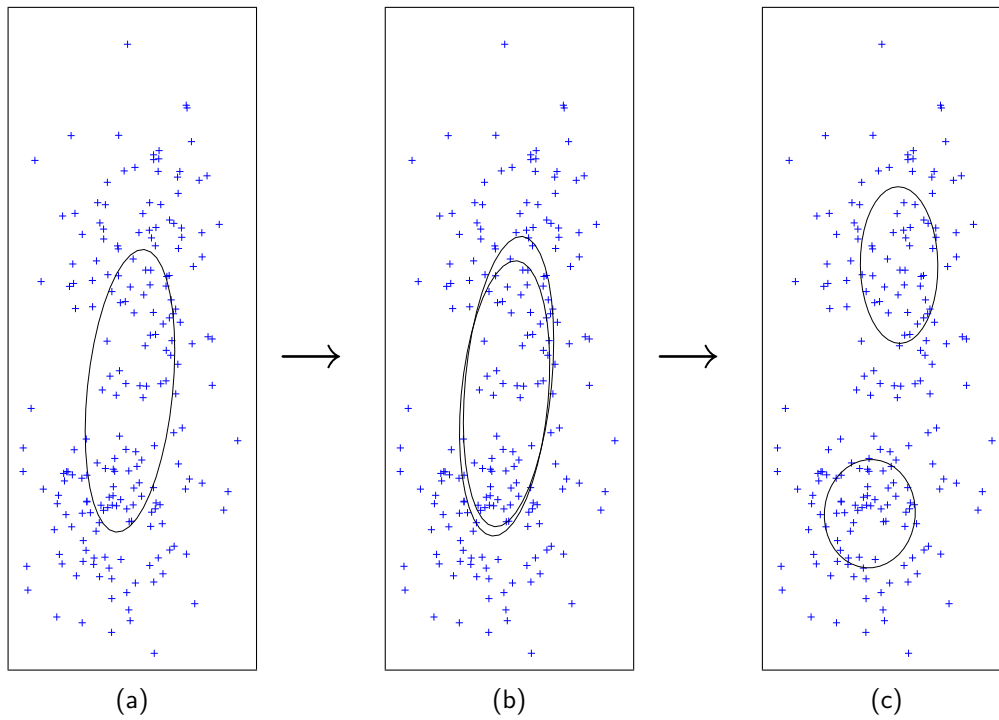


Figure 7.1: *Initialisation by splitting*. Random samples taken from two multivariate Gaussian distributions. (a) Fitting one model; (b) Random initialisation of two models by perturbation from (a); (c) Models after EM.

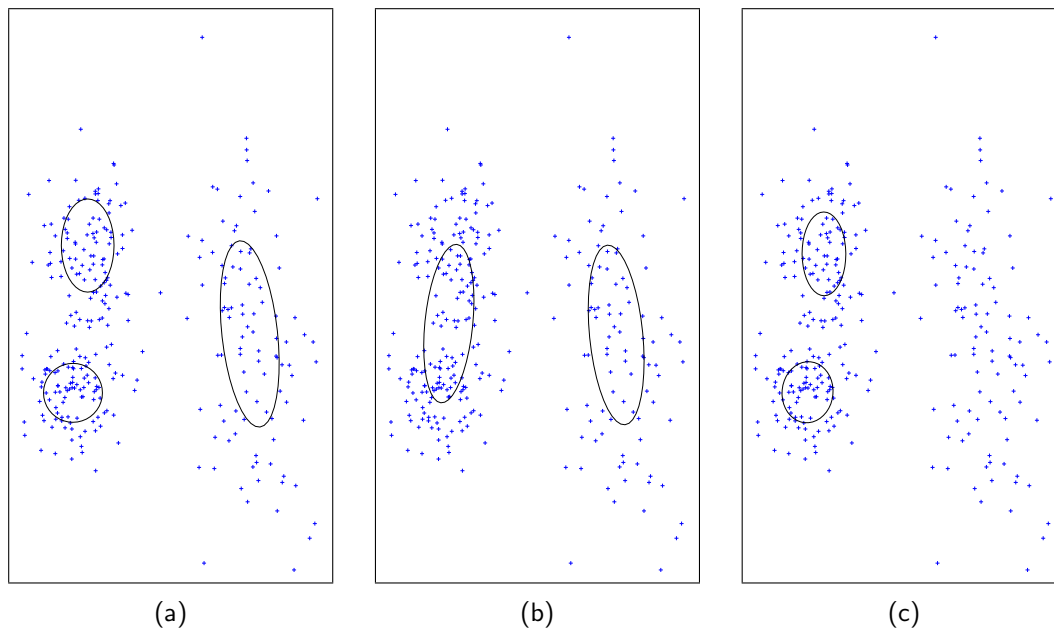


Figure 7.2: *Initialising with too few models*. If the data are best explained by 3 models (a), but only two models are fitted then two results are likely: either (b) the models absorb data points which should belong to an extra model or (c) a large set of data points are discarded as outliers.

could be split into two to reveal any smaller-level structure, or the outliers could be considered as a separate model. The RSEM algorithm tries each of these possible initialisations at each stage, selecting the best one before continuing.

### Splitting

Experience of EM in both this dissertation's application, and the work of Fergus [52], has shown that robustly fitting two models to data by EM is relatively untroubled by the problem of local maxima, and this is key to the splitting process. Given a model  $i$  which needs to be split (with parameters  $\Theta_i$ ), two new models  $j$  and  $k$  are initialised with parameters

$$\Theta_j = \Theta_i + \epsilon \quad \Theta_k = \Theta_i + \epsilon' \quad (7.1)$$

where  $\epsilon$  and  $\epsilon'$  are some small random perturbations. One simple way to achieve this is to take all the data points for which model  $i$  is the most likely and divide them randomly into two groups. Models  $\Theta_j$  and  $\Theta_k$  can then be estimated, one from each group, by standard maximum likelihood methods.

From these initial estimates, the models are updated to fit the local data by performing EM using only these two models and the local data (once again, the data for which model  $i$  was most likely). Figure 7.1 showed this process, starting with one model, then two random perturbations, and finally the solution after the local EM stage. After this local EM stage, model  $i$  is replaced in the global list of models by the split pair  $j$  and  $k$ . EM is then performed upon the global list (i.e. the previous motions, plus the split motion) to find the global optimum. This splitting process is attempted for each of the original motions.

### Outlier model

The other possibility is that one of the motions has been ignored through the use of robust methods. Taking the set of data points which are deemed to be outliers to the existing models (according to some suitable error threshold), an extra model is fitted to these points. This solution is then optimised by performing EM over the augmented set of models.

### Selecting the best $n + 1$ model solution

After EM has been performed from each starting point (trying a split of each original model, and the outlier model) the final solution given by each may be compared.

The solution with the highest likelihood (the value that EM is optimising) is taken as the solution for  $n + 1$  models. This solution may be further split to test whether further models would be appropriate.

## 7.2.2 Determining the best number of models

Increasing the number of models is guaranteed to improve the fit to the data, and increase the likelihood of the solution (as long as they converge to the global optimum). However, this should be balanced against the principle that simple solutions are the best, commonly referred to as Occam's Razor.<sup>1</sup>

The common method for imposing this principle is to define a cost function which decreases as the likelihood increases, but increases with model complexity. There have been a large number of these suggested in the literature, variously justifying the cost functions in terms of information theory (entropy), or the related field of coding theory, or Bayesian statistics. However all these approaches result in very similar expressions (for a survey, see Torr [144]).

The cost function used in this work is derived from a coding standpoint, following Rissanen's Minimum Description Length (MDL) principle [116]. This is popular in motion segmentation approaches, for example in work by Ayer and Sawhney [4], Brady and O'Connor [27] and Elias and Kingsbury [48]. This considers the cost of encoding the observed image motion in the minimum number of bits, by coding the model(s) and then, for each data point, any residual error from the model. A large number of models or a large residual error both give rise to a high cost.

The cost of encoding the model consists of two parts: first the parameters of each model, and second the labelling for each edge. If each number in the model is to be encoded to 10-bit precision (a typical figure) and each model has  $n_d$  parameters, the cost of encoding the models is  $10n_d n_m$  (where  $n_m$  is the number of models). To label each data point requires that each has one of  $n_m$  labels. In binary, this costs  $\log_2 n_m$  bits, so for  $n_e$  data points this costs a total of  $n_e \log_2 n_m$  bits. Finally, the residual errors must also be encoded. From information theory [125], the cost for an optimal encoded size (in bits) is equal to the total negative logarithm (to base 2) of the data likelihood,  $\mathcal{L}_e = \prod_i P(e_i | \Theta)$ . The total cost is thus given by:

$$C = 10n_d n_m + n_e \log_2 n_m + \sum_e \log_2 \mathcal{L}_e \quad (7.2)$$

The cost  $C$  can be evaluated after each attempted initialisation, and the smallest cost indicates the best solution and the best number of models. Equation 7.2 is

---

<sup>1</sup>William of Occam was an 14th-century English philosopher and theologian who wrote '*Pluralitas non est ponenda sine neccesitate*' ('Entities should not be multiplied unnecessarily').

completely general apart from one tunable parameter, the number of bits per model parameter. Ten bits is a typical figure and is successfully used in both [52] and here.

Table 7.1 gives an overview of the complete initialisation and model selection algorithm.

### 7.2.3 Implementation for edge-based motion segmentation

The RSEM algorithm has thus far been described in general terms. It is proposed as a general solution to the local maximum problem in EM, and has proved to be an effective solution in both of its current applications: fitting Gaussian mixture models in [52], and in fitting more than two parametric edge motions in the work described in this dissertation. This section describes the latter implementation in more detail.

In this implementation, one motion is first fitted and the cost of this solution evaluated according to (7.2). For a 2D affine model, which is usually used, the number of parameters  $n_d = 6$ , and the data to be coded are the edge labels and likelihoods. Then two models are fitted and the cost again evaluated.

If two motions are better than one, three motions must be tried. To begin splitting, the edges are separated into three groups: firstly the outliers are detected—these are edges for which the probability of a correct match  $P(\mathbf{e} = i | \mathbf{D}_i)$  is less than 0.5 under each motion (equation (4.36)). The remaining edges are then separated according to their most likely motion i.e. if  $P(\mathbf{e} = 1 | \mathbf{D}_1 \mathbf{D}_2) > 0.5$ , the edge is motion 1 (equation (4.31)).

Given these groups, three initialisations and trials of EM are run:

1. Calculate the motion of the outlier edges, add it to the list of motions and run EM.
2. Take the set of edges which best fit motion 1, split these into two random groups and run EM on these to fit two motions. Replace motion 1 with these two motions, and run EM.
3. As initialisation 2, but splitting motion 2.

In each case the cost is calculated once the three-motion EM has converged. If the smallest cost from among these three is less than the cost of the 2-motion solution then that 3-motion solution is stored as the current best model. The process is then repeated to try four motions, and then further motions.

Figure 7.3 demonstrates the motion-fitting process on a test sequence with three motions. First two motions are fitted (Figure 7.3(a)), then the three different initialisations are tried (Figure 7.3(b) shows the splitting of motion 1), and finally the

- Fit one model to all data ( $n = 1$ )
- Calculate cost  $C_1$
- Repeat
  - Set number of models  $n = n + 1$
  - Generate possible initialisations  $I_{1\dots n}$ :
    - $I_i$  The set of existing models, but with model  $i$  split into two ( $i = 1 \dots n - 1$ ). Splitting is performed by random assignment and then local EM.
    - $I_n$  The set of existing models plus a new model created to fit the outlier set
  - For each initialisation  $I_i$  ( $i = 1 \dots n$ )
    - \* Do EM
    - \* Calculate cost of solution  $C_n^i$
  - Evaluate best  $n$ -model solution  $C_n = \min_i C_n^i$
  - If  $C_n \geq C_{n-1}$ 
    - Report motion corresponding to  $C_{n-1}$  and terminate.
  - Else
    - Goto Repeat

Table 7.1: *Recursive splitting EM*. Overview of algorithm to repeatedly initialise EM from different (plausible) starting points, with increasing numbers of models, and select the best one.



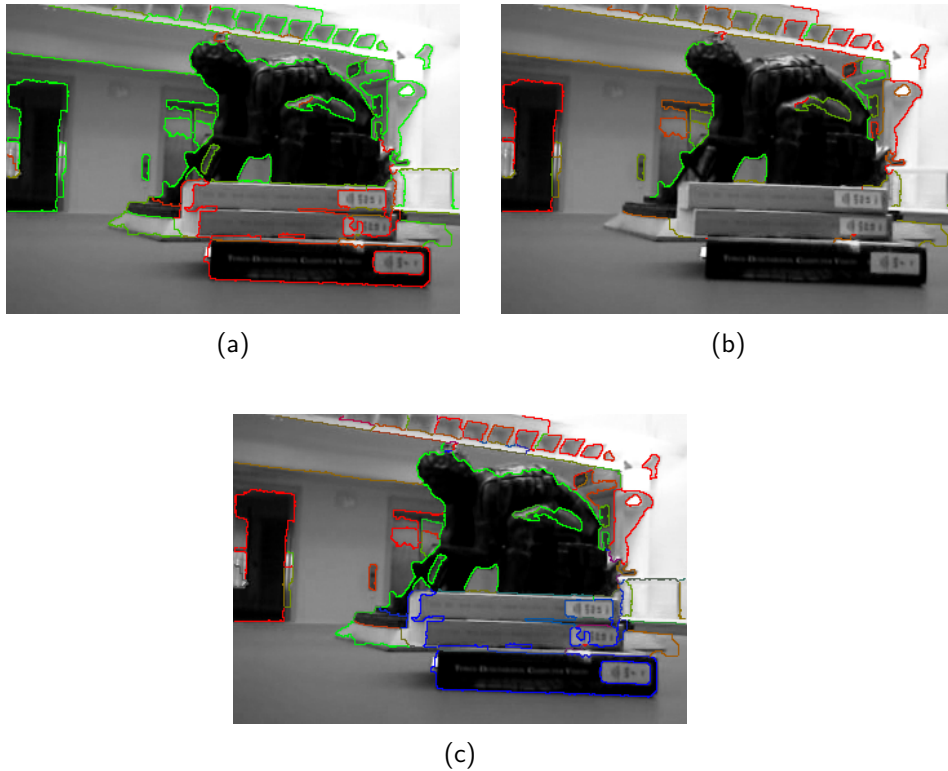


Figure 7.3: *Fitting three motions.* The books, statue and background each have a different motion. (a) Two motions are fitted to the sequence by EM—note that one motion (green) has fitted both the background and the statue; (b) The set of edges belonging to motion 1 are taken and two motions fitted to these by EM. This separates the statue from the background (two other initialisations are also tried, but these have a higher cost and are not shown); (c) The two new motions, and the original motion 2 are taken as initialisations for a three-motion solution and EM is performed over all edges. The edge probabilities are now displayed as a blend between three colours: red, green and blue, representing motions 1, 2 and 3 respectively.

best three-motion solution is selected. The cost associated with both two- and four-motion solutions is higher (see Section 7.7 for full results). These motion estimates, and the associated edge probabilities are sufficient for a reasonable region labelling.

### 7.3 Region labelling under multiple motions

Given a set of edges labelled with their motion probabilities, as is provided by the RSEM algorithm, a region segmentation may be performed and the regions labelled by simulated annealing in the same manner as described in Chapter 4. In the case of three motions this requires an optimisation over six possible layer orders, so the annealing stage must be repeated for each of these six possibilities and the maximum selected. For more motions this increases combinatorially, since

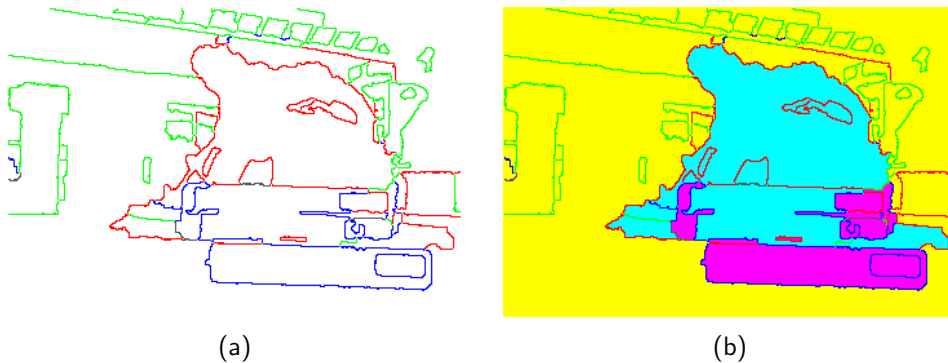


Figure 7.4: *Three-motion edge probabilities and region labels.* (a) The edge labels given by EM (solution of the RSEM algorithm of Section 7.2). Here the edges are labelled according to their most likely motion: red, green and blue represent motions 1, 2 and 3 respectively; (b) Maximum *a posteriori* region labelling and layer ordering. In increasing depth order, the layers are coloured magenta, cyan and yellow.

for  $n$  motions there are  $n!$  layer ordering possibilities. Fortunately, in real world sequences, there are rarely many independently moving objects (one or two are typical), and it is reasonable to assume that  $n$  is small.

Under each layer ordering, an initial guess is made to the labelling (this time each region may have one of  $n$  different labels). This is done in a similar manner to the two-motion case i.e. according to the majority edge labelling. Each region in turn is then considered for a new label, and the probability of each label calculated. Once again this probability is the product of the implied edge probabilities and the region prior. The implied edge probabilities may be used in exactly the same manner as before, but the region prior should, strictly speaking, be a joint distribution over the boundary fraction shared with each of the other two motions. This once again gives a combinatorial explosion and so, for simplicity, the distributions are assumed independent. This assumption is exact for the majority of regions, which are only bounded by edges of one or two different labels. The same annealing schedule, (4.42), is used and the regions are labelled by a Monte Carlo approach as before. Figure 7.4 shows a typical edge and region labelling. It can be seen that even with a noisy edge labelling, the region labelling is reasonable.

## 7.4 Global optimisation: EMC

A complete motion segmentation is determined via two independent optimisations, which use edges as an intermediate representation: first the best edge labels and motions are determined, and then the best region labelling given these edges. It has

thus far been assumed (see Section 3.5.1) that this gives a good approximation to the global optimum, but unfortunately this is not always the case, particularly with more than two motions.

In the first EM stage the edges are assigned purely on the basis of how well they fit each motion, with no consideration given to how likely that edge labelling is in the context of the wider segmentation. There are always a number of edges which are mislabelled—increasingly so with more motions—and these can have an adverse effect on both the region segmentation and the accuracy of the motion estimate.

One possible solution is to reinstate priors on the edge probabilities (these were assumed to be constant, and equal, in Chapter 4, particularly (4.26)). The prior labelling for an edge could be expressed as a function of the labellings of the edges surrounding it. However a simpler approach, and one also followed by Brady and O'Connor [27], is to introduce an extra constraint step into the EM algorithm, thus making the stages Expectation-Maximisation-Constrain, or EMC.

Acknowledging that the edge probabilities provided by the E-stage are noisy, the EMC algorithm uses the logical constraints imposed by a region labelling to provide a *discrete*, constrained edge labelling, where each edge is labelled with a probability of 1 for one motion and 0 for the others. Figure 7.4 showed the edge labels after the standard RSEM algorithm, and the resulting region labelling. This region labelling implies an edge labelling, shown in Figure 7.5. This implied edge labelling is self-consistent and, while not perfect, is better than that provided without any constraints. The EMC algorithm therefore uses these implied, discrete, edge labels in the M-step rather than the edge probabilities, giving the following steps, which are iterated as shown in Figure 7.6:

**Expectation** Estimate the edge label probabilities given the motions

$$P(\mathbf{e}|\Theta_n\mathbf{D}) \quad (7.3)$$

**Constrain** Calculate the most likely region labelling given these edge probabilities

$$\max_{\mathbf{R}, \mathbf{F}} P(\mathbf{R}, \mathbf{F}|\Theta\mathbf{D}) = \max_{\mathbf{R}, \mathbf{F}} \sum_{\mathbf{e}} P(\mathbf{R}, \mathbf{F}|\mathbf{e}) P(\mathbf{e}|\Theta\mathbf{D}) \quad (7.4)$$

$$= \max_{\mathbf{R}, \mathbf{F}} P(\mathbf{e}(\mathbf{R}, \mathbf{F})|\Theta\mathbf{D}) P(\mathbf{R}) \quad (7.5)$$

and from this the set of definite edge labels

$$\hat{\mathbf{e}} = \mathbf{e}(\mathbf{R}, \mathbf{F}) \quad (7.6)$$

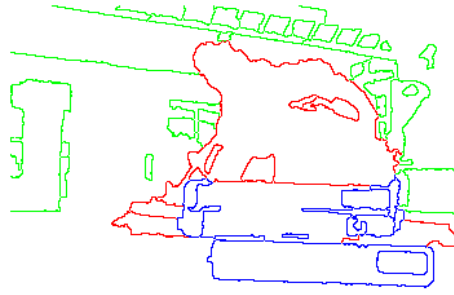


Figure 7.5: *Constrained edge labels.* Edge labels implied by the region labelling of Figure 7.4(b). Compare this with the original labelling of Figure 7.4(a) and note that several of the original edge labels were inconsistent with this region labelling.

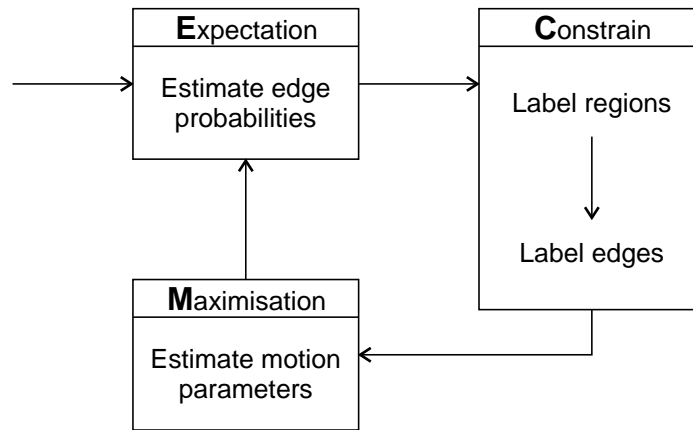


Figure 7.6: *Overview of the EMC algorithm.* After the edge probabilities are calculated, the region labelling is used to constrain the edge labels to definite, consistent, labellings. These constrained edge labels are used to estimate the motions.

**Maximisation** Use the implied edge labelling  $\hat{\mathbf{e}}$  to calculate a new set of motions

$$\arg \max_{\Theta_{n+1}} \sum_{\mathbf{e}} \log P(\hat{\mathbf{e}}|\mathbf{D}|\Theta_{n+1}) P(\hat{\mathbf{e}}|\Theta_n \mathbf{D}) \quad (7.7)$$

The iteration is continued until the likelihood of the region labelling in the C-step (7.5) is maximised. This region labelling, having been calculated from the most recent edge labels, is the final solution.

The EMC loop is best considered as a final global optimisation stage. Once again, as an iterative scheme, its initialisation is an important consideration. The constraints (i.e. a sensible segmentation) cannot be applied until the edge labels are reasonable. It is also time-consuming since each iteration requires the maximisation of the region labelling (via simulated annealing), which is itself an iterative scheme. As a result, starting from as close to the solution as possible is highly desirable and the EMC loop is performed as a separate refinement stage after the original EM (or

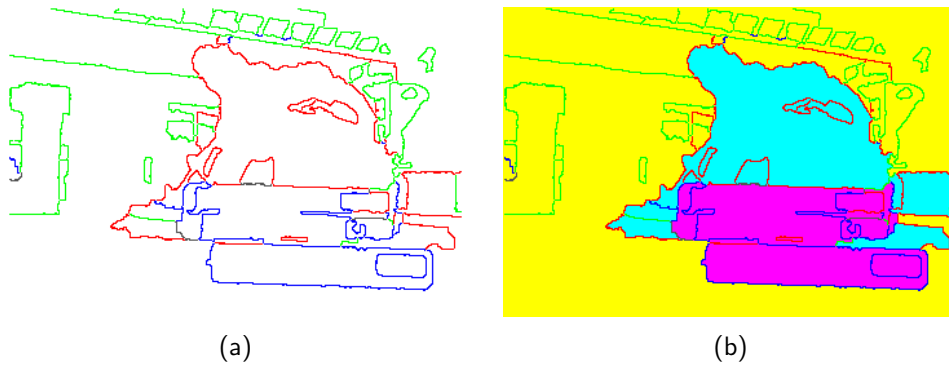


Figure 7.7: *Example EMC solution.* (a) Edge probabilities after EMC loop; (b) Region labelling using EMC edge probabilities. Note the improvement to both, compared with the original labelling in Figure 7.4.

RSEM) loop. EMC must also be performed for each possible layer ordering, as it is only with a hypothesised layer ordering that the region labelling may be maximised. The EMC process described here therefore does not scale to large numbers of motions, but only small numbers of motions are likely in real-world situations.

Figure 7.7 shows the edge labels and most likely region segmentation for the example sequence after the completion of the EMC loop. The optimisation takes about a minute to perform on a 300MHZ Pentium II (for images of  $352 \times 288$  pixels in size). The final edge and region labellings show a small, but significant, improvement over the standard EM solution of Figure 7.4.

## 7.5 ‘One region’ constraint

The Markov Random Field used for the region prior  $P(\mathbf{R})$  only considers neighbouring regions, and does not consider the wider context of the frame. This makes the simulated annealing efficient (since only local changes need to be considered), but does not enforce the belief that there should, usually, be only one connected group of regions representing each foreground object. It is common for some isolated background regions to be mislabelled as foreground when the edge probabilities are noisy (as seen in the *Coastguard* sequence, Figure 6.14), and this is a particular problem when the number of motions increases. Not only is it easier for this mislabelling to occur, but this error is also compounded if this labelling is then used in a feedback loop, such as the EMC described above, which would encourage the edge motions and probabilities to support this mislabelling.

It is possible to include a higher-level clustering term to the region prior, for example a measure of compactness, or simply a prior on the number of connected

components. One trivial change would be to set the prior for an isolated region (i.e. the case of  $f_i = 0$  in Section 4.6.2) to zero. This would prevent this scenario from being considered during the annealing stage. However, this is undesirable as it may be necessary for the annealing process to pass through such a state in order to reach the global optimum. Such a solution would also not prevent an isolated pair of neighbouring regions from being created and maintained.

Instead, a simple Procrustean approach is proposed as a post-processing stage.<sup>2</sup> After the annealing stage has converged and produced a labelling, a connected-component analysis is performed to determine the number of independent groups at the depth layer closest to the camera. (This is straightforward as details of a region's neighbours are already required for the MRF-style region prior.) Given these groups, region labellings are hypothesised which label all but one of these groups as belonging to a lower layer (i.e. further back). The most likely of these 'one object' region labellings is the one that is kept. This process is repeated for each layer in turn, working through the layers in order of depth until each layer apart from the background (the layer furthest from the camera) has been edited to leave only one group.

This approach enforces the hard constraint that there shall be only one simply-connected object at each layer. While not completely general (a probabilistic prior would be preferable), it is true in many cases and is simple and efficient to implement. This 'one region' processing stage is included within the EMC loop (Section 7.4) such that the edge labels which feed back to the motion estimation stage are consistent with this constraint.

## 7.6 Implementation overview

To provide a reliable segmentation of more than two motions using two frames it is necessary to use all of the extensions proposed in this chapter. Figure 7.8 provides an overview of the general algorithm.

Edges are found in the frame to be segmented and the mean motion is estimated. The system then enters the RSEM loop (Section 7.2), which repeatedly tries splitting the motion first into two, then three, and then further motions in order to (a) avoid the local maximum problem with EM, and (b) determine the most appropriate number of motions (using MDL). Once the edge motions have been determined, the

---

<sup>2</sup>Procrustes was a bandit in Greek legend who claimed that his bed would fit all guests, which he achieved by either stretching the victim or cutting off their legs. The term *procrustean* refers to a scheme which ruthlessly forces something to fit a pattern.

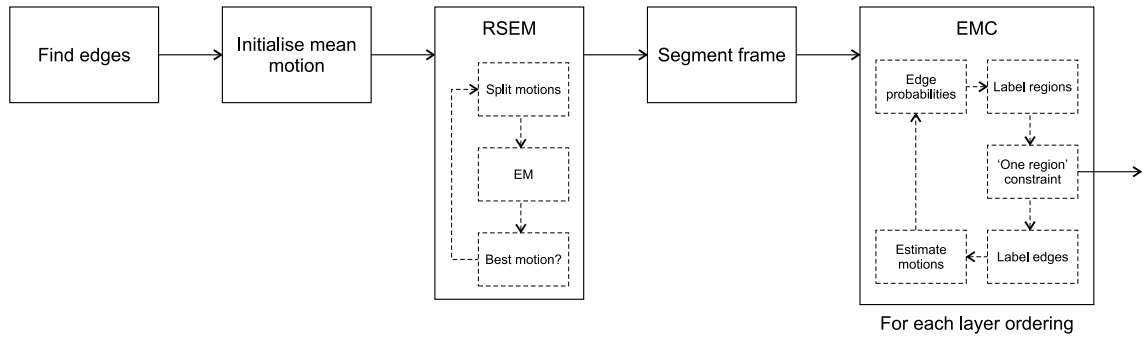


Figure 7.8: *Implementation for multiple motions.* Overview of the algorithm with all described additions.

static segmentation is carried out to find the image regions, and the system enters the EMC loop (Section 7.4), which performs both the region labelling and the global optimisation.

The EMC loop is performed once for each possible layer ordering. The regions are labelled, constrained to have only one object on each foreground layer (Section 7.5), and the implied edge labels used to refine the motions, and thus the edge probabilities. Having run this loop to convergence for each layer ordering, the maximum over all layer orderings is selected as the most likely solution and the final segmentation.

## 7.7 Evaluation

### 7.7.1 Overview

This section presents results for two three-motion sequences, shown in Figures 7.9 and 7.11. The segmentation performance is evaluated, and the discrimination of the MDL approach assessed by considering both three- and two-motion sequences. Further development and evaluation of multiple-motion edge-based segmentations are an avenue for future work.

### 7.7.2 Model selection

The RSEM algorithm attempts several different initialisations with different numbers of motions in order to find the ‘best’ solution according to the Minimum Description Length (MDL) principle [116, and Section 7.2.2]. Table 7.2 shows the coding cost for four sequences when different numbers of models are fitted. Two

	$n_m$	Motion	Edge	Residual	Total		$n_m$	Motion	Edge	Residual	Total
Foreman	1	60	0	5067	5127	Library	1	60	0	2341	2401
	2	120	482	3733	<b>4335</b>		2	120	133	1691	1944
	3	180	764	3467	4411		3	180	211	1450	<b>1841</b>
	4	240	964	3334	4538		4	240	266	1400	1906
Car	1	60	0	10491	10551	Car&Van	1	60	0	4158	4218
	2	120	518	5167	<b>5805</b>		2	120	322	3669	4131
	3	180	821	4931	5932		3	180	510	3109	<b>3799</b>
	4	240	1036	4763	6039		4	240	644	2944	3828

Table 7.2: *Selecting the best number of motions: Minimum Description Lengths.* For different numbers of motions ( $n_m$ ), the total cost is that of encoding the motion parameters ('Motion'), edge labelling ('Edge') and the residual error ('Residual'). The two sequences on the left are expected have two motions, the two on the right have three. The minimum costs (in bold) agree with the desired outcome.

sequences are expected to be best fitted by two models, and two by three models. These sequences are all correctly identified.

The minimum cost for the **Foreman** sequence (Figure 5.1) occurs when two motion models are used, although there is also some small support for fitting the girders at the bottom right corner as a third motion (they are at a greater depth than the concrete structure behind him). The **Car** sequence (Figure 5.7) is also correctly identified as a two-motion sequence.

The two three-motion sequences, **Library** and **Car&Van** each have a minimum cost under three motions, which is the desired outcome. In the **Car&Van** case, four motions is nearly as favourable. In this sequence, EM suffers greatly from local maxima, and any edge labelling is difficult, which makes the figures for this case noisier than they might otherwise be.

### 7.7.3 Library sequence

Figure 7.9 shows the two frames considered from the **Library** sequence. Here the scene is static but the camera moves from right to left. The books, statue and background are at different depths, and so have different image motions.

The RSEM algorithm works well to label the edges. As was shown in Figure 7.3(a), the best two-motion solution fits one motion to the books, and another to the other edges (the statue and the background). When this 'background' motion is split, it finds two motions—the statue and the background—and it is this solution which yields the smallest MDL score.

Labelling the motion of the horizontal lines in the scene is difficult as the camera (and hence the object) motion is horizontal, and so these edges could fit any of





Figure 7.9: *Library sequence*. Frames 36–37 from the Library sequence. The camera moves from right to left here, and the image motion is due to parallax, with the books, statue and background at different depths.

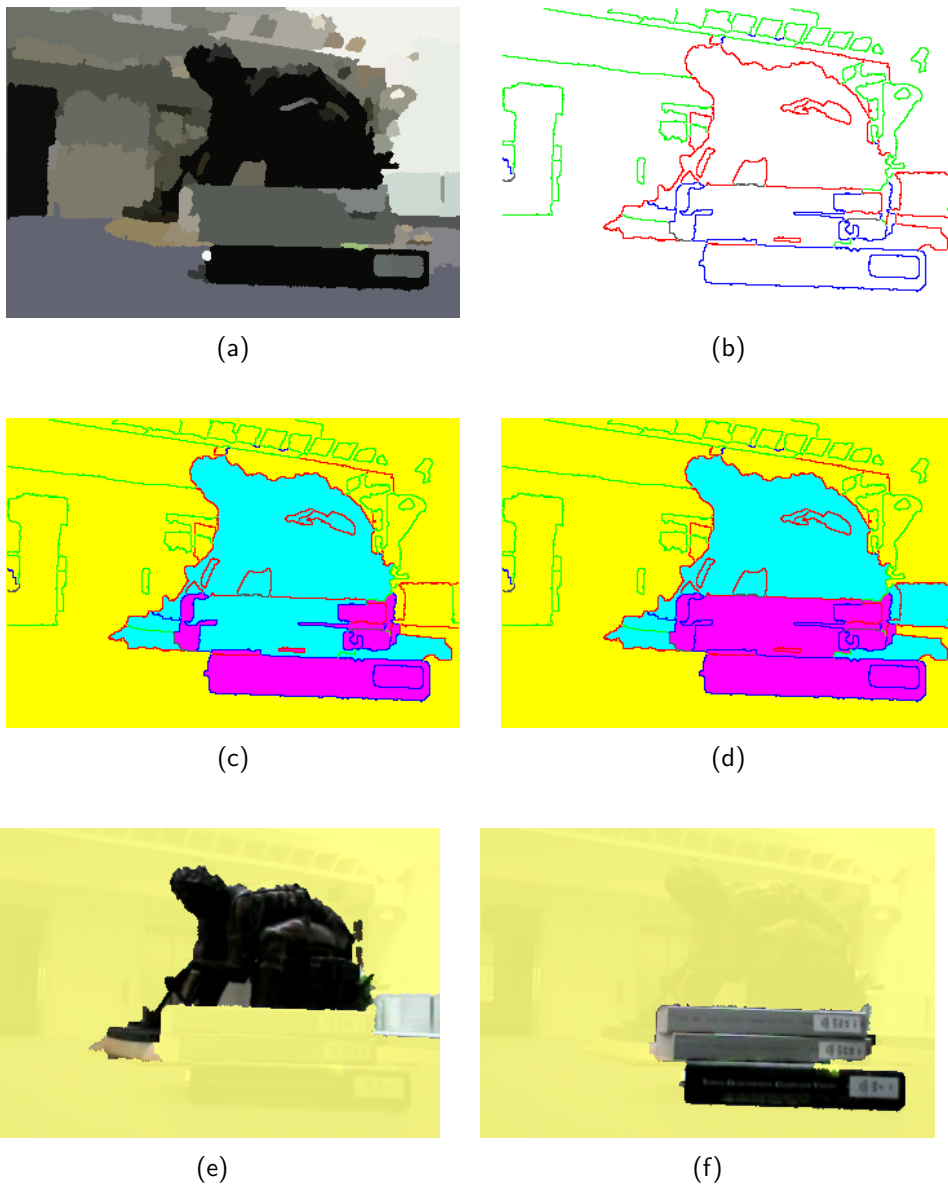


Figure 7.10: *Library segmentation from two frames*. (a) Region segmentation; (b) Region edges labelled according to their motion; (c) Region labelling (using original edge labels); (d) Region labelling after EMC; (e) Middle layer of final segmentation; (f) Front layer of final segmentation.

the motions equally well. Figure 7.10(b) shows the edge probabilities after RSEM and it can be seen that the edge marking the top of the books has been incorrectly labelled.<sup>3</sup> As a result, when the region labelling is performed (Figure 7.10(c)), some of the book regions are incorrectly assigned to the statue.<sup>4</sup>

The Expectation-Maximisation-Constrain (EMC) loop is then entered, which performs the global constrained optimisation. The edge labels of Figure 7.10(b) can be seen to have a number of mislabelled edges, and these are logically inconsistent. Figure 7.10(d) shows the region labelling (and edge probabilities) after the EMC loop, which shows an improvement. The constrained optimisation has now labelled the top edge of the books correctly, and the region labelling is now very good.

The EMC loop is performed for each possible layer ordering (six in this case)—the results here only show the final most likely ordering. Of the different layer orderings, the ones which label the green motion as the background layer (i.e. correctly) are much more likely. However, the ordering of the two foreground layers is more ambiguous in this case. The main horizontal edge dividing the two objects is not labelled with any great confidence, even after EMC, and there are very few other edges which contribute to the layer ordering decision. The correct ordering is selected, but only with a probability of 53% over the other foreground ordering.

The segmentation of this sequence takes about three minutes on a 300MHz PC. The majority of time is spent in the EMC loop, which has to be repeated six times to consider all possible layer orderings, and has to perform a complete region segmentation at each iteration of the loop.

#### 7.7.4 Car & Van sequence

The second sequence presented here is the *Car&Van* sequence, shown in Figure 7.11. Recorded with a hand-held MPEG-1 camera, this shows an essentially static background while the white car closest to the camera pulls out (to the left) as the yellow van speeds by. The size of the van's motion means that under two motions the van's edges are mainly outliers and it is here that the value of considering a third motion initialised from the outliers becomes apparent. The MDL process (Table 7.2) correctly selects three motions after the RSEM stage.

When the edges are labelled (Figure 7.12(b)), the car motion (green) also fits parts of the building well, particularly due to the repeating nature of the classical

<sup>3</sup>In the three-motion sequences, edges are displayed according to their most likely motion, with motions 1, 2 and 3 being red, green and blue respectively.

<sup>4</sup>The region labelling indicates the different depths in the following order (closest to the camera first): magenta, cyan, and then yellow for the background.



Figure 7.11: *Car&Van sequence*. Frames 73–74 from the Car&Van sequence. The yellow van passes by on the road as the white car pulls out of the side street.

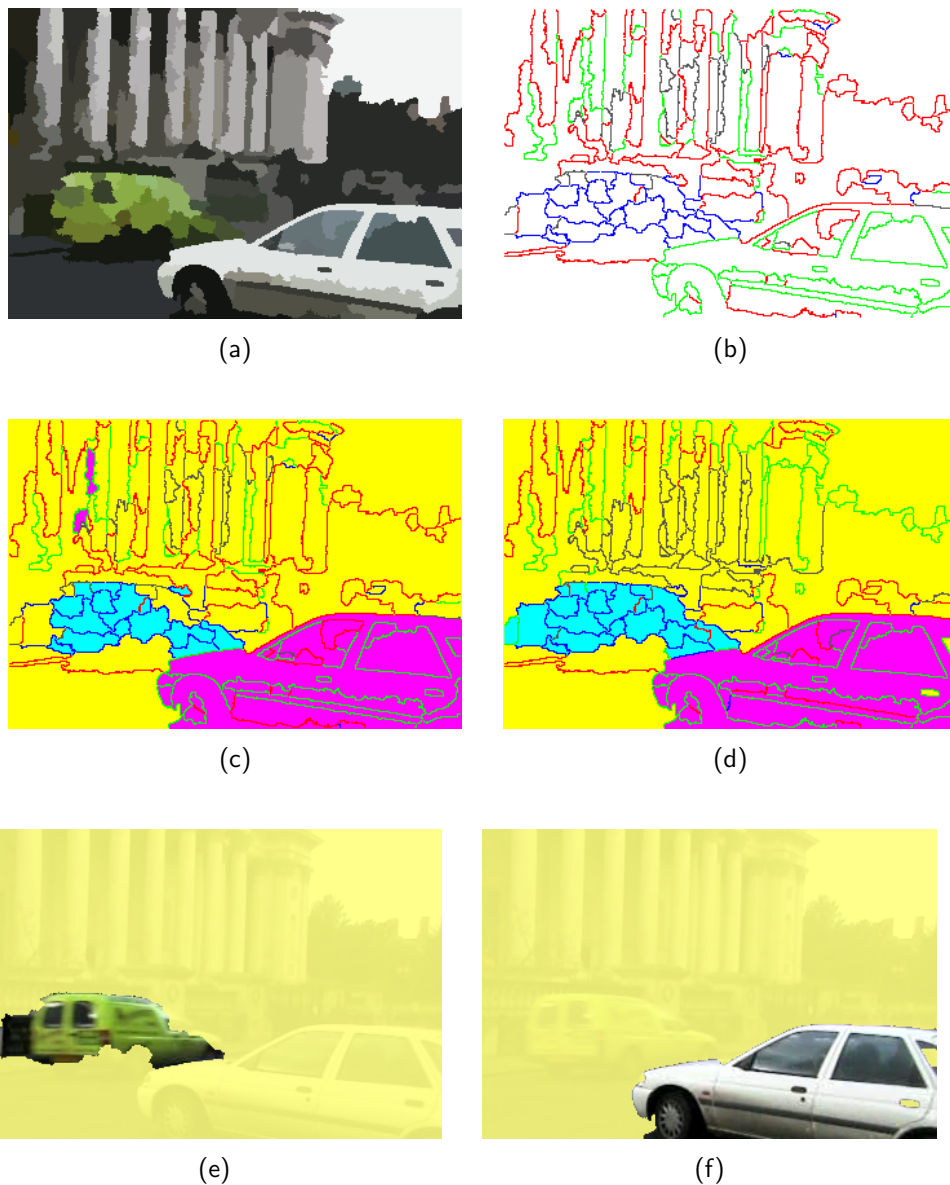


Figure 7.12: *Car&Van segmentation from two frames*. (a) Region segmentation; (b) Region edges labelled according to their motion; (c) Region labelling (using original edge labels); (d) Region labelling after EMC; (e) Middle layer of final segmentation; (f) Front layer of final segmentation.

architecture in the background. This presents some problems to the region labelling stage, as can be seen in Figure 7.12(c), where there are a few regions on the columns which are labelled with the car. It is in cases such as this that the ‘one region’ constraint is needed, in conjunction with EMC, to produce the clean results seen in Figure 7.12(d). This shows the final most likely region labelling and edge probabilities after EMC and again the labels can be seen to be much improved. The entire segmentation again takes a few minutes.

Determining the correct layer ordering is a little easier in this case than in the *Library* sequence and the depth ordering of the car, the van and then the background is significantly more likely. These two sequences have been carefully selected to have some interaction between all three layers, but although there is some, the task is still difficult, as there are very few T-junctions on which to base an opinion. In a more general three-motion sequence, with no object interaction, the relative ordering of the foreground objects would be completely ambiguous.

## 7.8 Discussion

This chapter has discussed the difficulties of fitting more than two motions using an edge-based approach. A robust initialisation for EM has been developed, which uses a sampling approach to avoid the problems of local maxima. This RSEM algorithm also selects the best number of motions according to the MDL principle. A global optimisation scheme has been developed which constrains the edge labels to obey the logical constraints laid down by a region labelling. A technique for constraining the segmentation to simply-connected foreground objects has also been described.

As these necessary extensions show, the segmentation of multiple foreground motions is a much more difficult proposition than a two-motion foreground/background segmentation. Various elements of the scheme increase combinatorially with the number of motions, and the labelling uncertainty introduced by having more motions places a heavier burden on priors and constraints.

The two sequences presented in the evaluation demonstrate the difficulty of labelling the edges in a three-motion case. The RSEM algorithm gives a reasonable edge labelling and motion estimates, and the number of motions is correctly determined. However, the process is rather more fragile: in the *Car&Van* case the edge labelling is barely satisfactory for a good segmentation. Also, the added complexity of the various additional stages increases the computation time by an order of magnitude, taking minutes rather than seconds.

This chapter has demonstrated that the segmentation of frames into multiple independently-moving objects under the edge-based framework is possible. The framework of Chapter 3 is general, and a frame with correctly labelled edges of any number of motions can be completely segmented, up to unresolvable ambiguities. Even in cases where the edge probabilities suffer from some noise, the techniques developed in this chapter, with the MRF-style prior, allow a good segmentation. However, the challenge with multiple-motion segmentation under this framework is one of obtaining a good edge labelling. In Chapter 6, it was shown that the use of multiple frames can greatly improve noisy and ambiguous edge labellings, and this will no doubt be a partial solution to the problem.

Regardless of the use of cumulative edge statistics, the motion and labelling in each individual frame must be determined. The RSEM algorithm presented in this chapter is a useful tool in negotiating a reasonable edge labelling from one pair of frames. Ameliorating the local maxima problem in EM, or developing alternative methods for determining the motion labelling of edges when there are many motions present, are obvious avenues for future work.



---

# Conclusion

---

## 8.1 Summary

This dissertation has considered the problem of segmenting the frames of an image sequence into semantically meaningful areas, using image and motion information. The thesis presented is that image edges are fundamental to obtaining an accurate motion segmentation, since they both provide the boundaries of objects in the image, and can be efficiently tracked using robust statistical techniques.

Chapter 3 developed the theory linking the motion labelling of edges with that of the image regions they bound. It was shown that not only is an edge labelling sufficient to label regions, but also that occlusion constraints make logical reasoning over edges and regions necessary for a complete segmentation. This provides both the complete dense labelling, and the relative depth ordering of the different segmented areas. A Bayesian framework was outlined which formalised this approach, making it possible to perform a motion segmentation using edges in the presence of the noise and uncertainty which are unavoidable features of real sequences.

An implementation of this framework was presented in Chapter 4, and extended in Chapters 6 and 7. Two separate optimisation stages are required to determine the edge labelling and then the region labelling. The first is implemented using the Expectation-Maximisation (EM) algorithm, and the second by simulated annealing. The extension to multiple frames (Chapter 6) showed that accumulating edge motion information across frames resolves ambiguities and provides a more robust labelling. Extending the implementation to multiple motions in Chapter 7

demonstrated the generality of the framework. This chapter also introduced a new initialisation stage for EM which also selects the best number of motions (using the Minimum Description Length principle), and a global initialisation and constraint stage. Extensive results for the basic two-motion, two-frame case were discussed in Chapter 5, featuring real video sequences covering many different genres. Evaluations of the multiple-frame and multiple-motion cases were also presented in the relevant chapters. These showed the success of this framework, particularly in the two-motion case, giving clean, accurate segmentations.

## 8.2 Discussion

The edge-based framework introduced in this dissertation is a new, general scheme for motion segmentation. The analysis of image edges should form an integral part to *any* motion segmentation scheme concerned with the accurate extraction of motion boundaries. The framework presented here integrates segmentation using edges with edge-based motion estimation, creating an efficient and elegant unified approach.

The system developed for this dissertation demonstrates that a fast implementation of this framework is possible for the segmentation of two motions. Despite the necessity for two separate iterative optimisation stages, this (relatively unoptimised) implementation can segment a frame in a few seconds on standard hardware. With the continual increase in affordable computing power, and with an optimised implementation, a real-time implementation of this framework for two motions is within reach.

The results for the two-frame implementation demonstrate that it works very well when there are two clear motions (i.e. the background and one large foreground object). The main limitation of the system is the EM process used for edge labelling. When there are only a few edges representing an object, this can fail to converge to the global maximum, and the edge labelling can be poor. When the edge labels are good, the thesis that these are sufficient for a complete labelling (up to unresolvable ambiguities) is verified. The region-based approach to a dense labelling gives accurate boundaries and clean segmentations, but is dependent on the initial static segmentation. This is usually very good, but problems occur when the occluding boundary is weak, or when the regions are required to represent fine image detail.

Using multiple frames significantly increases the robustness and accuracy of the edge labels, with the result that the segmentation is much improved. Many sequences are segmented very accurately using this multi-frame approach, and complete sequences can be segmented. The speed and accuracy of the edge-based seg-



mentation scheme will allow many different video analysis applications to be built upon this framework.

Segmenting multiple motions is much more difficult than segmenting two motions. In these sequences, each object often has fewer edges, and this again causes problems when determining the edge labelling. The proposed extensions have met with some success, but segmenting multiple motions pushes the limit of the current algorithms. Dealing with these cases in a more satisfactory manner is an area for future research.

## 8.3 Suggestions for further work

This dissertation has demonstrated the effectiveness of an edge-based approach to motion segmentation, and has presented a successful implementation for two motions. However, there are a number of interesting ways that this work could be extended, or improved.

**Application development** Motion segmentation is an enabling technology, and the first stage in many video analysis technologies. The existing edge-based implementation gives excellent results for most two-motion sequences, and applications can be built upon this. For example, video description tools could be developed, which consider a mosaic of the background (see Section 6.7), and the manner in which the foreground object moves over this background and its shape changes. The segmentation scheme should also be tested as part of an MPEG-4 coding scheme [82, 129].

**Improved edge statistics** The labelling of edges, which is integral to this framework, would be improved by a better statistical model of the sample point errors. For example, a Markov chain approach [61] (as considered in Appendix C) could be pursued. This is particularly important when segmenting a larger number of motions.

**Alternative motion parameterisations** The edge tracking scheme used in this dissertation is not restricted to 2D projective motions, and the use of alternative models could be investigated. Other deformation modes could, for example, be determined by a Principal Component Analysis of the sample points, as in Cootes and Taylor's Active Shape Models [38, 39]. The Lie algebra tracking approach can easily be extended to include any parametric deformations, and has been extended to track articulated objects [45], which should also be considered.

**Motion-assisted static segmentation** Region labelling errors are usually due to errors in the static segmentation, in particular a missing occluding boundary. If only part of the boundary is missing, there are usually strong foreground edges which are surrounded by background. Because of this, the error can be detected. In these cases the occluding edge could be introduced either by a model-based edge completion scheme, or by searching for a weak edge which is present at that location in both frames. Using this motion information to assist the static segmentation creates a truly integrated *motion* segmentation of a frame or sequence.

**Multi-frame segmentation** The multiple frame aspects of the implementation could benefit from a more sophisticated treatment. The sample point motion hypotheses should be correctly treated—if a sample point is only propagated because it finds a match under motion 1, it should from then on only be allowed to search for matches under motion 1. A full multi-frame implementation should also, if possible, accumulate information from all of the sequence for all of the frames and ensure consistency of edge labels across the sequence.

**Multiple motions** An extended investigation into multiple motions should be conducted. In particular, alternative approaches to the motion estimation and edge labelling should be considered. This may involve multi-resolution approaches [9, 78], or motion estimates from other (e.g. pixel-based) sources.

## 8.4 A final word: Edges vs pixels

The evaluation of Section 5.9 showed that in some types of sequence, the widely-used pixel-based approaches have the upper hand, and in others the feature-based approach advocated in this work is more appropriate. Pixel-based schemes perform well in textured images, but less well otherwise. This dissertation's edge- and region-based approach is distracted by excessive texture, but generates excellent, fast segmentations in many other cases, including sequences with low texture between edges. A truly general and successful motion segmentation scheme will have to combine elements of both approaches. Image edges *must* be used as they provide the only guaranteed means of obtaining an accurate object boundary, and they also allow analysis of the relative depths. However, if textured surfaces are available, it is foolish to reject this valuable source of motion information.

Although this dissertation has presented pixels and edges as two alternatives, the approaches are not mutually exclusive. They could be combined in several ways.

A pixel-based approach could be used to estimate the image motions, and then the edges and regions could be labelled; this may sidestep the problems with the edge-based EM process in textured images. Alternatively, pixels could be brought in at a later stage, as another piece of evidence in the labelling of regions, or could be used to bring out fine detail. The future direction for motion segmentation will lie in combined pixel- and edge-based approaches.



---

# Parameter estimation

---

## A.1 Motion estimation

The Lie group formulation outlined in Chapter 4 reveals that the motion of a sample point  $k$  which obeys a 2D projective motion (or one of its subgroups) can be expressed as a linear combination of the *group generators* at that point,  $\mathbf{L}_j^k$ :

$$\text{Motion at } k = \sum_{j=1}^{n_d} \alpha_j \mathbf{L}_j^k \quad (\text{A.1})$$

where there are  $n_d$  generators. The  $\alpha_j$  are the parameters of the motion, and are the same for all sample points.<sup>1</sup> Measurements are taken along edge normals in order to estimate these motion parameters.

The residual error measured at each sample point,  $r^k$ , is the difference between the measured normal motion  $d^k$  and the motion predicted by the current motion parameters when it is projected onto the unit edge normal  $\hat{\mathbf{n}}^k$ :

$$\text{Residual error} = r^k = d^k - \left( \sum_j \alpha_j \mathbf{L}_j^k \right) \cdot \hat{\mathbf{n}}^k \quad (\text{A.2})$$

Since the summation and dot product are linear, this may be rewritten as

$$= d^k - \sum_j \alpha_j (\mathbf{L}_j^k \cdot \hat{\mathbf{n}}^k) \quad (\text{A.3})$$

---

<sup>1</sup>There are eight parameters in the full 2D projective group  $P(2)$ , and six in the 2D affine  $GA(2)$  (see Tables 4.3 and 4.4).

which, for brevity, may be simplified by writing  $x_j^k = (\mathbf{L}_j^k \cdot \hat{\mathbf{n}}^k)$ , giving:

$$r^k = d^k - \sum_j \alpha_j x_j^k \quad (\text{A.4})$$

which is of the form usually solved using *linear least squares*.

## A.2 Least squares solution

The least squares solution to the linear equation given in (A.4) is given by those parameter values  $\boldsymbol{\alpha}$  which, over the whole set of  $K$  data points, minimise the total squared error. This error is defined as:

$$\text{Error} = \sum_{k=1}^K (r^k)^2 = \sum_{k=1}^K \left( d^k - \sum_{j=1}^{n_d} \alpha_j x_j^k \right)^2 \quad (\text{A.5})$$

The necessary condition for this is that the error with respect to each parameter be a minimum i.e:

$$\frac{\partial(\text{Error})}{\partial \alpha_i} = 0 \quad (\text{A.6})$$

Performing this minimisation explicitly for the the squared error defined in (A.5) gives the following system of equations:

$$\sum_{k=1}^K 2x_i^k \left( d^k - \sum_{j=1}^{n_d} \alpha_j x_j^k \right) = 0 \quad i = 1 \dots n_d \quad (\text{A.7})$$

The factor of two may be ignored, and rearranging (A.7) gives

$$\sum_{k=1}^K d^k x_i^k = \sum_{k=1}^K x_i^k \sum_{j=1}^{n_d} \alpha_j x_j^k \quad (\text{A.8})$$

$$= \sum_{k=1}^K \sum_{j=1}^{n_d} x_i^k x_j^k \alpha_j \quad (\text{A.9})$$

as the equation which must hold for the least squares estimate of each parameter  $\alpha_i$ .

Over all the parameters, these equations can be gathered into matrix-vector form by writing the left-hand side as a vector  $\mathbf{v}$ , and the right-hand side as a matrix  $\mathbf{M}$  which multiplies  $\boldsymbol{\alpha} = (\alpha_1 \dots \alpha_{n_d})$ , leaving

$$\mathbf{v} = \mathbf{M}\boldsymbol{\alpha} \quad (\text{A.10})$$

where the elements of  $\mathbf{v}$  and  $\mathbf{M}$  are given by:

$$v_i = \sum_{k=1}^K d^k x_i^k \quad (\text{A.11})$$

$$M_{ij} = \sum_{k=1}^K x_i^k x_j^k \quad (\text{A.12})$$

The parameter vector  $\boldsymbol{\alpha}$  may be easily found by inverting the square matrix  $\mathbf{M}$ :

$$\boldsymbol{\alpha} = \mathbf{M}^{-1} \mathbf{v} \quad (\text{A.13})$$

This is usually best solved by Singular Value Decomposition (SVD) and back-substitution [112].

## A.3 M-estimators

In any parameter estimation problem, the problem of erroneous data points, outliers, must be considered. Parameter estimation techniques typically consider the residual error  $r^k$  between the  $k$ th observation and the fitted value (for example (A.5)). The standard least squares method, minimising  $\sum_k (r_k)^2$ , is particularly unstable to outliers since points with a large residual will have a disproportionate influence on the result. Additionally, the least squares solution is also only the *maximum likelihood* estimate of the parameters if the errors are independent and their distribution is Gaussian, which is often not the case.<sup>2</sup>

M-estimators [69, 70, 115] are a popular method of robust fitting, since they still provide a maximum-likelihood style of estimation. They are a generalisation of the least squares solution, replacing the error based on squared residuals (A.5) by that using a general function,  $\rho()$ :

$$\arg \min_{\boldsymbol{\alpha}} \sum_{k=1}^K \rho(r^k) \quad (\text{A.14})$$

This is the maximum likelihood estimator for this data if the probability distribution of residuals,  $P(r^k)$ , is given by

$$P(r^k) \propto e^{-\rho(r^k)} \quad (\text{A.15})$$

---

<sup>2</sup>The sample point residuals considered in this dissertation are clearly not Gaussian, and are closer to a Laplacian distribution (see Section 4.4.4, particularly Figure 4.6).

since in this case, for independent data, the *likelihood* is defined to be the product over the samples,  $\prod_{k=1}^K P(r^k)$ , and so the maximum likelihood estimation is then given by the parameters which maximise this:

$$\begin{aligned} \arg \max_{\alpha} \prod_{k=1}^K P(r^k) &= \arg \max_{\alpha} \sum_{k=1}^K \ln P(r^k) \\ &= \arg \min_{\alpha} \sum_{k=1}^K \rho(r^k) \end{aligned} \quad (\text{A.16})$$

which is the same as (A.14).

As with the least squares case considered in the previous section, this is minimised when the partial derivative with respect to each parameter is zero:

$$\frac{\partial}{\partial \alpha_i} \sum_{k=1}^K \rho(r^k) = \sum_{k=1}^K \frac{d\rho(r^k)}{dr^k} \frac{\partial r^k}{\partial \alpha_i} \quad (\text{A.17})$$

$$= \sum_{k=1}^K \psi(r^k) \frac{\partial r^k}{\partial \alpha_i} = 0 \quad i = 1 \dots n_d \quad (\text{A.18})$$

where  $\psi$  is the derivative of the weight function,

$$\psi(x) = \frac{d\rho(x)}{dx} \quad (\text{A.19})$$

which is called the *influence function*. This is a measure of the influence of a data point on the value of the parameter estimate. With the least squares estimate,  $\rho(x) = x^2$ , the influence function is  $\psi(x) = 2x$  i.e. the influence of a data point increases linearly with the size of the error, which is clearly non-robust. For a robust estimator, this influence should be bounded (see Table A.1).

Defining the *weight function*  $w(x)$  as

$$w(x) = \frac{\psi(x)}{x} \quad (\text{A.20})$$

equation (A.18) becomes:

$$\sum_{k=1}^K w(r^k) r^k \frac{\partial r^k}{\partial \alpha_i} = 0 \quad i = 1 \dots n_d \quad (\text{A.21})$$

This last manipulation allows M-estimators to be implemented within a least squares framework, since (A.21) leads to exactly the same series of equations as are required



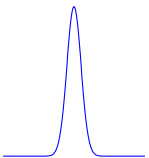
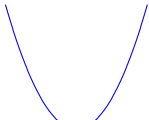
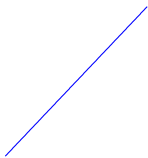
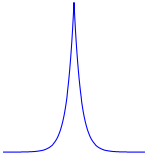
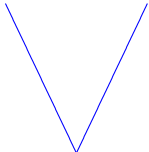
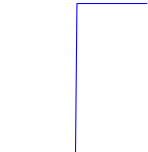
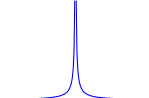
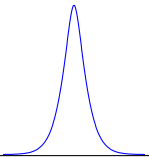
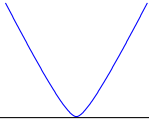
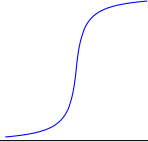
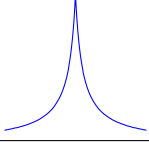
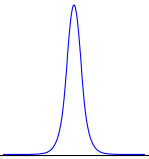
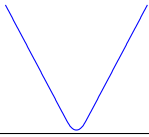
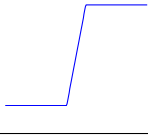
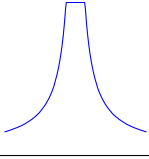
Type	$P(x)$	$\rho(x)$	$\psi(x)$	$w(x)$
$L_2$	$e^{-\frac{x^2}{2}}$ 	$\frac{x^2}{2}$ 	$x$ 	1 
$L_1$	$e^{- x }$ 	$ x $ 	$\text{sign}(x)$ 	$\frac{1}{ x }$ 
'Fair'	$e^{-c x } \left(1 + \frac{ x }{c}\right)^{c^2}$ 	$c^2 \left[ \frac{ x }{c} - \ln \left(1 + \frac{ x }{c}\right) \right]$ 	$\frac{x}{1 + \frac{ x }{c}}$ 	$\frac{1}{1 + \frac{ x }{c}}$ 
Huber	$\begin{cases} e^{-\frac{x^2}{2}} & \text{if }  x  \leq k \\ e^{-k( x  - \frac{k}{2})} & \text{if }  x  \geq k \end{cases}$ 	$\begin{cases} \frac{x^2}{2} & \text{if }  x  \leq k \\ k( x  - \frac{k}{2}) & \text{if }  x  \geq k \end{cases}$ 	$\begin{cases} x & \text{if }  x  \leq k \\ k \text{sign}(x) & \text{if }  x  \geq k \end{cases}$ 	$\begin{cases} 1 & \text{if }  x  \leq k \\ \frac{k}{ x } & \text{if }  x  \geq k \end{cases}$ 

Table A.1: *M-estimators*. Influence functions for least squares ( $L_2$ ), least absolute value ( $L_1$ ), 'Fair' and Huber M-estimators.

to solve the weighted least squares problem:

$$\arg \min_{\alpha} \sum_{k=1}^K w(r^k) r^{k^2} \quad (\text{A.22})$$

This is implemented as an iterative process, where the weight term  $w(r^k)$  uses the residual value calculated using the current parameters. Each residual is multiplied by square root of its weight and then the new parameter values may be found by standard least squares.

In terms of the matrix-vector solution given earlier, (A.12) and (A.11) become:

$$v_i = \sum_{k=1}^K d^k x_i^k w(r^k) \quad (\text{A.23})$$

$$M_{ij} = \sum_{k=1}^K x_i^k x_j^k w^2(r^k) \quad (\text{A.24})$$

where the residual error  $r^k$  is that calculated using the previous parameter values, according to (A.4).

Table A.1 shows a number of possible influence functions, weights, and the probability distributions for which they are the maximum likelihood estimator.<sup>3</sup> Many more M-estimators are suggested in the literature [70, 115, 162] and, in cases where the probability distribution is known, there is no excuse for not using an appropriate M-estimator rather than Least Squares, even to the extent of deriving the correct weight function.

## A.4 Regularisation

In some parameter estimation cases, the problem is ill-posed, hence the solution is very sensitive to noise. This occurs when the available data does not adequately constrain all of the degrees of freedom in the model.<sup>4</sup> Regularisation is a technique for including prior information in the least squares process, so that in these cases the solution can be guided to the one which, *a priori*, is more likely.

<sup>3</sup>For the sample point errors in this dissertation, experiments (Section 4.4.4) indicate that the probability distribution is exponential, which is reasonably well modelled by the latter three functions in Table A.1. Of these, the ‘Fair’ function is used for this system as it has fewer discontinuities, and is found to yield good convergence [115]. A value of  $c = 1$  is used.

<sup>4</sup>The Car sequence discussed in Chapter 5 is an example of this. The two motions in the scene are horizontal translations of different magnitudes, but an alternative solution would include shear terms. Regularisation is needed to penalise this solution and encourage the more likely translation solution.

This is performed by modifying the cost function to include a penalty term which is a function of the solution vector. The usual approach, Tikhonov regularisation [142], minimises

$$\arg \min_{\boldsymbol{\alpha}} \left[ r^k{}^2 + \lambda^2 \|\mathbf{L}\boldsymbol{\alpha}\|_2^2 \right] \quad (\text{A.25})$$

for some penalty matrix  $\mathbf{L}$ . This expression can, of course, also be used for M-estimators by changing the first term. In the solution, the penalty simply augments the matrix  $\mathbf{M}$ , giving:

$$\boldsymbol{\alpha} = (\mathbf{M} + \lambda \mathbf{L}^T \mathbf{L})^{-1} \mathbf{v} \quad (\text{A.26})$$

Often  $\mathbf{L}$  is chosen to be the identity matrix,  $\mathbf{L} = \mathbf{I}$ , thus putting a penalty on the total energy of the solution vector. In this case, the parameter  $\lambda$  is added to each  $M_{ii}$ , the diagonal elements of  $\mathbf{M}$ . The  $\lambda$  parameter controls the weight of the regularisation term on the cost function. This should be large enough to have the desired effect of guiding the solution, but should not outweigh the data term.<sup>5</sup>

## A.5 Normalisation

The least squares solution (A.13) calls for the inversion of a matrix  $\mathbf{M}$ , which is formed from the modes of the parametric model (the  $x_i$  in (A.12)). If these modes have vastly different magnitudes, this matrix can be ill-conditioned and accuracy of the inverse can be very poor.<sup>6</sup> One means to achieve good conditioning is to redefine the modes such that their magnitudes are all in the same range, but a more general solution is to *normalise* the matrices used to calculate the solution.

The equation which must be solved to find the solution  $\boldsymbol{\alpha}$ , (A.10), is repeated here:

$$\mathbf{M}\boldsymbol{\alpha} = \mathbf{v} \quad (\text{A.27})$$

This may be premultiplied and augmented by an invertible matrix  $\mathbf{S}$ , to give the equivalent expression:

$$\mathbf{S}\mathbf{M}(\mathbf{S}\mathbf{S}^{-1})\boldsymbol{\alpha} = \mathbf{S}\mathbf{v} \quad (\text{A.28})$$

---

<sup>5</sup>The system in this dissertation encourages translational motions by penalising all motion parameters apart from the first two. This is done by incrementing the  $M_{ii}$  for  $i = 2 \dots n_d$  by  $\lambda$ , where  $\lambda$  is the mean of all the diagonal elements of  $\mathbf{M}$ . This give the prior and the measurements approximately equal weight.

<sup>6</sup>The modes in this dissertation's work are the projections of the various vector fields  $\mathbf{L}_j$ . The magnitudes of the different  $\mathbf{L}_j$  vary greatly:  $|\mathbf{L}_{1,2}| \approx 1$ ,  $|\mathbf{L}_{3-6}| \approx (\text{size of image})$ ,  $|\mathbf{L}_{3-6}| \approx (\text{size of image})^2$ . See Table 4.3 for details of the different fields. This matrix is therefore frequently highly ill-conditioned.

<ul style="list-style-type: none"> <li>• <b>Calculate normalisation factors</b> For <math>i = 1 \dots n</math> <math display="block">S_i = \frac{1}{\sqrt{M_{ii}}}</math> </li> <li>• <b>Pre-normalise</b> For <math>i, j = 1 \dots n</math> <math display="block">v'_i = v_i S_i</math> <math display="block">M'_{ij} = M_{ij} S_i S_j</math> </li> <li>• <b>Compute <math>\alpha' = M'^{-1} \mathbf{v}'</math> using SVD</b></li> <li>• <b>Post-normalise</b> For <math>i = 1 \dots n</math> <math display="block">\alpha_i = \alpha'_i S_i</math> </li> </ul>
--

Table A.2: *Matrix normalisation.* Conditioned solution  $\mathbf{M}\alpha = \mathbf{v}$  to calculate the vector  $\alpha$ .

This can be expressed as

$$\mathbf{M}'\alpha' = \mathbf{v}' \quad (\text{A.29})$$

by defining:

$$\mathbf{M}' = \mathbf{S}\mathbf{M}\mathbf{S} \quad \alpha' = \mathbf{S}^{-1}\alpha \quad \mathbf{v}' = \mathbf{S}\mathbf{v} \quad (\text{A.30})$$

If  $\mathbf{S}$  is a diagonal matrix defined as

$$\mathbf{S} = \begin{cases} S_{ij} = \frac{1}{\sqrt{M_{ii}}} & i = j \\ S_{ij} = 0 & i \neq j \end{cases} \quad (\text{A.31})$$

then the matrix  $\mathbf{M}'$  will have ones along the leading diagonal and will thus be much better conditioned.

This normalisation scheme may be used wherever an equation of the form  $\mathbf{M}\alpha = \mathbf{v}$  needs to be solved, as outlined in Table A.2.

---

# Maximum likelihood estimation via EM

---

## B.1 The EM algorithm

The Expectation-Maximisation (EM) algorithm [43] is the standard approach for finding model parameters when some of the data is missing. That is, of the complete data  $Z = \{XY\}$ , when only the data  $X$  are known. As outlined in Section 4.4.2, for the desired parameters  $\Theta$ , given some initial guess  $\Theta^g$ , a function  $Q$  can be defined:

$$Q(\Theta, \Theta^g) = \mathcal{E} [\log \mathcal{L} [\Theta; X, Y] | X, \Theta^g] \quad (\text{B.1})$$

$$= \sum_{\mathbf{y}} \log \mathcal{L} [\Theta; X, \mathbf{y}] P(\mathbf{y} | X, \Theta^g) \quad (\text{B.2})$$

This considers the expected value of the complete-data log-likelihood  $\mathcal{L} [\Theta; X, Y]$  with respect to the unknown data  $Y$ , given the observed data  $X$  and the initial parameter estimates  $\Theta^g$ . The current set of parameters are used to evaluate the expression and (B.2) is maximised over the updated set of parameters  $\Theta$ . This Appendix considers how this maximisation can be achieved in the case of mixture models.

## B.2 Estimation of mixture model parameters

The estimation of parameters for mixture models is the most common application of the EM algorithm. A mixture model is a probability distribution constructed from a weighted sum of distributions, for example constructing a multi-modal distribution from the sum of Gaussian distributions. The general model is:

$$P(\mathbf{x}|\Theta) = \sum_{m=1}^M c_m P(\mathbf{x}|\theta_m) \quad (\text{B.3})$$

where the parameters are  $\Theta = \{c_1, \dots, c_M, \theta_1, \dots, \theta_M\}$  such that  $\sum_{m=1}^M c_m = 1$ .

The *maximum likelihood* estimate of these parameters is given by maximising the likelihood, or equivalently the log-likelihood, of the data. Using only the observed data  $X$ , the log-likelihood for this distribution is (assuming independent data)

$$\log \mathcal{L}[\Theta; X] = \log \prod_{i=1}^N P(x_i|\Theta) = \sum_{i=1}^N \log \left( \sum_{m=1}^M c_m P(x_i|\theta_m) \right) \quad (\text{B.4})$$

which is difficult to optimise because it contains the log of a sum. The problem can be made tractable by positing the existence of some unobserved data  $Y$ , a label for each data point. Each data point is considered to be generated by one of the component densities, and by identifying this component the sum is removed. The likelihood then becomes:

$$\log \mathcal{L}[\Theta; XY] = \sum_{i=1}^N \log(c_{y_i} P(x_i|\theta_{y_i})) \quad (\text{B.5})$$

where the  $y_i$  now select the particular component. Since the problem now involves this hidden variable, the EM algorithm is a natural choice for performing the optimisation.

The EM algorithm consists of two stages. The E-stage calculates the label probabilities,  $P(y_i|x_i\Theta^g)$ , and from these the second term in (B.2) can be calculated (again assuming independence):

$$P(\mathbf{y}|X\Theta^g) = \prod_{i=1}^N P(y_i|x_i, \Theta^g) \quad (\text{B.6})$$

The M-stage then uses these label probabilities in maximising the expected likelihood of the data (B.2). This appendix considers this second maximisation stage,

assuming that the E-stage has already been completed.<sup>1</sup>

Having determined (B.5) and (B.6), these may be substituted into (B.2) to give

$$\begin{aligned}
Q(\Theta, \Theta^g) &= \sum_{\mathbf{y}} \log(\mathcal{L}[\Theta; X, \mathbf{y}]) P(\mathbf{y}|X, \Theta^g) \\
&= \sum_{\mathbf{y}} \sum_{i=1}^N \log(c_{y_i} P(x_i|\theta_{y_i})) \prod_{j=1}^N P(y_j|x_j, \Theta^g) \\
&= \sum_{y_1=1}^M \sum_{y_2=1}^M \cdots \sum_{y_N=1}^M \sum_{i=1}^N \log(c_{y_i} P(x_i|\theta_{y_i})) \prod_{j=1}^N P(y_j|x_j, \Theta^g) \\
&= \sum_{y_1=1}^M \sum_{y_2=1}^M \cdots \sum_{y_N=1}^M \sum_{i=1}^N \sum_{\ell=1}^M \delta_{\ell, y_i} \log(c_{\ell} P(x_i|\theta_{\ell})) \prod_{j=1}^N P(y_j|x_j, \Theta^g) \\
&= \sum_{i=1}^N \sum_{\ell=1}^M \log(c_{\ell} P(x_i|\theta_{\ell})) \sum_{y_1=1}^M \sum_{y_2=1}^M \cdots \sum_{y_N=1}^M \delta_{\ell, y_i} \prod_{j=1}^N P(y_j|x_j, \Theta^g) \quad (\text{B.7})
\end{aligned}$$

The second half of this expression can then be greatly simplified by considering that for each  $\ell \in 1 \dots M$ :

$$\begin{aligned}
&\sum_{y_1=1}^M \sum_{y_2=1}^M \cdots \sum_{y_N=1}^M \delta_{\ell, y_i} \prod_{j=1}^N P(y_j|x_j, \Theta^g) \\
&= \left( \sum_{y_1=1}^M \cdots \sum_{y_{i-1}=1}^M \sum_{y_{i+1}=1}^M \cdots \sum_{y_N=1}^M \prod_{y_j=1, j \neq i}^N P(y_j|x_j, \Theta^g) \right) P(\ell|x_i, \Theta^g) \quad (\text{B.8})
\end{aligned}$$

$$= \prod_{y_j=1, j \neq i}^N \left( \sum_{y_j=1}^M P(y_j|x_j, \Theta^g) \right) P(\ell|x_i, \Theta^g) = P(\ell|x_i, \Theta^g) \quad (\text{B.9})$$

since  $\sum_{y_j=1}^M P(y_j|x_j, \Theta^g) = 1$ . Using (B.9), expression (B.7) can be rewritten as

$$\begin{aligned}
Q(\Theta, \Theta^g) &= \sum_{\ell=1}^M \sum_{i=1}^N \log(c_{\ell} P(x_i|\theta_{\ell})) P(\ell|x_i, \Theta^g) \\
&= \sum_{\ell=1}^M \sum_{i=1}^N \log(c_{\ell}) P(\ell|x_i, \Theta^g) + \sum_{\ell=1}^M \sum_{i=1}^N \log(P(x_i|\theta_{\ell})) P(\ell|x_i, \Theta^g) \quad (\text{B.10})
\end{aligned}$$

Therefore when maximising  $Q$ , the term containing  $c_{\ell}$  can be maximised independently of the term containing  $\theta_{\ell}$ . Both terms are weighted sum over the probabilities that the data was drawn from a particular component density.

<sup>1</sup>In this dissertation, the responsibilities  $P(y_i|x_i, \Theta^g)$  are the edge label probabilities  $P(\mathbf{e}|\mathbf{D}_1\mathbf{D}_2)$ , assigned by considering the edge sample point errors under each motion.

### B.2.1 Finding the weights $c_\ell$

When solving (B.10) for  $c_\ell$ , there is the additional constraint that the mixture weights must sum to 1, i.e.  $\sum_\ell c_\ell = 1$ . This constraint can be added to the maximisation using a Lagrange multiplier  $\lambda$ , giving:

$$\begin{aligned} \frac{\partial}{\partial c_\ell} \left[ \sum_{\ell=1}^M \sum_{i=1}^N \log(c_\ell) P(\ell|x_i, \Theta^g) + \lambda \left( \sum_\ell c_\ell - 1 \right) \right] \\ = \sum_{i=1}^N \frac{1}{c_\ell} P(\ell|x_i, \Theta^g) + \lambda = 0 \end{aligned} \quad (\text{B.11})$$

and hence

$$c_\ell = -\frac{1}{\lambda} \sum_{i=1}^N P(\ell|x_i, \Theta^g) \quad (\text{B.12})$$

Summing both sides over  $\ell$ , and remembering that  $\sum_\ell c_\ell = \sum_\ell P(\ell|x_i, \Theta^g) = 1$ , gives that  $\lambda = -N$ , resulting in

$$c_\ell = \frac{1}{N} \sum_{i=1}^N P(\ell|x_i, \Theta^g) \quad (\text{B.13})$$

Where the  $P(\ell|x_i, \Theta^g)$  are simply the data label probabilities determined in the E-stage.

### B.2.2 Finding the model parameters $\theta_\ell$

The details of the maximisation over  $\theta_m$  are dependant on form of the underlying component distributions  $P(x_i|\theta_\ell)$ . It is commonly assumed that the errors are Gaussian, i.e.

$$P(x_i|\theta_\ell) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{r_i^2}{2\sigma^2}\right) \quad (\text{B.14})$$

where  $r_i$  is the error between the observed data  $x_i$  and the data predicted by the model parameters,  $x(\theta_\ell)$ :

$$r^i = x_i - x(\theta_\ell) \quad (\text{B.15})$$

Using this distribution, the second maximisation in (B.10) can be written as



$$\begin{aligned}
 \frac{\partial}{\partial \theta_\ell} & \left[ \sum_{\ell=1}^M \sum_{i=1}^N \log(P(x_i|\theta_\ell)) P(\ell|x_i, \Theta^g) \right] \\
 &= \frac{\partial}{\partial \theta_\ell} \left[ \sum_{\ell=1}^M \sum_{i=1}^N \log \left( \frac{1}{\sqrt{2\pi\sigma^2}} \exp - \left( \frac{r^{i^2}}{2\sigma^2} \right) \right) P(\ell|x_i, \Theta^g) \right] \\
 &= \frac{\partial}{\partial \theta_\ell} \left[ \frac{-1}{2\sigma^2} \sum_{i=1}^N r^{i^2} P(\ell|x_i, \Theta^g) \right]
 \end{aligned} \tag{B.16}$$

is identical to finding the parameters by minimising the weighted squared error in  $r^i$ :

$$\arg \min_{\theta_\ell} \sum_i w(r^i) r^{i^2} \tag{B.17}$$

where in this case the label probability  $P(\ell|x_i, \Theta^g)$  is the weight function  $w(r^i)$ . This can be solved by standard techniques.

### B.3 The M-stage for edge motion parameters

In this dissertation, the M-stage estimates the motion parameters  $\Theta$  from the residual edge errors, given the edge label probabilities  $P(\mathbf{e}|\Theta\mathbf{D})$  from the E-stage. As indicated by (B.17) these can be calculated a motion at a time by weighted least squares, where in each case sample is weighted by the probability that its edge obeyed that motion. Although the residual errors do not follow a Gaussian distribution (see Section 4.4.4), this is accommodated by another weighting term (an M-estimator) as described in Appendix A. Each sample is therefore weighted by two terms, as highlighted in Table 4.7.

For a number of reasons, this implementation of the M-stage does not achieve optimality. To calculate the maximum likelihood estimate of the motion parameters using an M-estimator, the least squares solution must be iterated. In this implementation, however, only one iteration is used, partly for reasons of speed, but also because it is used with the larger EM loop, which itself leads to the solution being iterated. This in itself does not affect convergence, since EM only requires an *increase* in likelihood at each iteration, rather than a full maximisation.<sup>2</sup> Instead, the non-optimality in the implementation come from the data used in the optimisation, the sample points. For M-estimators to converge, the data is required to be linear and continuous, neither of which are strictly true in this application (particularly since new sample point matches are found at each iteration). The approach, nonetheless, is still reasonable approximation, and achieves good results.

<sup>2</sup>If the M-step does not perform a full maximisation, this is then referred to as the *generalised* EM algorithm, or GEM [43].



# The independence of sample points

---

## C.1 Introduction: Edges and sample points

Edges form the fundamental basis of this thesis. They are used to estimate the motion between frames, and are labelled with the probability that they move according to each of the motions. This edge labelling then enables the rest of the frame to be segmented into the regions which obey each motion.

For both the estimation of a motion from an edge and its labelling, *sample points* are taken at regular intervals along the edge. This appendix considers the labelling of the edge motion probabilities from these sample points. In Section 4.4.4 (which considered the case of two motions), two simplifying assumptions were made: that the data observed by an edge under *each motion* are independent; and that the data observed by sample points *along an edge* are independent. These two assumptions are tested in the following sections.

## C.2 Errors under different motions

The first stage in assigning a motion probability is to transform the edge under each motion, and for its sample points to measure the residual error (the distance to the nearest edge) in each case. These two sets of readings comprise the data  $\mathbf{D}_1$  and  $\mathbf{D}_2$

‘correct’ mean	1.4
‘correct’ variance	10.9
‘incorrect’ mean	4.7
‘incorrect’ variance	27.8
covariance	2.8
correlation	0.16

Table C.1: *Correlation of sample point distances under each motion.* (Using absolute residual distances.) As would be expected, there is a larger variance in distances under the incorrect motion. The covariance term and the correlation are both small, but significant, indicating that the two measurements are *not* independent.

upon which the probabilities are based. In the implementation described in Chapter 4, these are assumed to be independent.

Clearly, the two sets of sample point data are independent if the edge maps to a completely different part of the frame under each motion. However, the inter-frame motion is usually small and, if under the correct motion the edge finds a match, then under the incorrect motion the errors will be only a little worse. A known case of dependence is where an edge which is weak in the first frame, or changes appearance greatly, fails to find a match in the second frame under either motion.

Specimen sample points have been taken from thirty test sequences, using one pair of frames from each. Using a hand-labelling, the errors measured under each of the two motions (‘correct’ and ‘incorrect’) have been gathered for each sample point. In all, 6782 sample points were analysed. Some statistics derived from this data are shown in Table C.1. The important figures from this table are the covariance and the correlation. Both of these figures are relatively large, which indicate that, on a sample point-by-sample point basis, there *is* a significant correlation between the error under each motion. A  $\chi^2$  independence test indicates that there is a vanishingly small probability of independence.

### C.2.1 The effect of assuming independence

The independence assumption is a very convenient simplification but, in the light of these findings, its applicability must be questioned. However, the probabilities returned by the implementation, which assumes independence, do not appear unreasonable. Given the test sample point data, a joint probability distribution over both sets of data has been estimated. This has been compared with a distribution which assumes independence and two revealing results have been determined:

**Mean probability error** The mean absolute difference between the probability estimated using the full, dependent, distribution and the distribution assuming

independence has been evaluated. The difference is 0.05 which, while not insignificant, is small enough to be considered reasonable.

**Labelling error** The correct motion label for each sample point is known, and this can be compared with the most likely labelling given by each sample point's probabilities. Under the full distribution, 91.3% of sample points are correctly labelled from their probabilities. Under the independence assumption this only falls to 89.6%.

These two results, together with the empirical evidence from the performance of the completed system, indicate that, while it is true that the data are not independent, the system nevertheless performs well if independence is assumed.

### C.3 Errors along an edge

Having found the sample point errors for a motion, the probability of that edge fitting the motion must be determined. The simplifying assumption here is that the sample points along the edge are independent, i.e. that knowing the errors at some sample points gives no information about the errors at the remaining points. This assumption is somewhat suspect, and this section considers an alternative model.

A now-standard way to consider dependence along a chain of samples is with a Markov chain [61]. In a first-order Markov chain, as will be considered here, the measurement at a point is dependent on the measurement at the previous point in the chain.<sup>1</sup> This dependence is expressed in terms of *transition probabilities*: in this case the probability of a particular sample point error given the previous error. These chain transition probabilities have been modelled from the sample point data for each of the two motions ('correct' and 'incorrect'), and these are displayed in Figure C.1. In these figures, the (known) error at the current sample point—the  $x$ -axis—determines the error distribution for the next sample point, given by the relevant column of the matrix.

Figure C.1(a) shows the transition probabilities for sample points on edges under the 'correct' motion. It can be seen that, regardless of the error at the current point, a low residual distance is likely—the errors are largely independent. This is true for all errors apart from the case when no match was found; in this case, a failed match at the next sample point is highly likely. Under the 'incorrect' motion (Figure C.1(b)), the errors are clearly not independent. Whatever the current error, the next

---

<sup>1</sup>A similar chain was used by MacCormick and Blake [91] to make their contour matching more robust to occlusion.

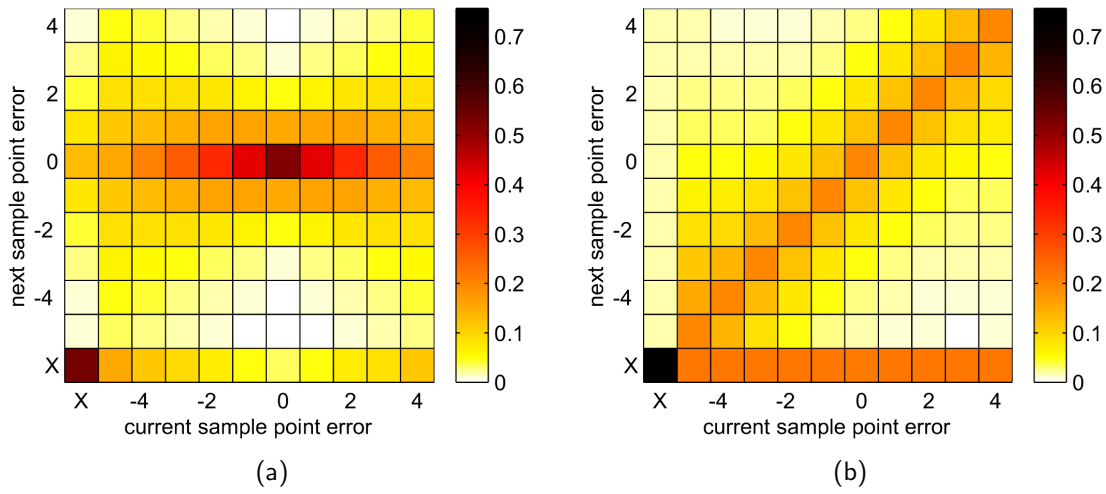


Figure C.1: *Markov chain transition probabilities..* (a) 'correct motion' chain; (b) 'incorrect motion' chain. The error at the current sample point indicates from which column the probability for the next sample point error should be taken. 'X' indicates the case where no match is found.

error is likely to be the same, or a failed match. The first case corresponds to the situations where an edge under the 'incorrect' motion still maps close to an edge, but is small distance away along its whole length.

These results indicate that, while the independence assumption is somewhat justified for the 'correct' motion, it is not appropriate under the 'incorrect' motion. The Markov chain described here provides a model of this dependence, and it has been tested with the system described Chapter 4. Perhaps surprisingly, however, the final segmentation performance is similar under both schemes. The failings of the independence assumption can be observed in the edge probabilities in some sequences—those with edges which could obey either motion. The probabilities in these cases should be very similar but, by assuming independence, a single point can make a large difference to the outcome, saturating the probabilities hugely in favour of one motion. This gives edge probabilities which are not as uncertain as they should be, with edges labelled with high probability in favour of one motion although they are, in truth, considerably more ambiguous. However, for these edges, the motion with the highest probability is still usually the correct one, and the region labelling scheme is able to ignore many of the errors which occur. For simplicity, therefore, given the minimal performance loss, sample point independence is adopted. Nonetheless, approaches such as this Markov chain scheme are worthy of future research, and should yield increased robustness.

## Complete multiple-frame results

---

### D.1 Introduction

This appendix presents the results of testing the edge-based motion segmentation scheme of Chapters 4 (two-motions, two-frames) and 6 (two-motions, multiple frames) on thirty-four different image sequences.

#### D.1.1 Image sequences

The image sequences used fall into three categories:<sup>1</sup>

**Standard MPEG-4 test sequences.** There are number of sequences commonly used for testing video segmentation and coding performance. Of these, **Coastguard**, **Foreman**, **FlowerGarden**, **HallMonitor** and **Tennis** have been tested over the course of this work.

**Terrestrial TV footage, from AT&TV.** Sequences have kindly been made available by AT&T Laboratories, from their AT&TV project [98], which maintains a seven-day archive of the four main terrestrial channels. Twenty-five sequences from this archive were selected in February 2001 for testing.

**Home movies.** Four additional sequences were taken from camcorder footage in and around Cambridge.

---

<sup>1</sup>And a selection of these test sequences are available for download from <http://www-svr.eng.cam.ac.uk/~pas1001/Publications/videos.html>.

### D.1.2 Algorithm

All sequences were segmented automatically, first using the two-frame implementation of Chapter 4, and then further frames were segmented using the multiple-frame implementation described in Chapter 6. It is assumed that there are only *two motions* present in each sequence, and each motion is modelled by a 6-parameter *affine* model. Unless stated, the settings are the same for each sequence; in seven of the sequences it was necessary to either reduce the edge detection hysteresis thresholds to identify weaker edges, or skip frames in order to speed up slow motions. These are identified in the discussion accompanying each sequence.

### D.1.3 Presentation of results

For each sequence, the final edge probabilities and region labels are shown after two frames, three frames, and eleven frames have been processed (thereby being the labels for frames 1, 2 and 10). The edge probabilities are shown as a blend between red and green, where the red component indicates the probability of motion 1 and green the probability of motion 2. The region labelling shows the foreground regions.


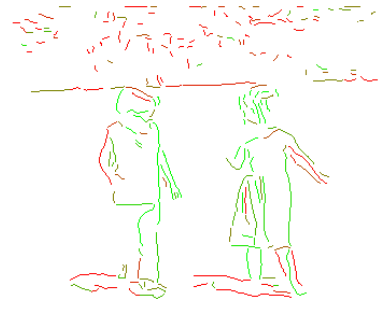
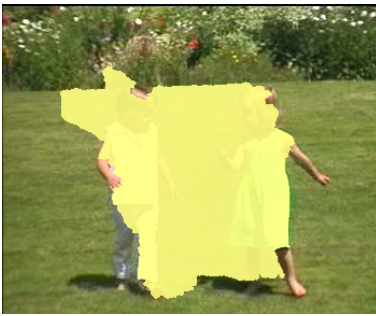
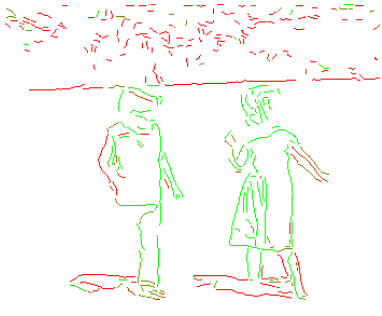



Two statistics are shown: the percentage of edges which would be labelled correctly if each were assigned to its most likely label, and the percentage of pixels labelled correctly as foreground or background. In each case this figure is a comparison with a hand-labelling of the image regions given by the static segmentation.


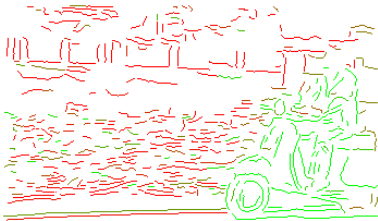



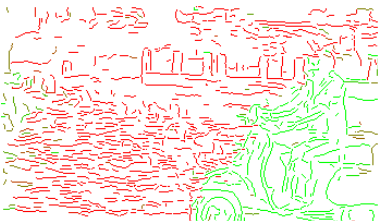

The results are discussed in Chapters 5 and 7 (for the two- and multiple-frame cases respectively). These chapters make reference to some of the sequences in this appendix; the sequences are referred to by name and are in alphabetical order.





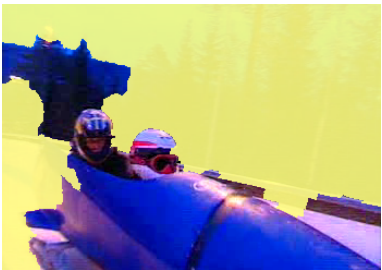
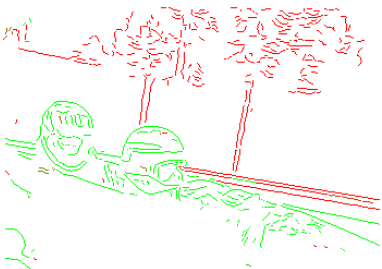

## D.2 Results


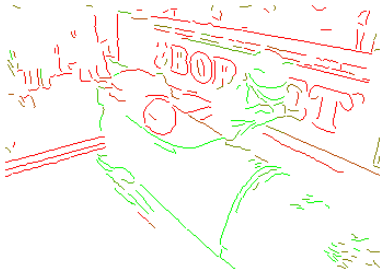

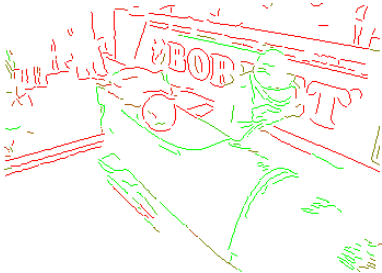

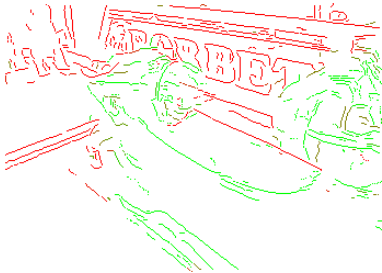
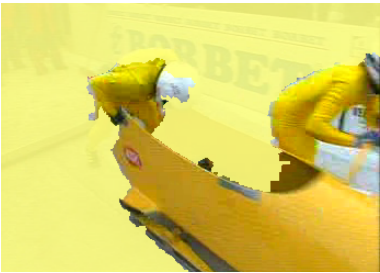
The individual results may be seen on the remaining pages of this appendix.










<b>AHvid</b> Home movie converted to MPEG (with thanks to Andy Hopper). The two girls walk together towards the camera and slightly to the right.		
Frame 1	 Edges correct: 76.3%	 Pixels correct: 11.7%
Frame 2	 Edges correct: 82.8%	 Pixels correct: 97.8%
Frame 10	 Edges correct: 78.9%	 Pixels correct: 84.9%
Since they are walking together, the two children are well modelled by one motion model, apart from the movement of the arms and legs which means they are sometimes incorrectly labelled. No edges are detected on the grass, which means that labelling this is difficult, and also that the only T-junctions are where the heads cross the edge of the lawn. As a result the layer ordering is incorrect in the first frame.		

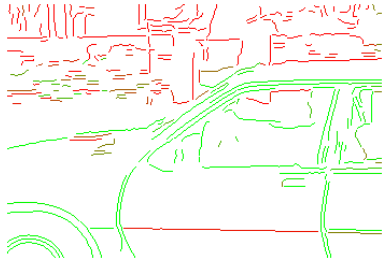

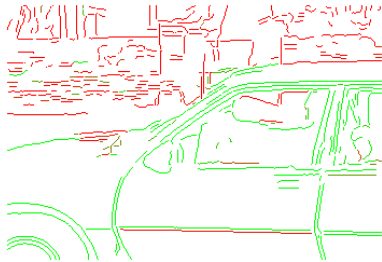



<div><div>Bike</div><div>A moped being tracked as it drives right to left. Sequence taken from AT&amp;TV database, from BBC2's 'The Bike's the Star'.</div></div>		
Frame 1	<div><div>Edges correct: 87.0%</div></div>	<div><div>Pixels correct: 91.0%</div></div>
Frame 2	<div><div>Edges correct: 87.7%</div></div>	<div><div>Pixels correct: 90.1%</div></div>
Frame 10	<div><div>Edges correct: 86.7%</div></div>	<div><div>Pixels correct: 85.5%</div></div>
<p>Both motions are horizontal, and many edges could fit either motion, which makes convergence difficult. Also, the independent motion is confined to one small area of the screen, and easily picks up edges from other parts of the screen. Nonetheless, the system performs creditably. The road genuinely is ambiguous (the moped could be on a conveyer belt), so errors in the segmentation of this are to be expected.</p>		

<b>Bobsled1</b> Bobsled coming to a stop towards the camera. Sequence taken from AT&TV database, from Channel 4's 'Transworld Sport'.			
Frame 1	 <p>Edges correct: 73.3%</p>	 <p>Pixels correct: 89.3%</p>	
Frame 2	 <p>Edges correct: 71.0%</p>	 <p>Pixels correct: 86.6%</p>	
Frame 10	 <p>Edges correct: 93.2%</p>	 <p>Pixels correct: 98.9%</p>	
<p>Sequences of a bobsled at full speed proved impossible to segment, due to both motion blur and a very large image motion. This sequence of the bobsled slowing down segments well. The edges are well labelled, apart from some ambiguities along the line of motion, and a reasonable segmentation is produced which becomes excellent with more observations.</p>			


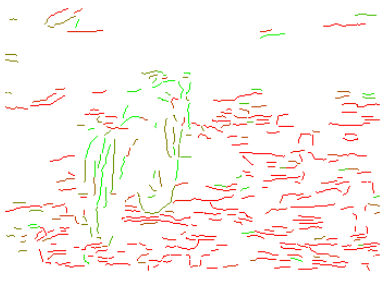

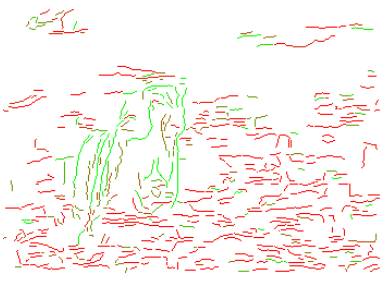

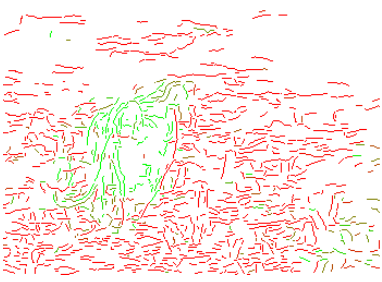

<div><div>Bobsled2</div><div>Bobsled team starting a run. Sequence taken from AT&amp;TV database, from Channel 4's 'Transworld Sport'.</div></div>		
Frame 1	<div><div>Edges correct: 77.8%</div></div> <div><div>Pixels correct: 81.1%</div></div>	
Frame 2	<div><div>Edges correct: 82.6%</div></div> <div><div>Pixels correct: 89.7%</div></div>	
Frame 10	<div><div>Edges correct: 85.6%</div></div> <div><div>Pixels correct: 88.7%</div></div>	
<p>The motion of the foreground is projective rather than affine, but the sled is tracked well even with an affine motion model. The motion of the brakeman is close to that of the background for the first few frames (as she pushes off), so it is a while before she is segmented. The top edge of the sled is not detected by the Canny edge detector, although the segmenter still pulls the sled out well. The final segmentation is reasonable.</p>		








<div><div>Buffy</div><div>Faith (dark hair) talks to Buffy. Sequence taken from AT&amp;TV database, from BBC 2's 'Buffy the Vampire Slayer'.</div></div>		
Frame 1	<div><div>Edges correct: 70.1%</div></div>	<div><div>Pixels correct: 73.5%</div></div>
Frame 2	<div><div>Edges correct: 81.3%</div></div>	<div><div>Pixels correct: 85.9%</div></div>
Frame 10	<div><div>Edges correct: 89.2%</div></div>	<div><div>Pixels correct: 91.6%</div></div>
<p>There are very few edges in the foreground—just some of the occluding boundary and areas of skin. With many more edges in the background, and the foreground motion small, the EM does not find a very good solution. After a few frames the edge labelling is good but, with large parts of the occluding boundary missing, only part of the foreground is segmented.</p>		



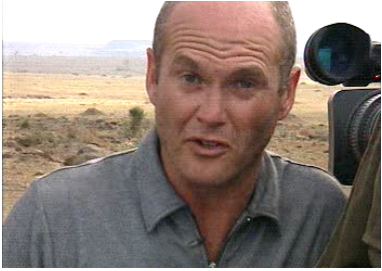

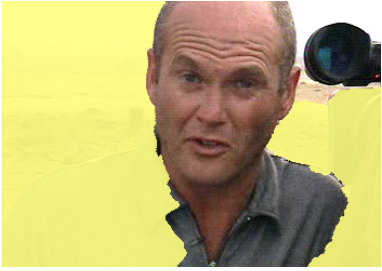

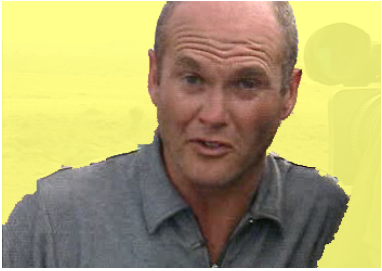

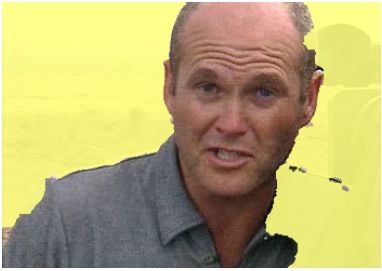
Car	
Camera tracking a car as it moves from right to left. Sequence recorded using a hand-held MPEG camera.	
Frame 1	 <p>Edges correct: 87.7%</p>
	
Frame 2	 <p>Edges correct: 91.9%</p>
	 <p>Pixels correct: 95.7%</p>
Frame 10	 <p>Edges correct: 93.5%</p>
	 <p>Pixels correct: 92.2%</p>

The car has a few edges along the line of motion which are labelled ambiguously, and the reflections on the roof cause a few problems. But the EM converges to a good solution and the car is well segmented. This sequence is discussed in detail in Chapters 5 and 7.

<b>Cats1</b> Camera tracking a lion running. Sequence taken from AT&TV database, from BBC 2's 'Big Cat Diary'.		
Frame 1	 <p>Edges correct: 87.9%</p>	 <p>Pixels correct: 97.4%</p>
Frame 2	 <p>Edges correct: 84.1%</p>	 <p>Pixels correct: 96.9%</p>
Frame 10	 <p>Edges correct: 87.6%</p>	 <p>Pixels correct: 97.1%</p>
<p>A very difficult subject, with a significant amount of motion blur. The lion quite definitely does not fit an affine motion, with the legs and tail undergoing large motions between frame. Regardless, the EM converges well and the lion's body is consistently segmented. The legs are occasionally included when the motion is close enough. There is little chance for the probabilities to be refined over the sequence since the tracker error is usually too great for them to be propagated between frames, but those that are propagated ensure that the edge labels are improved with time.</p>		

<div>Cats2</div> <div>Camera following a Land Rover driving across a plain. Sequence taken from AT&amp;TV database, from BBC 2's 'Big Cat Diary'.</div>		
Frame 1	<div><div>Edges correct: 70.8%</div></div> <div><div>Pixels correct: 89.9%</div></div>	
Frame 2	<div><div>Edges correct: 82.5%</div></div> <div><div>Pixels correct: 93.9%</div></div>	
Frame 10	<div><div>Edges correct: 94.8%</div></div> <div><div>Pixels correct: 98.8%</div></div>	
<div>The Land Rover's edges are well separated from the back-ground in the EM process, and the Land Rover is well segmented. The dust and the shadow, of course, are occasionally segmented with the Land Rover. After a few frames, a very good segmentation results.</div>		



<div>Cats3</div> <div>Simon King talking to the camera. Sequence taken from AT&amp;TV database, from BBC 2's 'Big Cat Diary'.</div>		
Frame 1	 <div>Edges correct: 80.2%</div>	 <div>Pixels correct: 75.1%</div>
	 <div>Edges correct: 93.3%</div>	 <div>Pixels correct: 99.9%</div>
Frame 10	 <div>Edges correct: 82.9%</div>	 <div>Pixels correct: 94.6%</div>
<div>Simon King moves his head enough while talking to segment well, and his shoulders soon follow suit. Part of the camera tripod is in <i>front</i> of the 'foreground', breaking the layered assumption, but the system copes with this small anomaly.</div>		

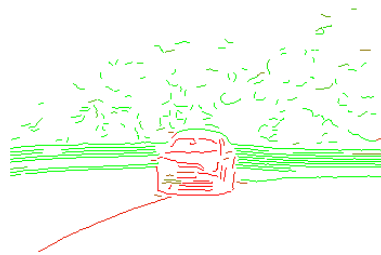


# Driven1

A car driving around a corner towards the camera. Sequence taken from AT&TV database, from Channel 4's 'Driven'.



Frame 1



---

Edges correct: 91.8%

---

Pixels correct: 99.2%

Frame 2



Edges correct: 87.8%



---

Pixels correct: 99.7%

Frame 10

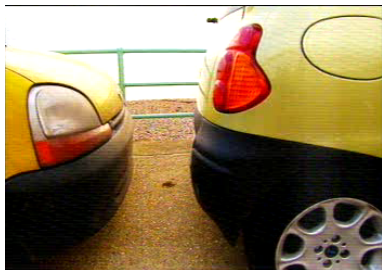
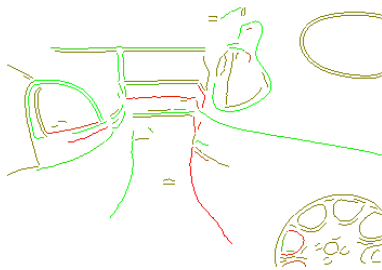

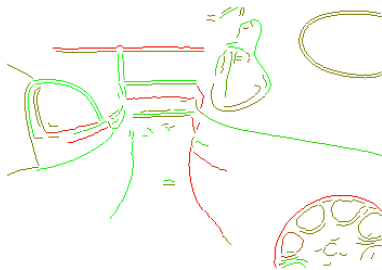

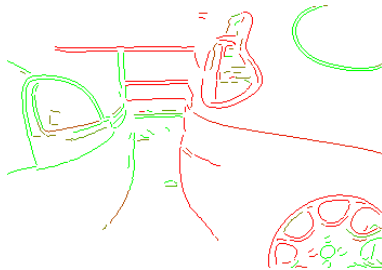



Edges correct: 81.6%



Pixels correct: 99.8%

Even with a small number of edges describing the car, the motion is fitted well and an excellent edge labelling and segmentation results. The edge of the road along the line of motion is mislabelled in the early frames, but does not affect the solution.

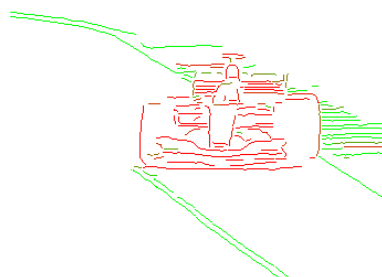
<h1>Driven2</h1> <p>A close-up of a car (on the right) slowly reversing into a parking space. Sequence taken from AT&amp;TV database, from Channel 4's 'Driven'.</p>				
Frame 1		Edges correct: 41.0%		Pixels correct: 29.3%
Frame 2		Edges correct: 42.7%		Pixels correct: 75.6%
Frame 10		Edges correct: 69.0%		Pixels correct: 75.5%
<p>This sequence is not a good candidate for segmentation. The inter-frame motion of the car is very small, and many of the background edges are horizontal, and thus ambiguous. One of the main sources of foreground edges, the wheel, is rotating and so moving with a different motion from that of the rest of the car. EM never converges to a good solution and the segmentations are all poor. Over a large number of frames a reasonable labelling of some edges is produced, but still not enough for a particularly good segmentation.</p>				

# F1

Michael Schumacher's Ferrari driving towards the camera. Sequence taken from AT&TV database, from ITV's coverage of the Australian Grand Prix.



Frame 1



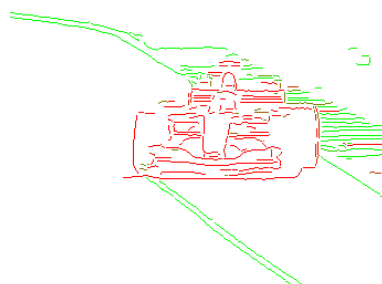
Edges correct: 88.7%



Pixels correct: 99.3%

---

Frame 2

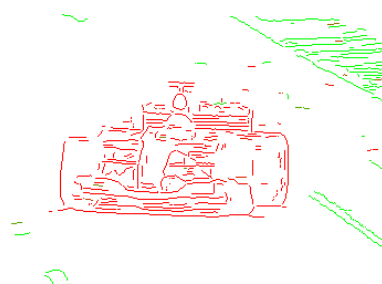


Edges correct: 91.4%



Pixels correct: 99.8%

Frame 10








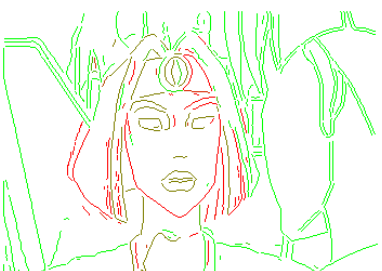
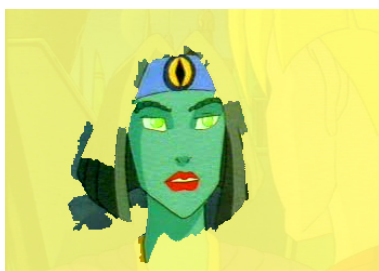
Edges correct: 93.5%


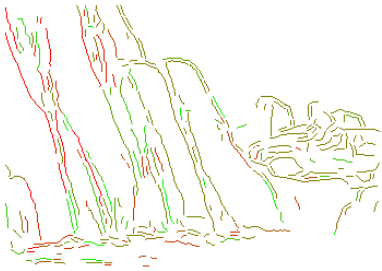

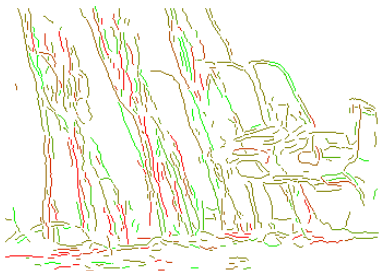

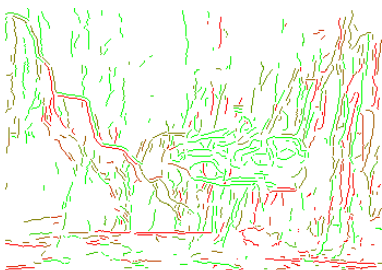





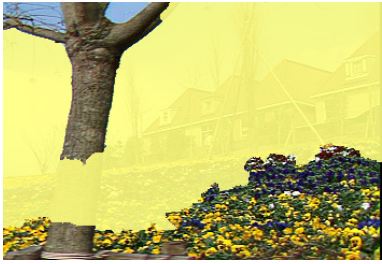
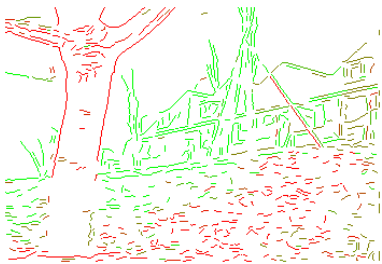

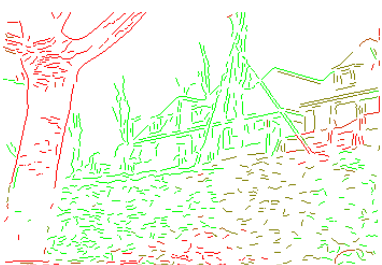

Pixels correct: 99.9%

A sequence which gives the segmentation scheme no trouble, with a good edge labelling and an excellent final segmentation. The large featureless expanse of tarmac is extracted as one region by the static segmenter and correctly labelled thanks to the white lines.




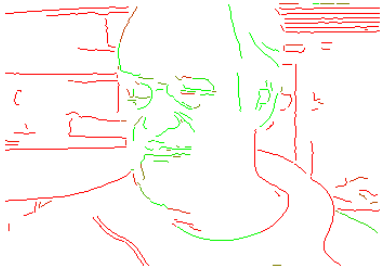










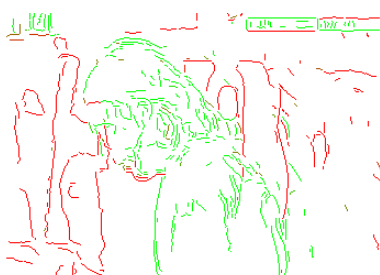

<b>FlashGordon1</b> Cartoon of a woman talking to Flash. Sequence taken from AT&TV database, from Channel 4's 'Flash Gordon'.		
Frame 1	 Edges correct: 83.4%	 Pixels correct: 92.7%
Frame 3	 Edges correct: 85.7%	 Pixels correct: 90.4%
Frame 19	 Edges correct: 65.4%	 Pixels correct: 86.2%
<p>Cartoons would be expected to segment well, and the static segmentation is indeed good. However, the motion is rather more difficult to establish. The first problem is that many cartoons (including this one) are only filmed at 15Hz, and so there is only motion in every other frame. Here, therefore, the settings have been changed to consider every other frame. The second problem is that much of the content of neighbouring frames is identical. Making cartoon characters talk is commonly achieved by keeping the frame static and animating only the necessary facial features, which provide too few features to successfully track. In the example here, the head does move as she begins talking, which allows it to be segmented. But, as the sequence progresses the only motion is in the lips, and the head begins to be considered as background.</p>		

<b>FlashGordon2</b> Cartoon tracking Flash flying his speeder through a canyon. Sequence taken from AT&TV database, from Channel 4's 'Flash Gordon'.		
Frame 1	 Edges correct: 35.8%	 Pixels correct: 41.1%
Frame 3	 Edges correct: 43.5%	 Pixels correct: 55.9%
Frame 19	 Edges correct: 46.0%	 Pixels correct: 44.1%
The problem with animating fast-moving objects at only 15Hz, as this cartoon does, is that the inter-frame motions are then very large. The background in this sequence moves about 60 pixels between frames, which is far too large to be effectively (and efficiently) tracked. A random edge labelling, and a random segmentation results.		

<div>FlowerGarden</div> <div>The camera moves to the right and the tree, closer to the camera, has a different image motion from that of the flowerbed and houses. Part of a standard test sequence.</div>		
Frame 1	 <div>Edges correct: 53.8%</div>	 <div>Pixels correct: 70.9%</div>
	 <div>Edges correct: 60.1%</div>	 <div>Pixels correct: 75.0%</div>
	 <div>Edges correct: 78.9%</div>	 <div>Pixels correct: 79.6%</div>
<div>The motion of tree's edges is well labelled as distinct from the background, but the edges in the flowerbed close to the camera are also labelled with this motion. In addition, the edges on the far right can also be included in the foreground motion while still leaving the tree's edges in good agreement. Another problem is that the Canny edge detector does not pick up the edge of the tree against the flowerbed, so part of the tree is missing here.</div>		










<b>Food&amp;Drink</b> Antony Worrall Thompson turning his head to the right. Sequence taken from AT&TV database, from BBC 2's 'Food and Drink'.		
Frame 1	 <p>Edges correct: 65.7%</p>	 <p>Pixels correct: 25.8%</p>
Frame 2	 <p>Edges correct: 67.4%</p>	 <p>Pixels correct: 70.3%</p>
Frame 10	 <p>Edges correct: 76.6%</p>	 <p>Pixels correct: 79.1%</p>
<p>The motion of his head is non-affine, but it is reasonably well tracked by the affine model with the sample point propagation. Unfortunately the occluding edge at the back of his head is not extracted and, together with edge label errors, this means that the layer ordering is incorrectly identified in the first frame. This error is corrected in subsequent frames but, with the occluding edge still missing, the segmentation can still not be complete. The statistics quoted are poorer than they could perhaps be since the hand segmentation in this case expected his shoulders also to be segmented.</p>		

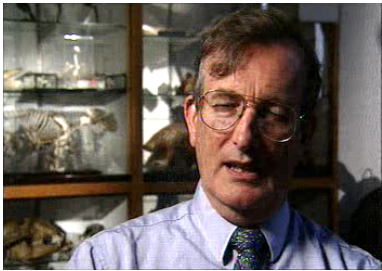






<div><div>Football</div><div>Close-up following a footballer walking to the left. Sequence taken from AT&amp;TV database, from Channel 4's 'Football Italia'.</div></div>		
Frame 1	<div><div>Edges correct: 89.1%</div></div> <div><div>Pixels correct: 75.1%</div></div>	
Frame 2	<div><div>Edges correct: 91.8%</div></div> <div><div>Pixels correct: 98.6%</div></div>	
Frame 10	<div><div>Edges correct: 81.6%</div></div> <div><div>Pixels correct: 94.1%</div></div>	
<p>One of the problems of telephoto lenses is their small depth of field. In this case, this means that the background is out of focus, and the Canny edge detector picks up very few edges. There are, however, just about enough to be tracked independently of the foreground. After a few frames the correct background edge labels are established and a good segmentation results. The edge-based system, of course, fails on general football shots where there are many different motions of small objects.</p>		


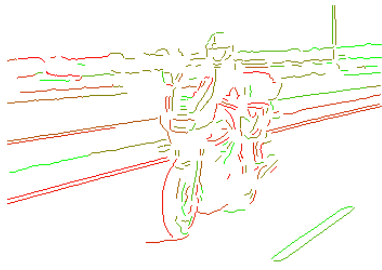

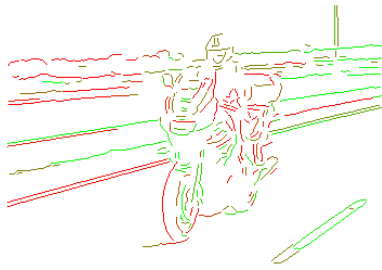

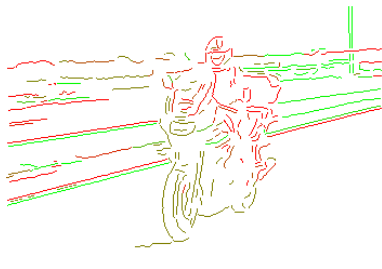
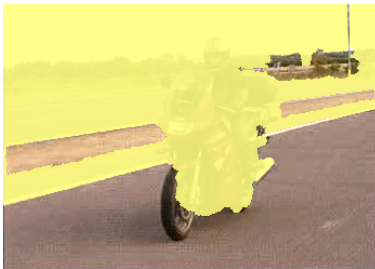


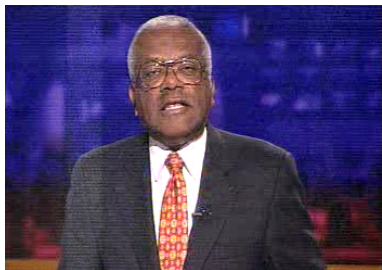


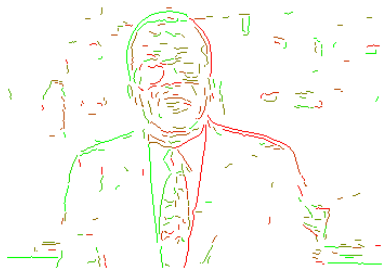

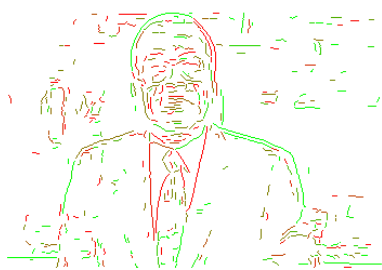
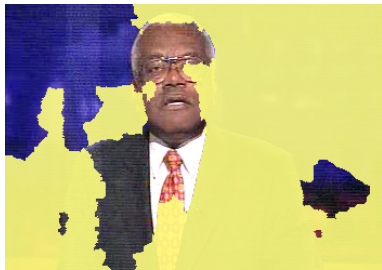


<h1>HallMonitor</h1> <p>A stationary camera views a man walking down a corridor. Part of a standard test sequence.</p>			
Frame 1	 <p>Edges correct: 66.5%</p>	 <p>Pixels correct: 95.9%</p>	
Frame 2	 <p>Edges correct: 72.8%</p>	 <p>Pixels correct: 96.9%</p>	
Frame 10	 <p>Edges correct: 73.2%</p>	 <p>Pixels correct: 85.3%</p>	
<p>The system performs excellently here, particularly given the small number of edges representing the foreground object. The edges of the cubicles on the right are ambiguous in many frames, and it is perhaps fortunate that it is not until the tenth frame that they are labelled in error, and segmented incorrectly.</p>			



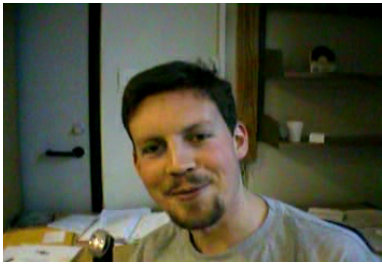
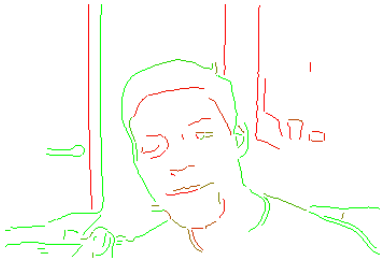





<div><div>Horizon1</div><div>Man talking to the camera. Sequence taken from AT&amp;TV database, from BBC 2's 'Hori- zon'.</div></div>		
Frame 1	<div><div>Edges correct: 48.5%</div></div>	<div><div>Pixels correct: 24.8%</div></div>
Frame 11	<div><div>Edges correct: 68.0%</div></div>	<div><div>Pixels correct: 27.2%</div></div>
Frame 91	<div><div>Edges correct: 57.2%</div></div>	<div><div>Pixels correct: 75.3%</div></div>
<p>People move surprisingly little as they talk, and in this case the inter-frame motion is substantially less than one pixel. If the sequence is instead sampled every 10 frames, there is sufficient motion to label the head motion as independent from the background. Unfortunately, the occluding edge of his head which is in shadow is not picked out by the Canny edge detector, so that the side of his head bleeds into the background and is labelled as background. Together with the noisy edge labels, this makes the layer ordering difficult to determine and it selects the incorrect layer as foreground, giving a very poor segmentation. By the tenth frame the random nature of the edge labelling has labelled the head with the other motion, and so it is now (by chance) better segmented.</p>		


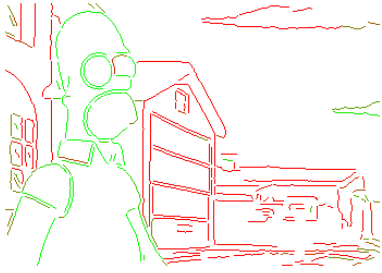

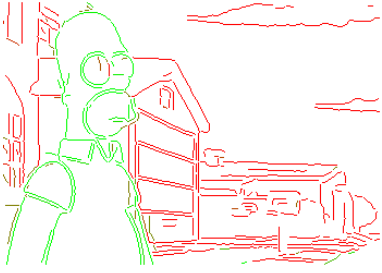

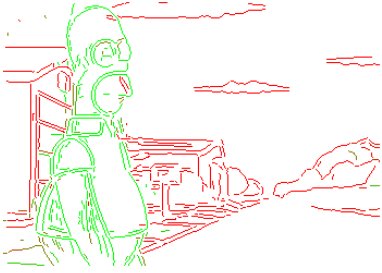

<h1>Horizon2</h1> <p>Camera tracking a motorbike as it drives down a road. Sequence taken from AT&amp;TV database, from BBC 2's 'Horizon'.</p>			
Frame 1	 <p>Edges correct: 55.7%</p>	 <p>Pixels correct: 42.8%</p>	
Frame 2	 <p>Edges correct: 46.4%</p>	 <p>Pixels correct: 79.0%</p>	
Frame 10	 <p>Edges correct: 32.4%</p>	 <p>Pixels correct: 42.7%</p>	
<p>While there are only two rigid motions in this sequence, the background motion is highly projective and so is not selected as one motion. When a full-projective motion model is used, EM fails to converge to a reasonable solution. Instead the edges are shared between the background and the motorbike, and a near-random segmentation results.</p>			

ITN			
Frame 1			
	Edges correct: 46.3%	Pixels correct: 79.7%	
Frame 11			
	Edges correct: 53.8%	Pixels correct: 68.8%	
Frame 91			
	Edges correct: 53.2%	Pixels correct: 59.4%	
<p>As with many ‘talking heads’, Trevor McDonald does not move much as he talks, and a sample every 10 frames is required to yield a visible inter-frame motion. Unfortunately, the background on this news program is also very smooth, with very few edges detected even with greatly reduced hysteresis thresholds (10&amp;10). The movement of his shoulder is detected in the first few frames, but the labelling of the rest of the edges is poor and a poor segmentation is the result.</p>			











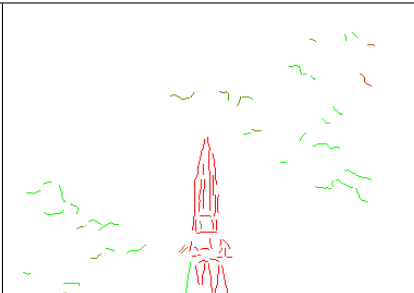
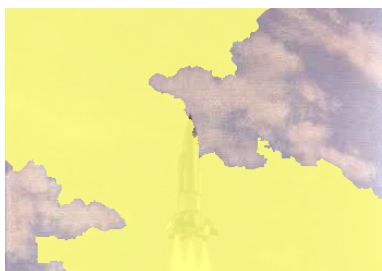






<div><div>Nick</div><div>AT&amp;T's Nick Hollinghurst posing for the camera (with thanks to Nick). Here he tips his head back. Sequence recorded using a hand-held MPEG camera.</div></div>		
Frame 1	<div><div>Edges correct: 44.8%</div></div>	<div><div>Pixels correct: 61.4%</div></div>
Frame 2	<div><div>Edges correct: 81.9%</div></div>	<div><div>Pixels correct: 93.2%</div></div>
Frame 10	<div><div>Edges correct: 95.2%</div></div>	<div><div>Pixels correct: 99.1%</div></div>
<div>The motion of Nick's head is difficult to model with an affine motion. Also few background edges intersect with foreground edges, which makes the layer ordering difficult to determine. The edge labelling is poor in the first frame, but by the second frame the edges are well labelled and after more evidence they give an excellent segmentation.</div>		


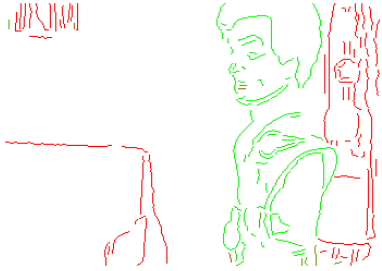





<p><b>Simpsons</b></p> <p>Cartoon following Homer Simpson walking to the right. Sequence taken from AT&amp;TV database, from BBC 2's 'The Simpsons'.</p>		
Frame 1	 <p>Edges correct: 89.8%</p>	 <p>Pixels correct: 99.5%</p>
Frame 2	 <p>Edges correct: 95.2%</p>	 <p>Pixels correct: 98.9%</p>
Frame 10	 <p>Edges correct: 92.0%</p>	 <p>Pixels correct: 99.7%</p>
<p>The Simpsons is more professionally produced than the Flash Gordon cartoon considered earlier in the test set—the frames are at a full 25Hz, and there is more animation between frames. This means that the motion of Homer can be very easily detected and with the edges and static segmentation trivial to detect and perform, an excellent segmentation results. A mosaic of the background to this sequence is shown in Section 6.7.</p>		






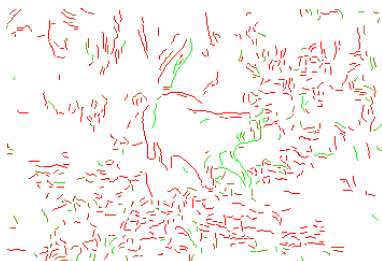



<div><div>Tennis2</div><div>Close-up of a tennis player walking away from the camera and to the right. Sequence taken from AT&amp;TV database, from Channel 4's 'Transworld Sport'.</div></div>		
Frame 1	<div><div>Edges correct: 70.1%</div></div>	<div><div>Pixels correct: 86.2%</div></div>
Frame 2	<div><div>Edges correct: 68.8%</div></div>	<div><div>Pixels correct: 83.2%</div></div>
Frame 10	<div><div>Edges correct: 91.8%</div></div>	<div><div>Pixels correct: 98.6%</div></div>
<p>Another case where the background is out of focus, and in this case Canny cannot detect any of the edges with the standard threshold. Lowering the upper threshold from 30 to 15 does generate enough background edges, and a reasonable segmentation results, which becomes excellent over time. The problem with a lower threshold is that many of the creases on the player's shirt are extracted as edges. These flutter as he walks, and sometimes the incorrect (i.e. background) motion model is fitted to them, leading to a poor segmentation there. This is another case where the edge labels improve greatly as evidence is gathered.</p>		



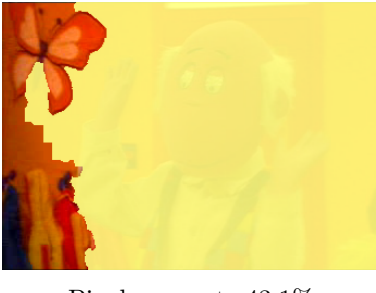




<h1>Thunderbirds1</h1> <p>Thunderbird 1 taking off against a cloudy sky. Sequence taken from AT&amp;TV database, from BBC 2's 'Thunderbirds'.</p>			
Frame 1			
	Edges correct: 81.1%	Pixels correct: 61.7%	
Frame 2			
	Edges correct: 90.5%	Pixels correct: 100.0%	
Frame 10			
	Edges correct: 86.9%	Pixels correct: 97.1%	
<p>Again, very few background edges; the example here shows a lowered upper hysteresis threshold of 15 and there are still barely sufficient to detect the background motion. However, it is detected and the edge labels throughout are excellent. With no interaction between the edges of different layers, the labelling is difficult to determine and it is incorrect in the first frame, but after that the simpler structure of the segmented rocket yields a more likely segmentation.</p>			


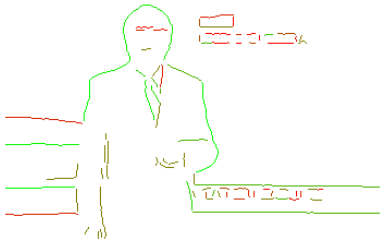
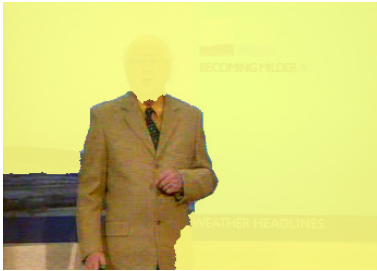

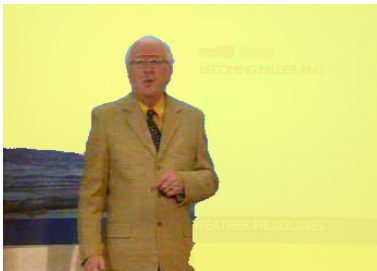
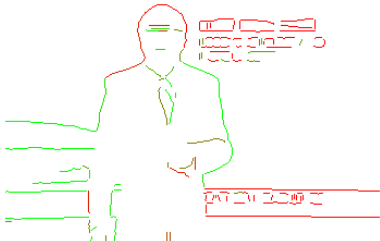
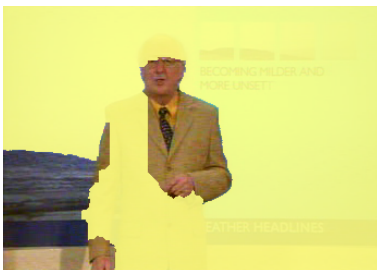


<h2>Thunderbirds2</h2> <p>The camera tracks Scott Tracy as he walks from right to left. Sequence taken from AT&amp;TV database, from BBC 2's 'Thunderbirds'.</p>		
Frame 1	 <p>Edges correct: 95.9%</p>	 <p>Pixels correct: 99.9%</p>
Frame 2	 <p>Edges correct: 94.6%</p>	 <p>Pixels correct: 99.1%</p>
Frame 10	 <p>Edges correct: 90.7%</p>	 <p>Pixels correct: 92.4%</p>
<p>The advantage of segmenting puppets is that they undergo less deformation as they walk than humans do. In this case, the edges are extracted well, and the motions are correctly estimated. An excellent edge labelling and segmentation results. In the final frame there are no motion-labelled edges bounding one of the background regions, apart from the foreground edge, meaning that it is ambiguous. In this case it is mislabelled.</p>		

<div><div>Trin</div><div>Trin the cat walking across the grass towards the camera and to the right (with thanks to Ken Wood). Sequence recorded using a hand-held MPEG camera.</div></div>		
Frame 1	<div><div>Edges correct: 73.7%</div></div>	<div><div>Pixels correct: 98.9%</div></div>
Frame 2	<div><div>Edges correct: 84.5%</div></div>	<div><div>Pixels correct: 92.4%</div></div>
Frame 10	<div><div>Edges correct: 78.8%</div></div>	<div><div>Pixels correct: 93.6%</div></div>
<div>An easy sequence for the static segmenter, but there are very few long edge contours extracted. Trin’s motion is, of course, non-affine but the edges associated with her are in general correctly identified, and some sample points are also propagated from frame to frame. The segmentation is a reasonable attempt given the difficult subject.</div>		



<b>Tweenies</b> Max dancing (moving right to left in the first few frames), with Jake entering the frame later. Sequence taken from AT&TV database, from BBC 2's 'Tweenies'.		
Frame 1	 Edges correct: 79.1%	 Pixels correct: 43.1%
Frame 2	 Edges correct: 74.1%	 Pixels correct: 73.5%
Frame 10	 Edges correct: 82.3%	 Pixels correct: 61.9%
<p>Like cartoons, brightly coloured children's TV characters would be expected to segment well, and the static segmentation is indeed good. The layer ordering is incorrect in the first case, with missing occluding boundaries and errors in the edge labelling, but the second frame is a reasonable attempt. Unfortunately, the amount and speed with which the characters bounce around is such that the motion is too large and non-affine for the EM to converge on the correct edge labelling later in the sequence, particularly with the addition of another moving object.</p>		

<b>Weather</b> Michael Fish presenting the weather. Between frames he moves a little to his right. Sequence taken from AT&TV database, from BBC 1's weather forecast.		
Frame 1	 Edges correct: 51.2%	 Pixels correct: 91.0%
Frame 2	 Edges correct: 65.4%	 Pixels correct: 91.1%
Frame 10	 Edges correct: 63.7%	 Pixels correct: 75.9%
Weather men present similar difficulties to news readers, but fortunately undergo more motion and have guaranteed structure in the background. Michael Fish moves enough as he talks for his edges to be labelled reasonably well, and there are just enough edges in the computer-generated backdrop to extract that and label it as background.		

---

# Bibliography

---

- [1] G. Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(4):384–401, July 1985.
- [2] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2(3):283–310, January 1989.
- [3] Elisabeth André, Gerd Herzog, and Thomas Rist. Multimedia presentation of interpreted visual data. In *Proc. AAAI-94 Workshop on Integration of Natural Language and Vision Processing*, pages 74–82, Seattle, WA, USA, June 1994.
- [4] S. Ayer and H. S. Sawhney. Layered representation of motion video using robust maximum-likelihood estimation of mixture models and MDL encoding. In *Proc. 5th International Conference on Computer Vision*, pages 777–784, Cambridge, MA, USA, June 1995.
- [5] S. Ayer, P. Schroeter, and J. Bigün. Segmentation of moving objects by robust motion parameter estimation over multiple frames. In *Computer Vision—ECCV '94 (Proc. 3rd European Conference on Computer Vision)*, volume 801 of *Lecture Notes in Computer Science*, pages 317–327, Stockholm, Sweden, May 1994. Springer-Verlag.

- [6] S. Baker, R. Szeliski, and P. Anandan. A layered approach to stereo reconstruction. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 434–441, Santa Barbara, CA, USA, June 1998.
- [7] D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice-Hall, New Jersey, 1982.
- [8] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, January 1994.
- [9] J. R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Computer Vision—ECCV '92 (Proc. 2nd European Conference on Computer Vision)*, volume 588 of *Lecture Notes in Computer Science*, pages 237–252, Santa Margherita Ligure, Italy, May 1992. Springer-Verlag.
- [10] J. R. Bergen, P. J. Burt, R. Hingorani, and S. Peleg. Computing two motions from three frames. In *Proc. 3rd International Conference on Computer Vision*, pages 27–32, Osaka, Japan, December 1990.
- [11] J. R. Bergen, P. J. Burt, R. Hingorani, and S. Peleg. A three-frame algorithm for estimating two-component image motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(9):886–895, September 1992.
- [12] L. Bergen and F. Meyer. Motion segmentation and depth ordering based on morphological segmentation. In *Computer Vision—ECCV '98 (Proc. 5th European Conference on Computer Vision)*, volume 1407 of *Lecture Notes in Computer Science*, pages 531–547, Freiburg, Germany, June 1998. Springer-Verlag.
- [13] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
- [14] M. J. Black. Combining intensity and motion for incremental segmentation and tracking over long image sequences. In *Computer Vision—ECCV '92 (Proc. 2nd European Conference on Computer Vision)*, volume 588 of *Lecture Notes in Computer Science*, pages 485–493, Santa Margherita Ligure, Italy, May 1992. Springer-Verlag.
- [15] M. J. Black. *Robust Incremental Optical Flow*. PhD thesis, Yale University, USA, 1992.

- [16] M. J. Black and P. Anandan. A model for the detection of motion over time. In *Proc. 3rd International Conference on Computer Vision*, pages 33–37, Osaka, Japan, December 1990. Also Research Report YALEU/DCS/RR-822, Yale University, September 1990.
- [17] M. J. Black and P. Anandan. Robust dynamic motion estimation over time. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 296–302, Maui, HI, USA, June 1991.
- [18] M. J. Black and D. J. Fleet. Probabilistic detection and tracking of motion boundaries. *International Journal of Computer Vision*, 38(3):229–243, July 2000.
- [19] M. J. Black and A. D. Jepson. Estimating optical flow in segmented images using variable-order parametric models with local deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):972–986, October 1996.
- [20] A. Blake and M. Isard. *Active Contours*. Springer-Verlag, 1998.
- [21] A. Blake and A. Zisserman. *Visual Reconstruction*. The MIT Press Series in Artificial Intelligence. MIT Press, Cambridge, MA, 1987.
- [22] G. Borgefors. Distance transformations in digital images. *Computer Vision, Graphics and Image Processing*, 34(3):344–371, June 1986.
- [23] P. Bouthemy. A maximum likelihood framework for determining moving edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):499–511, May 1989.
- [24] P. Bouthemy and E. François. Motion segmentation and qualitative dynamic scene analysis from an image sequence. *International Journal of Computer Vision*, 10(2):157–182, April 1993.
- [25] Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 648–655, Santa Barbara, CA, USA, June 1998. Also Cornell CS technical report TR97-1658, December 1997.
- [26] Y. Y. Boykov and M. P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *Proc. 8th International Conference on Computer Vision*, volume 1, pages 105–112, Vancouver, Canada, July 2001.

- [27] N. Brady and N. O'Connor. Object detection and tracking using an EM-based motion estimation and segmentation framework. In *Proc. IEEE International Conference on Image Processing*, volume 1, pages 925–928, Lausanne, Switzerland, September 1996.
- [28] A. Broadhurst and R. Cipolla. The applications of uncalibrated occlusion junctions. In *Proc. 10th British Machine Vision Conference*, pages 245–254, Nottingham, September 1999.
- [29] P. J. Burt and E. H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, April 1983.
- [30] P.J. Burt, R. Hingorani, and R.J. Kolczynski. Mechanisms for isolating component patterns in the sequential analysis of multiple motion. In *IEEE Workshop on Visual Motion*, pages 187–193, Princeton, NJ, USA, October 1991.
- [31] B. F. Buxton, H. Buxton, D. W. Murray, and N. S. Williams. Machine perception of visual motion. *GEC Journal of Research*, 3(3):145–161, 1985.
- [32] H. Buxton and S. Gong. Visual surveillance in a dynamic and uncertain world. *Artificial Intelligence*, 78(1–2):431–459, October 1995.
- [33] J. F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, November 1986.
- [34] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, and J. Malik. Blobworld: A system for region-based image indexing and retrieval. In *Proc. 3rd International Conference on Visual Information and Information Systems*, pages 509–516, Amsterdam, The Netherlands, June 1999.
- [35] T. J. Cham and R. Cipolla. A statistical framework for long-range feature matching in uncalibrated image mosaicing. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 442–447, Santa Barbara, CA, USA, 1998.
- [36] R. Chellappa and A. Jain, editors. *Markov Random Fields: Theory and Application*. Academic Press, Boston, 1993.
- [37] R. Cipolla and A. Blake. Surface orientation and time to contact from image divergence and deformation. In *Computer Vision—ECCV '92 (Proc. 2nd European Conference on Computer Vision)*, volume 588 of *Lecture Notes in*

- Computer Science*, pages 187–202, Santa Margherita Ligure, Italy, May 1992. Springer-Verlag.
- [38] T. F. Cootes and C. J. Taylor. Active shape models - ‘smart snakes’. In *Proc. 3rd British Machine Vision Conference*, pages 266–275, Leeds, UK, September 1992. Springer-Verlag.
- [39] T. F. Cootes and C. J. Taylor. Statistical models of appearance for computer vision. Technical report, Wolfson Image Analysis Unit, Imaging Science and Biomedical Engineering, University of Manchester, UK, February 2001.
- [40] G. Csurka and P. Bouthemy. Direct identification of moving objects and background from 2D motion models. In *Proc. 7th International Conference on Computer Vision*, pages 566–571, Kerkyra, Greece, September 1999.
- [41] P. Dani and S. Chaudhuri. Automated assembling of images: Image montage preparation. *Pattern Recognition*, 28(3):431–445, March 1995.
- [42] T. J. Darrell and A. P. Pentland. Cooperative robust estimation using layers of support. Technical Report 163, MIT Media Lab Vision and Modeling Group, Cambridge, MA, USA, February 1991. (Revised March 1994).
- [43] A. P. Dempster, H. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society: Series B (Methodological)*, 39(1):1–38, January 1977.
- [44] T. Drummond and R. Cipolla. Application of Lie algebras to visual servoing. *International Journal of Computer Vision*, 37(1):21–41, June 2000.
- [45] T. Drummond and R. Cipolla. Real-time tracking of multiple articulated structures in multiple views. In *Computer Vision—ECCV 2000 (Proc. 6th European Conference on Computer Vision)*, volume 1843 of *Lecture Notes in Computer Science*, pages 20–36, Dublin, Ireland, June/July 2000. Springer-Verlag.
- [46] F. Dufaux, F. Moscheni, and A. Lippman. Spatio-temporal segmentation based on motion and static segmentation. In *Proc. International Conference on Image Processing*, volume 1, pages 306–309, Washington DC, USA, October 1995.
- [47] D. P. Elias. *The Motion-Based Segmentation of Image Sequences*. PhD thesis, University of Cambridge, UK, August 1998.

- [48] D. P. Elias and N. G. Kingsbury. The recovery of a near optimal layer representation for an entire image sequence. In *Proc. International Conference on Image Processing*, volume 1, pages 735–738, Santa Barbara, CA, USA, October 1997.
- [49] M. Etoh and Y. Shirai. Segmentation and 2D motion estimation by region fragments. In *Proc. 4th International Conference on Computer Vision*, pages 192–198, Berlin, May 1993.
- [50] O. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *Computer Vision—ECCV '92 (Proc. 2nd European Conference on Computer Vision)*, volume 588 of *Lecture Notes in Computer Science*, pages 563–578, Santa Margherita Ligure, Italy, May 1992. Springer-Verlag.
- [51] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, Cambridge, MA, 1993.
- [52] R. Fergus. Unsupervised model learning for recognition. MEng project report, Cambridge University Engineering Department, UK, June 2000.
- [53] J. Fernyhough, A. G. Cohn, and D. C. Hogg. Building qualitative event models automatically from visual input. In *Proc. 6th International Conference on Computer Vision*, pages 350–355, Bombay, India, January 1998.
- [54] M. A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.
- [55] M. Flickner, H. Sawhney, and W. Niblack. Query by image and video content: The QBIC system. *IEEE Computer Magazine*, 28(9):23–32, September 1995.
- [56] L. Gaucher and G. Medioni. Accurate motion flow estimation with discontinuities. In *Proc. 7th International Conference on Computer Vision*, volume 2, pages 695–702, Kerkyra, Greece, September 1999.
- [57] M. Gelgon and P. Bouthemy. Determining a structured spatio-temporal representation of video content for efficient visualisation and indexing. In *Computer Vision—ECCV '98 (Proc. 5th European Conference on Computer Vision)*, volume 1407 of *Lecture Notes in Computer Science*, pages 595–609, Freiburg, Germany, June 1998. Also Research Report 1157, IRISA Rennes, France, December 1997.



- [58] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis*. Chapman & Hall, London, 1995.
- [59] S. Geman and D. Geman. Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, November 1984.
- [60] P.R. Giaccone and G.A. Jones. Segmentation of global motion using temporal probabilistic classification. In *Proc. 9th British Machine Vision Conference*, volume 2, pages 619–628, Southampton, September 1998.
- [61] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, editors. *Markov Chain Monte Carlo in Practice*. Chapman & Hall, London, 1996.
- [62] C. G. Harris and M. Stephens. A combined corner and edge detector. In *Proc. 4th Alvey Vision Conference*, pages 147–151, Manchester, UK, August 1988.
- [63] R. Hartley. Estimation of relative camera positions for uncalibrated cameras. In *Computer Vision—ECCV '92 (Proc. 2nd European Conference on Computer Vision)*, volume 588 of *Lecture Notes in Computer Science*, pages 579–587, Santa Margherita Ligure, Italy, May 1992. Springer-Verlag.
- [64] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
- [65] B. K. P. Horn. *Robot Vision*. MIT Press, Cambridge, MA, 1986.
- [66] B. K. P. Horn and B. G. Schunk. Determining optical flow. *Artificial Intelligence*, 17(1–3):185–203, August 1981.
- [67] S. Hsu, P. Anandan, and S. Peleg. Accurate computation of optical flow by using layered motion representations. In *Proc. International Conference on Pattern Recognition*, pages 743–746, Jerusalem, Israel, October 1994.
- [68] Y. Huang, D. Paulus, and H. Niemann. Background-foreground segmentation based on dominant motion estimation and static segmentation. In *Proc. International Workshop on Signal, Image Analysis and Processing*, pages 13–15, Pula, Croatia, June 2000.
- [69] P. J. Huber. Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35(1):73–101, 1964.

- [70] P. J. Huber. *Robust Statistics*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, New York, 1981.
- [71] S. Intille and A. Bobick. Representation and visual recognition of complex multi-agent actions using belief networks. In *Proc. CVPR'98 Workshop on the Interpretation of Visual Motion*, June 1998. Also at ECCV'98 Workshop on the Perception of Human Action, and Technical Report 454, MIT Media Lab, Cambridge, MA, USA.
- [72] S. S. Intille and A. F. Bobick. Closed-world tracking. In *Proc. 5th International Conference on Computer Vision*, pages 672–678, Cambridge, MA, USA, June 1995.
- [73] M. Irani and P. Anandan. Video indexing based on mosaic representations. *Proceedings of the IEEE*, 86(5):905–921, May 1998.
- [74] M. Irani and P. Anandan. All about direct methods. In *Vision Algorithms: Theory and Practice (Proc. International Workshop Vision Algorithms '99)*, volume 1883 of *Lecture Notes in Computer Science*, pages 267–277, Kerkyra, Greece, September 1999. Springer-Verlag.
- [75] M. Irani, P. Anandan, J. Bergen, R. Kumar, and S. Hsu. Efficient representations of video sequences and their representations. *Signal Processing: Image Communication*, 8(4):327–351, May 1996.
- [76] M. Irani, P. Anandan, and S. Hsu. Mosaic based representations of video sequences and their applications. In *Proc. 5th International Conference on Computer Vision*, pages 605–611, Cambridge, MA, USA, June 1995.
- [77] M. Irani and S. Peleg. Motion analysis for image enhancement: Resolution, occlusion, and transparency. *Journal of Visual Communication and Image Representation*, 4(4):324–335, December 1993.
- [78] M. Irani, B. Rousso, and S. Peleg. Computing occluding and transparent motions. *International Journal of Computer Vision*, 12(1):5–16, January 1994.
- [79] H. Ishikawa and I. H. Jermyn. Region extraction from multiple images. In *Proc. 8th International Conference on Computer Vision*, volume 1, pages 509–516, Vancouver, Canada, July 2001.
- [80] *Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/S*. ISO/IEC 11172, 1993–1995. MPEG-1 standard.

- [81] *Information technology – Generic coding of moving pictures and associated audio information*. ISO/IEC 13818, 1997–2000. MPEG-2 standard.
- [82] *Information technology – Coding of audio-visual objects*. ISO/IEC 14496, 1999–2001. MPEG-4 standard (draft).
- [83] J. Jain and A. Jain. Displacement measurement and its application in inter-frame image coding. *IEEE Transactions on Communications*, 29(12):1799–1804, December 1981.
- [84] E. T. Jaynes. Probability theory as extended logic. Unpublished book. Available at <http://bayes.wustl.edu/> (last accessed 18 July 2001), 1998.
- [85] A. Jepson and M. Black. Mixture models for optical flow computation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 760–761, New York, NY, June 1993. Also Technical Report RBCV-TR-93-44, Department of Computer Science, University of Toronto, April 1993.
- [86] S. X. Ju, M. J. Black, S. Minneman, and D. Kimber. Summarization of video-taped presentations: Automatic analysis of motion and gesture. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):686–696, September 1998.
- [87] S. B. Kang. A survey of image-based rendering techniques. Technical Report CRL 97/4, Digital Equipment Corporation Cambridge Research Lab, MA, USA, August 1997.
- [88] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, May 1983.
- [89] K. Koffka. *Principles of Gestalt Psychology*. Harcourt, Brace & World, New York, 1935.
- [90] A. C. Kokaram. *Motion Picture Restoration*. PhD thesis, University of Cambridge, UK, May 1993.
- [91] J. MacCormick and A. Blake. Spatial dependence in the observation of visual contours. In *Computer Vision—ECCV ’98 (Proc. 5th European Conference on Computer Vision)*, volume 1407 of *Lecture Notes in Computer Science*, pages 765–781, Freiburg, Germany, June 1998.

- [92] J. Malik. On binocularly viewed occlusion junctions. In *Computer Vision—ECCV '96 (Proc. 4th European Conference on Computer Vision)*, volume 1064 of *Lecture Notes in Computer Science*, pages 167–174, Cambridge, UK, April 1996.
- [93] D. Marr and E. C. Hildreth. Theory of edge detection. In *Proc. Royal Society of London*, volume 207, pages 187–217, 1980.
- [94] D. Marr and S. Ullman. Directional selectivity and its use in early visual processing. *Proc. Royal Society London Series B*, 211:151–180, 1981.
- [95] D. Martin, C. Fowlkes, D. Tai, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th International Conference on Computer Vision*, volume 2, pages 416–423, Vancouver, Canada, July 2001.
- [96] S. J. Maybank and O. D. Faugeras. A theory of self calibration of a moving camera. *International Journal of Computer Vision*, 8(2):123–151, August 1992.
- [97] T. Meier and K. N. Ngan. Automatic segmentation of moving objects for video object plane generation. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):525–538, September 1998.
- [98] T. J. Mills, D. Pye, N. J. Hollinghurst, and K. R. Wood. AT&TV: Broadcast television and radio retrieval. In *Proc. RIAO 2000 (Recherche d'Informations Assistée Par Ordinateur, Content-Based Multimedia Information Access)*, volume 2, pages 1135–1144, Paris, France, April 2000.
- [99] J.L. Mitchell, W.B. Pennebaker, C.E. Fogg, and D.J. LeGall. *MPEG Video Compression Standard*. Digital Multimedia Standards. Chapman & Hall, New York, 1997.
- [100] R. D. Morris. *Image Sequence Restoration Using Gibbs Distributions*. PhD thesis, University of Cambridge, UK, May 1995.
- [101] F. Moscheni and S. Bhattacharjee. Robust region merging for spatio-temporal segmentation. In *Proc. International Conference on Image Processing*, volume 1, pages 501–504, Lausanne, Switzerland, September 1996.
- [102] F. Moscheni, S. Bhattacharjee, and M. Kunt. Spatiotemporal segmentation based on region merging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(9):897–915, September 1998.

- [103] F. Moscheni and F. Dufaux. Region merging based on robust statistical testing. In *Proc. SPIE Visual Communications and Image Processing*, volume 2727, Orlando, Florida, USA, March 1996.
- [104] K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proc. 15th Conference on Uncertainty in Artificial Intelligence*, pages 467–475, Stockholm, Sweden, July/August 1999. Morgan Kaufmann.
- [105] D. W. Murray and B. F. Buxton. Scene segmentation from visual motion using global optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(2):220–228, March 1987.
- [106] H. H Nagel. On the estimation of optic flow: Relations between different approaches and some new results. *Artificial Intelligence*, 33(3):299–324, November 1987.
- [107] J. M. Odobez and P. Bouthemy. Separation of moving regions from background in an image sequence acquired with a mobile camera. In *Video Data Compression for Multimedia Computing*, pages 283–311. Kluwer Academic Publishers, 1997.
- [108] N. Oliver, B. Rosario, and A. Pentland. A Bayesian computer vision system for modeling human interactions. In *Proc. International Conference on Computer Vision Systems*, pages 254–272, Las Palmas, Gran Canaria, Spain, January 1999.
- [109] L. Paletta, D. Sinclair, and A. Pinz. Classification of edges under motion. In *Proc. 10th Scandinavian Conference on Image Analysis*, pages 613–620, Lappeenranta, Finland, June 1997.
- [110] I. Patras, E. A. Hendriks, and R. L. Legendijk. Video segmentation by MAP labeling of watershed segments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):326–332, March 2001.
- [111] M. Pollefeys, R. Koch, and L. Van Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. *International Journal of Computer Vision*, 32(1):7–25, August 1999.
- [112] W. Press, B Flannery, S Teukolsky, and W Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1988.

- [113] P. Pritchett and A. Zisserman. Wide baseline stereo matching. In *Proc. 6th International Conference on Computer Vision*, pages 754–760, Bombay, India, January 1998.
- [114] P. Remagnino, T. Tan, and K. Baker. Agent orientated annotation in model based visual surveillance. In *Proc. 6th International Conference on Computer Vision*, pages 857–862, Bombay, India, January 1998.
- [115] W. J. J. Rey. *Introduction to Robust and Quasi-Robust Statistical Methods*, volume 690 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1978.
- [116] J. Rissanen. Minimum description length principle. In S. Kotz and N. L. Johnson, editors, *Encyclopedia of Statistical Sciences*, volume 5, pages 523–527. John Wiley & Sons, New York, 1985.
- [117] K. Rodden, W. Basalaj, D. Sinclair, and K. Wood. Evaluating a visualisation of image similarity as a tool for image browsing. In *IEEE Symposium on Information Visualisation (InfoVis '99)*, pages 36–43, San Francisco, CA, USA, October 1999.
- [118] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, New York, 1987.
- [119] D. H. Sattinger and O. L. Weaver. *Lie Groups and Algebras with Applications to Physics, Geometry, and Mechanics*, volume 61 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 1986.
- [120] D. Saur, Y.-P. Tan, S. R. Kulkarni, and P. J. Ramadge. Automated analysis and annotation of basketball video. In *Proc. SPIE Storage and Retrieval for Image and Video Databases*, volume 3022, pages 176–187, San Jose, CA, USA, February 1997.
- [121] H. S. Sawhney and S. Ayer. Compact representations of videos through dominant and multiple motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):814–830, August 1996.
- [122] H. S. Sawhney, S. Ayer, and M. Gorkani. Model-based 2D & 3D dominant motion estimation for mosaicing and video representation. In *Proc. 5th International Conference on Computer Vision*, pages 583–590, Cambridge, MA, USA, June 1995.

- [123] C. Schmid, R. Mohr, and C. Bauckhage. Comparing and evaluating interest points. In *Proc. 6th International Conference on Computer Vision*, pages 230–235, Bombay, India, January 1998.
- [124] J. G. Semple and G. T. Kneebone. *Algebraic Projective Geometry*. Oxford Classic Texts in the Physical Sciences. Oxford University Press, 1952.
- [125] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423; 623–656, July and October 1948.
- [126] J. Shi and J. Malik. Normalized cuts and image segmentation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 731–737, San Juan, Puerto Rico, June 1997.
- [127] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *Proc. 6th International Conference on Computer Vision*, pages 1154–1160, Bombay, India, January 1998.
- [128] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, August 2000.
- [129] T. Sikora. MPEG-4 video standard verification model. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(1):19–31, February 1991.
- [130] D. Sinclair. Voronoi seeded colour image segmentation. Technical Report 1999.3, AT&T Laboratories Cambridge, UK, 1999.
- [131] P. Smith, T. Drummond, and R. Cipolla. Edge tracking for motion segmentation and depth ordering. In *Proc. 10th British Machine Vision Conference*, pages 584–593, Nottingham, UK, September 1999.
- [132] P. Smith, T. Drummond, and R. Cipolla. Motion segmentation by tracking edge information over multiple frames. In *Computer Vision—ECCV 2000 (Proc. 6th European Conference on Computer Vision)*, volume 1843 of *Lecture Notes in Computer Science*, pages 396–410, Dublin, Ireland, June/July 2000. Springer-Verlag.
- [133] P. Smith, T. Drummond, and R. Cipolla. Segmentation of multiple motions by edge tracking between two frames. In *Proc. 11th British Machine Vision Conference*, pages 342–315, Bristol, UK, September 2000.

- [134] P. Smith, D. Sinclair, R. Cipolla, and K. Wood. Effective corner matching. In *Proc. 9th British Machine Vision Conference*, pages 545–556, Southampton, UK, September 1998.
- [135] S. M. Smith. Reviews of optic flow, motion segmentation, edge finding and corner finding. Technical Report TR97SMS1, Department of Clinical Neurology, Oxford University, UK, 1997.
- [136] S. M. Smith and J. M. Brady. ASSET-2: Real-time motion segmentation and shape tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):814–820, August 1995. Also Technical Report TR95SMS2b, Department of Clinical Neurology, Oxford University, UK, 1995.
- [137] I. Sobel. An isotropic  $3 \times 3$  image gradient operator. In H. Freeman, editor, *Machine Vision for Three-Dimensional Scenes*, pages 376–379. Academic Press, 1990.
- [138] R. Szeliski. Image mosaicing for tele-reality applications. In *Proc. IEEE Workshop on Applications of Computer Vision (WACV'94)*, pages 44–53, Sarasota, FL, USA, December 1994. Also Technical Report CRL 94/2, Digital Equipment Corporation Cambridge Research Lab, May 1994.
- [139] H. Tao and H. S. Sawhney. Global matching criterion and color segmentation based stereo. In *Proc. IEEE Workshop on Applications of Computer Vision (WACV2000)*, pages 246–253, Palm Springs, CA, USA, December 2000.
- [140] C. W. Therrien. *Decision Estimation and Classification*. John Wiley & Sons, New York, 1989.
- [141] W. B. Thompson. Combining motion and contrast for segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(6):543–549, November 1980.
- [142] A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-Posed Problems*. Scripta Series in Mathematics. Winston, New York, 1977.
- [143] P. H. S. Torr. *Motion Segmentation and Outlier Detection*. PhD thesis, University of Oxford, UK, December 1995.
- [144] P. H. S. Torr. An assessment of information criteria for motion model selection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 47–53, San Juan, Puerto Rico, June 1997.



- [145] P. H. S. Torr and D. W. Murray. Outlier detection and motion segmentation. In *Proc. SPIE Sensor Fusion VI*, volume 2059, pages 432–443, Boston, MA, USA, September 1993.
- [146] P. H. S. Torr and D. W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *International Journal of Computer Vision*, 3(24):271–300, September 1997.
- [147] P. H. S. Torr, R. Szeliski, and P. Anandan. An integrated Bayesian approach to layer extraction from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):297–303, March 2001.
- [148] P. H. S. Torr and A. Zisserman. Feature based methods for structure and motion estimation. In *Vision Algorithms: Theory and Practice (Proc. International Workshop Vision Algorithms '99)*, volume 1883 of *Lecture Notes in Computer Science*, pages 278–295, Kerkyra, Greece, September 1999. Springer-Verlag.
- [149] P. H. S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156, April 2000.
- [150] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: Principles and practice of background maintenance. In *Proc. 7th International Conference on Computer Vision*, volume 1, pages 255–261, Kerkyra, Greece, September 1999.
- [151] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzibbon. Bundle adjustment – a modern synthesis. In *Vision Algorithms: Theory and Practice (Proc. International Workshop Vision Algorithms '99)*, volume 1883 of *Lecture Notes in Computer Science*, pages 298–375, Kerkyra, Greece, September 1999. Springer-Verlag.
- [152] B. Triggs, A. Zisserman, and R. Szeliski, editors. *Vision Algorithms: Theory and Practice (Proc. International Workshop Vision Algorithms '99)*, volume 1883 of *Lecture Notes in Computer Science*, Kerkyra, Greece, September 1999. Springer-Verlag.
- [153] D. Tweed and A. Calway. Integrated segmentation and depth ordering of motion layers in image sequences. In *Proc. 11th British Machine Vision Conference*, pages 322–331, Bristol, UK, September 2000.

- [154] N. Ueda and R. Nakano. Deterministic annealing EM algorithm. *Neural Networks*, 11(2):271–282, April 1998.
- [155] N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton. SMEM algorithm for mixture models. *Neural Computation*, 12(9):2109–2128, September 2000.
- [156] V. S. Varadarajan. *Lie Groups, Lie Algebras, and their Representations*, volume 102 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1984.
- [157] L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–589, June 1991.
- [158] J. Y. A. Wang and E. H. Adelson. Layered representation for motion analysis. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 361–366, New York, NY, USA, June 1993.
- [159] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing*, 3(5):625–638, September 1994.
- [160] Y. Weiss and E. H. Adelson. A unified mixture framework for motion segmentation: incorporating spatial coherence and estimating the number of models. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 321–326, San Francisco, CA, USA, June 1996.
- [161] K. D. Youw, B. Yeo, M. M. Yeung, and B. Liu. Analysis and presentation of soccer highlights from digital video. In *Proc. 2nd Asian Conference on Computer Vision*, pages 499–503, Singapore, December 1995.
- [162] Z. Zhang. Parameter estimation techniques: A tutorial with application to conic fitting. *Image and Vision Computing*, 15(1):59–76, January 1997. Also INRIA Research Report No. 2676, October 1995.
- [163] Z. Zhang, R. Deriche, O. Faugeras, and Q. T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78(1–2):87–119, October 1995. Also INRIA Research Report No. 2273, May 1994.
- [164] I. Zoghlami, O. Faugeras, and R. Deriche. Using geometric corners to build a 2D mosaic from a set of images. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 420–425, San Juan, Puerto Rico, June 1997.

---

# Author Index

---

- Adiv [1985], 20, 64  
Anandan [1989], 15  
André et al. [1994], 4  
Ayer and Sawhney [1995], 25, 100, 112–114, 156  
Ayer et al. [1994], 14, 24, 30, 32, 68, 70  
Baker et al. [1998], 23  
Ballard and Brown [1982], 20  
Barron et al. [1994], 11  
Bergen and Meyer [1998], 29–32, 82, 116, 117  
Bergen et al. [1990], 21  
Bergen et al. [1992], 14, 15, 21, 62, 64, 109, 153, 176  
Bishop [1995], 71  
Black and Anandan [1990], 21  
Black and Anandan [1991], 13, 21  
Black and Fleet [2000], 31  
Black and Jepson [1996], 22  
Black [1992], 20, 26, 27, 86  
Blake and Isard [1998], 147  
Blake and Zisserman [1987], 152  
Borgefors [1986], 83  
Bouthemy and François [1993], 26, 27  
Bouthemy [1989], 57, 58  
Boykov and Jolly [2001], 28  
Boykov et al. [1998], 27  
Brady and O'Connor [1996], 25, 29, 31, 82, 156, 161  
Broadhurst and Cipolla [1999], 32  
Burt and Adelson [1983], 15  
Burt et al. [1991], 24, 62  
Buxton and Gong [1995], 4  
Buxton et al. [1985], 59  
Canny [1986], 16, 37, 57  
Carson et al. [1999], 4  
Cham and Cipolla [1998], 18, 19  
Chellappa and Jain [1993], 21, 26, 86  
Cipolla and Blake [1992], 62  
Cootes and Taylor [1992], 175  
Cootes and Taylor [2001], 175  
Csurka and Bouthemy [1999], 24, 26, 27, 68, 86  
Dani and Chaudhuri [1995], 147  
Darrell and Pentland [1991], 22, 23  
Dempster et al. [1977], 6, 25, 50, 71, 73, 152, 187, 191  
Drummond and Cipolla [2000], 64, 65, 175

- Dufaux et al. [1995], 30, 114, 115  
 Elias and Kingsbury [1997], 25, 156  
 Elias [1998], 25, 32, 100, 102, 113, 114  
 Etoh and Shirai [1993], 21  
 Faugeras [1992], 18  
 Faugeras [1993], 15, 16, 62  
 Fergus [2000], 153, 155, 157  
 Fernyhough et al. [1998], 4  
 Fischler and Bolles [1981], 18, 22, 70  
 Flickner et al. [1995], 4  
 Gaucher and Medioni [1999], 31  
 Gelgon and Bouthemy [1998], 4, 30, 31, 147  
 Gelman et al. [1995], 48, 51, 74  
 Geman and Geman [1984], 21, 26, 52, 86, 89, 90  
 Giaccone and Jones [1998], 25, 32  
 Gilks et al. [1996], 32, 76, 175, 195  
 Harris and Stephens [1988], 16, 17  
 Hartley and Zisserman [2000], 15, 16, 62  
 Hartley [1992], 18  
 Horn and Schunk [1981], 12, 14  
 Horn [1986], 12  
 Hsu et al. [1994], 22  
 Huang et al. [2000], 24, 30, 70, 82  
 Huber [1964], 181  
 Huber [1981], 24, 67, 153, 181, 184  
 ISO [1993–1995], 3, 12  
 ISO [1997–2000], 3, 12  
 ISO [1999–2001], 3, 147, 175  
 Intille and Bobick [1995], 4  
 Intille and Bobick [1998], 4  
 Irani and Anandan [1998], 4, 15, 147  
 Irani and Anandan [1999], 12, 15  
 Irani and Peleg [1993], 3, 5  
 Irani et al. [1994], 14, 24, 25, 31, 32, 62, 68, 109, 153, 176  
 Irani et al. [1995], 15  
 Irani et al. [1996], 3, 15, 68, 147  
 Ishikawa and Jermyn [2001], 28  
 Jain and Jain [1981], 12  
 Jaynes [1998], 48, 51, 74  
 Jepson and Black [1993], 21, 71  
 Ju et al. [1998], 4  
 Kang [1997], 147  
 Kirkpatrick et al. [1983], 52, 90  
 Koffka [1935], 20  
 Kokaram [1993], 5  
 MacCormick and Blake [1998], 96, 195  
 Malik [1996], 32, 45  
 Marr and Hildreth [1980], 57  
 Marr and Ullman [1981], 13, 59  
 Martin et al. [2001], 20, 96  
 Maybank and Faugeras [1992], 18  
 Meier and Ngan [1998], 24, 28  
 Mills et al. [2000], 4, 96, 197  
 Mitchell et al. [1997], 3, 12  
 Morris [1995], 5, 26  
 Moscheni and Bhattacharjee [1996], 30, 116, 117  
 Moscheni and Dufaux [1996], 30, 96, 100, 114, 115  
 Moscheni et al. [1998], 30, 82  
 Murphy et al. [1999], 92  
 Murray and Buxton [1987], 20, 26  
 Nagel [1987], 14  
 Odobez and Bouthemy [1997], 14, 24–27, 32, 62, 68, 70, 86  
 Oliver et al. [1999], 5  
 Paletta et al. [1997], 32  
 Patras et al. [2001], 29, 31, 82, 102  
 Pollefeys et al. [1999], 18

- Press et al. [1988], 67, 181  
Pritchett and Zisserman [1998], 17, 18  
Remagnino et al. [1998], 5  
Rey [1978], 181, 184  
Rissanen [1985], 23, 25, 31, 151, 156, 165  
Rodden et al. [1999], 82  
Rousseeuw and Leroy [1987], 18, 24, 70  
Sattinger and Weaver [1986], 64, 65  
Saur et al. [1997], 4  
Sawhney and Ayer [1996], 3, 15, 24, 25, 70, 71, 147  
Sawhney et al. [1995], 24, 25, 62  
Schmid et al. [1998], 17  
Semple and Kneebone [1952], 62  
Shannon [1948], 156  
Shi and Malik [1997], 27  
Shi and Malik [1998], 28  
Shi and Malik [2000], 27  
Sikora [1991], 3, 147, 175  
Sinclair [1999], 6, 40, 42, 58, 82, 83  
Smith and Brady [1995], 22  
Smith et al. [1998], 16–18  
Smith [1997], 11, 17  
Sobel [1990], 37, 57, 61  
Szeliski [1994], 15  
Tao and Sawhney [2000], 29  
Therrien [1989], 23  
Thompson [1980], 30, 82  
Tikhonov and Arsenin [1977], 185  
Torr and Murray [1993], 70  
Torr and Murray [1997], 18, 24, 70  
Torr and Zisserman [1999], 12  
Torr and Zisserman [2000], 19  
Torr et al. [2001], 23  
Torr [1995], 22  
Torr [1997], 156  
Toyama et al. [1999], 147  
Triggs et al. [1999], 12, 19  
Tweed and Calway [2000], 30, 31, 82, 96  
Ueda and Nakano [1998], 152  
Ueda et al. [2000], 153  
Varadarajan [1984], 64, 65  
Vincent and Soille [1991], 29, 82  
Wang and Adelson [1993], 22, 23, 31, 62  
Wang and Adelson [1994], 19, 23, 111, 112  
Weiss and Adelson [1996], 25–27, 31, 86, 112, 114  
Youw et al. [1995], 4  
Zhang et al. [1995], 18, 70  
Zhang [1997], 184  
Zoghلامي et al. [1997], 18