

DEVELOPMENT OF A MOTION DISTILLATION  
PARADIGM FOR VISUAL SURVEILLANCE

BY

MARK SUGRUE

A THESIS SUBMITTED FOR THE DEGREE OF  
DOCTOR OF PHILOSOPHY IN THE UNIVERSITY OF LONDON

ROYAL HOLLOWAY,  
UNIVERSITY OF LONDON

2007

1



260 360629 X

*I* hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signed

Mark Sugrue



## Acknowledgements

I am greatly indebted to my colleagues and friends at Royal Holloway, particularly my supervisor, Prof. Roy Davies. Roy has been exceptionally generous with his time and has provided considerable scientific input and encouragement. Thanks also to my friends and colleagues in the Machine Vision and Signal Processing group; Matt Butler, Dan Ellin, Dr. Xiaopeng Hu, Dr. Tieying Lu, Dr. Hamadi Nait-Charif, and Dr. Stuart Flockton; who shared their knowledge and experience and ensured a productive and enjoyable environment. Specific thanks go also to Hamadi for helping prepare Video 3 and for Figure 4.8.

I am also immeasurably grateful to my wife, Valeria, who not only put up with me these past years, but created the pedestrian animation used in Chapter 5 and helped me write the following 'poetic abstract'...

---

### The Machine Vision Poem

---

My research in Machine Vision,  
improves motion recognition.  
Unlike methods in tradition,  
it tracks objects in transition,  
no regard for their appearance,  
or their modelling adherence.  
No subtraction of the background,  
is needed to be found.

By studying the Human Brain,  
we hope to make massive gain.  
For humans, when they're seeing,  
take *The Motion* from *The Being*,  
and the pattern of the motion,  
may provoke a smile... or caution!

In our system we use filters,  
3D Haar and Gaussian,  
applied to a spatio-temporal stack  
of video frames in real-time...  
...but unfortunately, that doesn't rhyme!

---

## ABSTRACT

THE HUGE NUMBER OF CCTV CAMERAS AND SECURITY APPLICATIONS PLACES INCREASING REQUIREMENTS ON AUTOMATIC VISUAL TRACKING AND BEHAVIOUR CLASSIFICATION SYSTEMS. THE BEST WORKING EXAMPLE OF SUCH A TRACKER IS THE HUMAN VISUAL SYSTEM (HVS) WHICH CAN FLAWLESSLY DETECT, TRACK AND UNDERSTAND ALMOST ANY OBJECT OR EVENT.

THE RESEARCH DESCRIBED IN THIS THESIS USES LESSONS LEARNT FROM STUDIES OF THE HVS TO DEVELOP A NOVEL APPROACH FOR COMPUTER-BASED VISUAL TRACKING. IN THIS APPROACH, INITIAL DETECTION OF MOVING OBJECTS IS ACHIEVED USING A NEW MOTION DISTILLATION PARADIGM WHICH EMPLOYS SPATIO-TEMPORAL WAVELET DECOMPOSITION OF VIDEO. THE METHOD IS SHOWN TO BE MORE ROBUST THAN TRADITIONAL BACKGROUND MODELLING TECHNIQUES WHILE BEING COMPUTATIONALLY LESS EXPENSIVE.

AS WITH THE HVS, THE APPROACH USES A DUAL-CHANNEL TRACKING ARCHITECTURE TO PERFORM TRACKING. THE MOTION CHANNEL, GENERATED THROUGH MOTION DISTILLATION, HANDLES OBJECT DETECTION AND INITIALISES TRACKING. THE FORM CHANNEL IS USED TO RESOLVE TRACKING AMBIGUITIES AND OCCLUSIONS. QUALITATIVE AND QUANTITATIVE TRACKING RESULTS ILLUSTRATE THE ADVANTAGES OF THIS APPROACH.

THIS THESIS ALSO DESCRIBES A NEW APPROACH TO THE TASK OF OB-

JECT (E.G. HUMAN) BEHAVIOUR ANALYSIS - A SUBJECT WHICH IS OF GREAT IMPORTANCE, YET WHICH IS STILL AN UNDER-RESEARCHED ASPECT OF VISUAL TRACKING. IN THE WORK DESCRIBED HERE, OBJECTS ARE CATEGORISED INTO VEHICLES, PEDESTRIANS, RUNNERS, GROUPS AND UNKNOWN PEDESTRIAN BEHAVIOUR.

# Contents

Abstract	4
<b>1 Introduction</b>	<b>18</b>
1.1 Human Visual System . . . . .	19
1.2 Aims and assumptions . . . . .	23
1.3 Thesis structure . . . . .	26
<b>2 Literature review</b>	<b>29</b>
2.1 $2D+1$ – Foreground techniques . . . . .	31
2.1.1 Optical flow . . . . .	31
2.1.2 Feature tracking . . . . .	32
2.1.3 Location prediction . . . . .	33
2.2 Appearance models . . . . .	35
2.2.1 Illumination . . . . .	37

2.3	<i>1D+2</i> – Background modeling . . . . .	38
2.3.1	Image differencing . . . . .	38
2.3.2	Statistical filtering . . . . .	40
2.3.3	Other methods . . . . .	42
2.3.4	Maintenance . . . . .	43
2.3.5	Thresholding . . . . .	45
2.3.6	Tracking after detection . . . . .	46
2.4	<i>3D</i> – Spatio-temporal techniques . . . . .	48
2.5	Behaviour analysis . . . . .	49
2.6	Human Visual System . . . . .	52
2.7	Summary . . . . .	55
<b>3</b>	<b>Motion Detection</b>	<b>58</b>
3.1	Filtering for noise reduction . . . . .	59
3.2	Background modelling . . . . .	61
3.2.1	Median filter . . . . .	64
3.2.2	Thresholding and video noise . . . . .	65
3.2.3	Gaussian Mixture Model . . . . .	67
3.2.4	Criticism of background modelling . . . . .	68

3.3	Motion Distillation . . . . .	69
3.3.1	Temporal edge detection . . . . .	69
3.3.2	Wavelet decomposition . . . . .	72
3.3.3	The spatio-temporal Haar wavelet . . . . .	74
3.3.4	Detection comparison . . . . .	77
3.3.5	Computational cost . . . . .	84
3.3.6	Edge zoom . . . . .	86
3.3.7	Shake protection . . . . .	88
3.3.8	Comparison to the Linköping s-t method . . . . .	90
3.3.9	Aperture problem . . . . .	91
3.4	General spatio-temporal wavelets . . . . .	95
3.4.1	Difference of Offset Gaussians . . . . .	96
3.4.2	General cuboid filters . . . . .	99
3.5	Summary . . . . .	108
<b>4</b>	<b>Dual-Channel Tracking</b>	<b>112</b>
4.1	Prediction . . . . .	115
4.1.1	Kalman filter . . . . .	115
4.1.2	Particle filter . . . . .	118

4.1.3	General problems . . . . .	119
4.2	Appearance model . . . . .	122
4.2.1	Template matching . . . . .	125
4.2.2	Histogram matching . . . . .	125
4.2.3	AM comparison test . . . . .	127
4.3	Dual-channel approach . . . . .	130
4.3.1	Bayesian Networks . . . . .	135
4.4	Results . . . . .	137
4.4.1	Video 1 . . . . .	138
4.4.2	Video 2 . . . . .	138
4.4.3	Video 3 . . . . .	141
4.4.4	Other videos . . . . .	142
4.5	Summary . . . . .	144
<b>5</b>	<b>Behaviour Analysis</b>	<b>147</b>
5.1	Biological models . . . . .	149
5.2	Recognition approaches . . . . .	150
5.2.1	Makris and Ellis . . . . .	150
5.2.2	Dee and Hogg . . . . .	152

5.2.3	Bobick and Davis . . . . .	154
5.2.4	Stauffer and Grimson . . . . .	155
5.3	Motion signal analysis . . . . .	156
5.3.1	View independence . . . . .	162
5.3.2	Filter dependence . . . . .	165
5.3.3	Gait . . . . .	168
5.3.4	Shape and Texture . . . . .	170
5.4	Behaviour classification . . . . .	171
5.5	Summary . . . . .	182
<b>6</b>	<b>Discussion</b>	<b>185</b>
6.1	Motion Distillation . . . . .	185
6.2	Tracking . . . . .	187
6.3	Behaviour . . . . .	188
6.4	Context . . . . .	189
6.5	Theory and synthesis . . . . .	191
6.6	Suggestions for future research . . . . .	194
<b>7</b>	<b>Conclusions</b>	<b>197</b>



<b>A Video data</b>	<b>199</b>
A.1 List of videos . . . . .	199
A.2 List of video sources . . . . .	205
<b>B Object labelling</b>	<b>207</b>
<b>C Glossary of acronyms</b>	<b>210</b>
<b>D Author's Publications</b>	<b>212</b>
<b>Bibliography</b>	<b>214</b>

## List of Figures

1.1	A cartoonist's view of walking and running . . . . .	22
1.2	A cartoonist adds emotion to his characters . . . . .	22
1.3	CCTV example frames . . . . .	25
2.1	Visual surveillance literature . . . . .	31
3.1	Background model tracking scheme . . . . .	62
3.2	Frame stacks: 1D pixel statistics . . . . .	70
3.3	The $s-t$ video cube . . . . .	70
3.4	Wavelet decomposition example . . . . .	73
3.5	Sobel temporal edge detection . . . . .	75
3.6	Multidimensional Haar wavelets . . . . .	76
3.7	Motion detection results . . . . .	78
3.8	Graph of MD costs . . . . .	86

3.9	Edge zoom example . . . . .	88
3.10	Video shake detection . . . . .	90
3.11	Aperture problem . . . . .	94
3.12	Graph of DoOG and reversed Gaussian plot . . . . .	98
3.13	DoOG for optic flow calculations . . . . .	100
3.14	DoOG optic flow example . . . . .	100
3.15	Diagram of general cuboid filter in 2D . . . . .	102
3.16	Graph of asymmetrical filter outputs . . . . .	105
3.17	Graph of filter pair ratios . . . . .	106
3.18	Diagram of general cuboid filter in 3D . . . . .	107
3.19	Graph of a filter pair ratio in $\gamma$ - $\beta$ space . . . . .	108
3.20	Example of contrast independent motion detection . . . . .	109
4.1	Foreground tracking scheme . . . . .	114
4.2	Diagram of hidden state vectors . . . . .	114
4.3	Kalman prediction scheme . . . . .	115
4.4	Graph illustrating the inertial behaviour of the Kalman filter .	117
4.5	A pixel process modelled using a Kalman filter . . . . .	118
4.6	Scheme of a prediction cycle of the particle filter . . . . .	120

4.7	Example of location prediction failure . . . . .	123
4.8	Example of face tracking using a particle filter . . . . .	123
4.9	Example of failure of Kalman prediction . . . . .	124
4.10	Graph of AM matching results for a vehicle . . . . .	128
4.11	Graph of AM matching results for a pedestrian . . . . .	128
4.12	Graph of AM sensitivity to poor segmentation . . . . .	129
4.13	Diagram of dual-channel algorithm flow . . . . .	133
4.14	Diagram of blob sorting . . . . .	134
4.15	Tracking results: Video 1 . . . . .	138
4.16	Tracking results: Video 2 . . . . .	140
4.17	Tracking results: Video 3 . . . . .	143
4.18	Tracking results: Other videos . . . . .	145
5.1	Point light displays . . . . .	150
5.2	Object track modeling . . . . .	151
5.3	Goal directed behaviour modelling . . . . .	153
5.4	Motion Energy and Motion History images . . . . .	155
5.5	Categorisation based on shape . . . . .	156
5.6	Examples of Motion Distillation images for traffic scenes . . . . .	161

5.7	Graph of motion signals for a car . . . . .	161
5.8	Graph of motion signals for a pedestrian . . . . .	162
5.9	Diagram of views of animated pedestrian . . . . .	163
5.10	Graphs of motion signal with viewing angle . . . . .	165
5.11	Graphs of phase change in motion signal . . . . .	166
5.12	Graphs of motion signal from above animated pedestrian . . .	166
5.13	Graphs of motion signal from DoOG and Haar wavelets . . . .	167
5.14	Graphs of motion signals and gait . . . . .	169
5.15	Textured and untextured test images . . . . .	170
5.16	Pedestrian model . . . . .	174
5.17	Graph of $\eta$ vs. box width for walkers and runners . . . . .	175
5.18	Graphs of motion signals for behaviours: walking . . . . .	176
5.19	Graphs of motion signals for behaviours: jumping . . . . .	176
5.20	Graphs of motion signals for behaviours: running and group .	178
5.21	Graphs of motion signals for behaviours: waving . . . . .	179
5.22	Decision tree for behaviour analysis . . . . .	180
5.23	Pseudocode for object classification . . . . .	181
6.1	Complete Dual-Channel system . . . . .	195

B.1 Pseudocode for object labeling method . . . . . 209

## List of Tables

3.1	Motion detection performance: Blobs . . . . .	82
3.2	Motion detection performance: Pixels . . . . .	84
4.1	Tracking results . . . . .	132
4.2	Dual-Channel tracking rules . . . . .	136
5.1	Classification of rigid/periodic non-rigid objects . . . . .	160
5.2	Classification of behaviours: confusion matrix . . . . .	180
5.3	Classification of behaviours: results . . . . .	181

# Chapter 1

## Introduction

Automatic visual surveillance is a long held goal yet it is a relatively young science. Researchers have been working on video since the beginning of computer vision in the 1960's. Initially video compression and storage was the primary focus and it was not until the late 1970's that the concept of object tracking began to split off into a coherent and separate topic of research. One of the key figures in this early work was J. K. Aggarwal of the University of Texas at Austin. In 1979, Aggarwal, with N.I. Badler, held a workshop on "Computer Analysis of Time-Varying Imagery". This was followed in 1980 with the special issue of PAMI on *Motion and Time-Varying Imagery* and a book in 1988 called *Motion Understanding: Robot and Human Vision*, both of which introduced a wide range of new approaches. Some of these were later abandoned but a few proved to have lasting success. By the 1990's two competing paradigms, background modelling and particle filters, were vying for supremacy.



Today the subject has also acquired great economic and political significance due to the extraordinary number of CCTV cameras adorning city streets. While politicians tout issues of security and civil liberty groups question its impact on privacy, the CCTV industry is quietly questioning its effectiveness. Between 1996 and 1998, three-quarters of the UK Government's crime prevention budget was spent on CCTV, yet studies have shown that CCTV is less effective at preventing crime than simply improving street lighting.<sup>1</sup> Part of this failure is due to the human component. Human operators are simply not able to pay attention to boring CCTV feeds for long periods. On average there is one CCTV operator monitoring 60 camera feeds, clearly too much for a human to deal with.

This is a clear case where computers could be usefully employed. The aim is to detect and highlight only interesting or unusual events (such as a violent crime or a fight) and to attract the attention of a human operator. Such a system may also satisfy the misgivings of civil liberty groups. If only crimes are detected, and all other video data is deleted, the privacy of law abiding citizens is preserved. Unlike a human operator, a computer algorithm cannot be secretly racist or corrupt [76].

## 1.1 Human Visual System

While the science is young, the goal of visual surveillance predates even human evolution. The ability to detect and track a prey, or for the prey

---

<sup>1</sup>“Report says CCTV is overrated”, The Guardian, June 28, 2002.

to avoid a predator, evolved many hundreds of millions of years ago. Yet it is still largely a mystery how this is achieved. The Reverse Engineering of the Human Visual System (RE:HVS) project, of which this thesis is a part, has the aim of analysing the powerful structures of the visual cortex and replicating them in software.

Humans Beings evolved as hunters, and as such, have a highly developed tracking ability. Studies of the brain have shown that, while highly interconnected, tracking in the HVS can be divided into a number of stages: (1) Motion is detected at a local level (Motion sensitive neurons with a spatio-temporal response have been isolated in the visual cortex), (2) local detections of motion are integrated into connected objects, (3) these objects are then tracked using, according to some theories [75], a dual-channel form-motion structure. (4) Following this, or perhaps concurrent with it, the brain analyses the behaviour of the tracked object and categorises it.

The HVS is able to extract an enormous amount of information from motion, sometimes even the mood of a pedestrian from the way he walks. Figure 5.1 in Chapter 5 shows the classic 'point light display' set up where various human actions are filmed with only points at each joint made visible. Experiments have shown that while still images from such a video are meaningless, when in motion even complex actions such as dancing can be clearly seen. How does the HVS achieve this extraordinary feat? What signals are extracted from the video and how are these signals processed?

These questions remain unanswered, although we can safely assume that if the brain can achieve this trick the required information must be present

and the only impediment to building an automatic recognition system is finding the correct signal in the video data and processing it properly.

An interesting illustration of the presence of motion signals is in the work of cartoonists whose trade depends on picking out salient motion features and exaggerating them. This can be seen in figures produced by professional animator Richard Williams: Figure 1.1 shows the difference between a character walking and running. Note the differences in the swing of the limbs and the bobbing of the head. These are instantly apparent to the human eye but very difficult to reliably assess by computer. While 'walking' and 'running' can be detected by our eyes as nearly invariant attributes of these characters, it is quite difficult to consciously define those attributes. Cartoonists are taught to find them. In Figure 1.2 the animator has altered the characters to insert emotion into the walks. The left-hand character is made to appear angry. The animator explained this as due to the tilt of the body. The right-hand character is made to appear happy.

That our vision systems can so easily detect these complex differences proves that there is invariant behaviour information available. That we find it difficult to define this information suggests that the processing is at least partly at the preconscious level.

An important aspect of the tracking problem is the question of how objects in the mind relate to reality. How do we match the partial information we experience through our senses with notional objects we hold in our minds, and where do we acquire these object models? This question has been considered since antiquity. Plato discussed this in his "*Theory of Forms*",

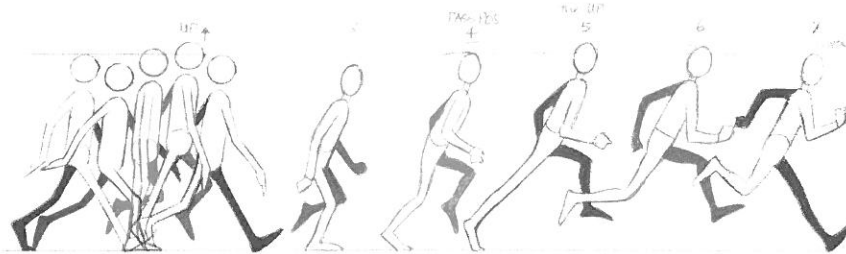


Figure 1.1: Cartoonist Richard Williams illustrates the differences between walking (left) and running (right) in terms of character posture and motion. Extract from *The Animator's Survival Kit* [184].

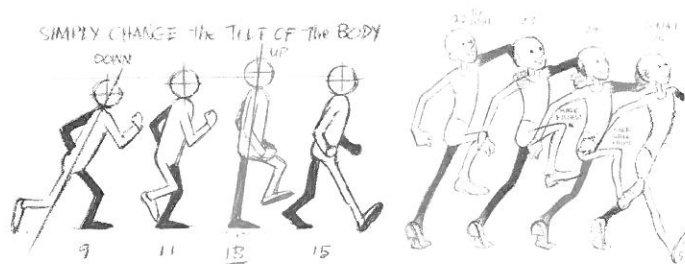


Figure 1.2: With relatively subtle changes to posture and motion, emotion can be clearly seen in these characters. Left shows an angry walk, with a note above by the cartoonist to explain how this was achieved. Right is a happy walk. Extract from [184].

explaining that the mind acquires 'forms' of perfect general objects from a world beyond this universe. (Of course, modern neurology takes a different view!)

In more recent times, the branch of philosophy called *Phenomenology* explores the connection between the *phenomenon*, the detectable experience of an object, and the *noumenon*, the notional object model stored in the mind. This issue is seen by some as being the primary question in philosophy and has been written about by such luminaries as Immanuel Kant, Georg Hegel and Karl Marx.

These ancient philosophical ideas, which might seem esoteric, have gained renewed significance in the context of computer science. The possibility of analysing *phenomena* computationally in order to build and match *noumena* is now a practical, and even commercial, research topic. From 1992-96 the European Union funded an extensive collaborative study of philosophical and computational approaches to phenomenology [20].

## 1.2 Aims and assumptions

This thesis addresses the task of analysing video to detect and track arbitrary moving objects and to understand and classify objects using behaviour information. Due to the recent proliferation of security applications, and their economic importance, research has been focused on CCTV surveillance videos.

The needs of CCTV applications allow the requirements of a successful system to be defined:

- **Flexibility** – CCTV is recorded in a wide range of different formats, video qualities, noise levels, etc. The algorithm must be robust to indoor and outdoor conditions, changing lighting and changing weather conditions.
- **Object model** – the algorithm must be able to quickly find and track arbitrary moving objects as they enter the scene. This means making as few assumptions as possible about object appearance and avoiding the use of predefined object models.
- **Short bootstrap** – unlike assembly line machine vision applications, the environment of CCTV cameras is uncontrolled and subject to change. Installation time and costs are important. Therefore the algorithm bootstrap time must be kept to a minimum. A scene specific learning routine is impractical as it increases installation time and may fail if the scene changes during operation.

Conversely, knowledge of CCTV allows sensible limitations and assumptions to be placed on these requirements:

- **Small objects** – CCTV cameras are generally installed in elevated locations, looking down at an angle on the scene. Cars and pedestrians will be small in size relative to the scene area.

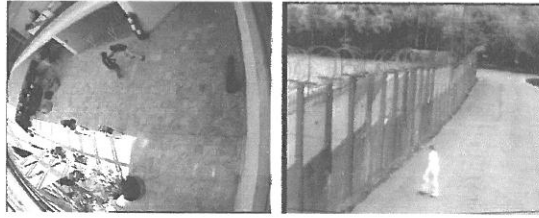


Figure 1.3: Examples of two CCTV videos used in testing. The left hand image is taken from the CAVIAR database. The right hand image is from the Home Office *i-LIDS* database. These two frames illustrate differences in viewing angle, colour, picture quality, and scene layout evident in CCTV applications.

- **Moving objects** – while objects may start and stop within the scene, it can be assumed that all objects of interest will enter and leave the scene in motion.
- **High Frame rate** – it can be assumed that the video frame rate will be sufficiently high so that interesting objects moving through the scene appear in many frames.
- **Static camera** – CCTV cameras are at fixed locations. Some have fixed views while others allow the user to pan, tilt and zoom the image. Here, it is assumed that the view is fixed; however, the practical problem of a shaking camera is dealt with in Chapter 3.

The algorithms developed in this project were tested on a wide range of video types: see Figure 1.3 and Appendix A for details.

Many of the key questions of visual tracking remain not just unanswered but rarely explicitly asked. Chapter 2 will show that many methods reported in the literature contain a motion detection component while others do not.

What is the appropriate role of motion detection in tracking? What is the best way to achieve it? Are approaches which avoid motion detection valid and what assumptions do they make?

Some methods, such as particle filtering, maintain a strong focus on an object appearance model combined with location prediction. Others contain neither component. Is location prediction necessary? What issues arise with appearance models in visual tracking and how does this relate to other aspects of the system?

As well as developing a new approach to all these tasks, this thesis also aims to answer these fundamental questions. The final task of behaviour analysis is largely an open question without a consensus evident in the literature. This thesis will take a novel approach which has been inspired by the HVS.

### **1.3 Thesis structure**

This thesis contains seven chapters, including this one, as follows:

#### **1. Introduction**

Chapter 1 discusses research motivations and context.

#### **2. Literature review**

Chapter 2 provides a thorough examination of the machine vision literature on visual detection and tracking, from the earliest days of "Image Sequence Analysis" to recent developments in particle filters and Gaus-



sian mixture models. Methods are categorised by their approach to the spatio-temporal nature of video data. Background modelling techniques, which work on 1D pixel processes, are described as  $1D+2$  approaches. Prediction based methods are described as  $2D+1$  approaches. Motion Distillation is  $3D$ .

### **3. Motion Detection**

Chapter 3 covers the new 'Motion Distillation' method which detects moving objects through spatio-temporal wavelet decomposition. Results are compared with those of background modelling methods. The chapter concludes with an exploration of general spatio-temporal wavelet theory.

### **4. Dual-Channel Tracking**

Chapter 4 explains the problems with location prediction and offers a novel dual-channel form-motion alternative. The dual-channel approach is used to track arbitrary objects through static and dynamic occlusions.

### **5. Behaviour Analysis**

Chapter 5 presents a new method of analysing the behaviour of tracked objects based on the non-binary motion detection information generated by the Motion Distillation. Pedestrians may be distinguished from vehicles and then further categorised into a range of behaviours.

### **6. Discussion**

Chapter 6 ties together the full process of detection, tracking and un-

derstanding and lays out suggestions for future work.

## Chapter 2

### Literature review

Motion processing schemes can be divided into three primary categories, each with its own spectrum of sub-categories. The first category, which includes the range of methods from optical flow to Kalman and particle filtering, are fundamentally frame-by-frame appearance-based with temporal prediction. These can be thought of as '2D+1' methods, meaning 2D spatial appearance search with temporal prediction. The position of a particular method in the 2D+1 spectrum depends on the number of features, their size and complexity and the complexity of the temporal predictive model.

Another category is composed of those methods which exploit temporal statistics of some form to segment moving regions in the video stream, with more complex methods being developed to counter unconstrained illumination states, noise and clutter. This can be called '1D+2', because following the 1D motion detection stage, 2D spatial tracking and object integration must be carried out. This large range of methods stretches from frame dif-

ferencing (still used in MPEG4 compression) through temporal mean, median and mode statistics, to linear prediction such as pixel-wise Kalman filter-based background modeling. The final category is the rarely applied spatio-temporal 3D methods. Included here are a number of hardware video processing designs, a handful of interesting wavelet-based methods which are sometimes labeled as optical flow calculations, the human visual system (HVS) itself, and the work of this thesis.

This chapter will explore in detail the use, strengths and results of most of these subcategories reported in the literature. Figure 2.1 shows a ‘map’ of visual tracking science representing the vast majority of reported systems. Techniques follow a path from the Video data through *Detection*, *Tracking* and finally some *Behaviour Analysis* method. The top path represents foreground, ‘2D+1’ methods. This route to tracking is shown as a dotted line because it is incomplete without *a priori* model. Techniques such as particle filtering follow this route. The bottom route represents ‘1D+2’ methods, primarily background modeling based motion detection. Some methods are initialised using the ‘1D+2’ route, and then switch to the top route for tracking. Among them is *BraMBLe* [91] (discussed in detail below) which uses background modeling to detect new objects and to acquire an appearance model but tracks them using a particle filter. The middle route represents true spatio-temporal 3D tracking. Behaviour analysis is carried out after tracking is achieved. However, there is currently little consensus on how this should be done.

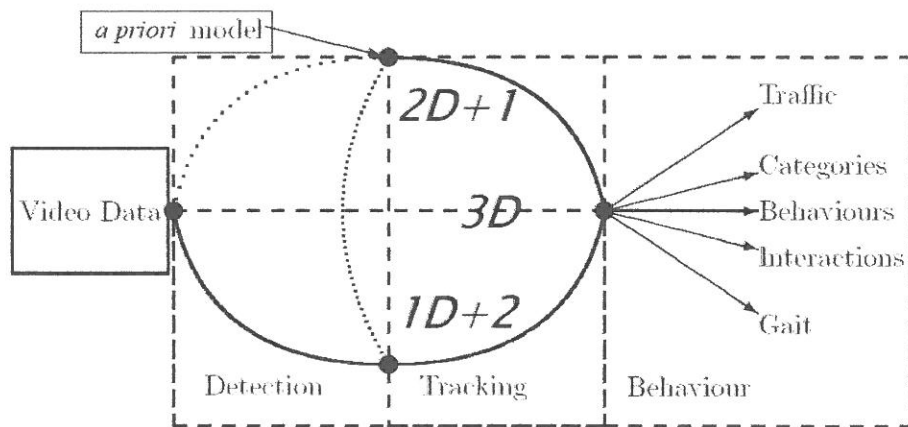


Figure 2.1: 'Mindmap' of Visual Surveillance Literature

## 2.1 $2D+1$ – Foreground techniques

### 2.1.1 Optical flow

Methods in computer vision often derive either from HVS theory or as accidentally discovered *ad hoc* solutions. Optical flow appears to fit both categories. The concept of optical flow originates in Gibson's (1950) early neurological work "The perception of the visual world" [74]. The technique, as applied to computer vision, is based on the tracking of single pixels through image sequences. The difficulties of this approach were recognized in the important 1981 paper by Horn and Schunck (1981) [84]. The intensity of a pixel offers only one constraint but the velocity of a point has two components; thus optical flow cannot be calculated locally. This effect is often referred to as 'the aperture problem'. The solution is smoothing [162] or (often implicit) object model based constraints [4]. However, the former causes problems at

real discontinuities and the latter fails when the limits of necessarily simple models are reached in cases of 3D rotation or non-rigid motion. Aggarwal (1998) [5] noted two classes of solution; using the piece-wise or region-wise smoothness assumption and the coarse-to-fine multiscale approach. Research through the 1990's focused on hardware implementations of these theories as retinomorphing devices [57] and calculations of optic flow using wavelets [26]. However, as Horn noted in his 1993 retrospective on his earlier work, optical flow still faced a fundamental sensitivity of illumination change. McFarlane and Schofield (1995) [120] noted that in surveillance applications optical flow fails because objects are small with respect to the scene and there are many motion discontinuities. Added to this, a deeper understanding of the HVS has eroded optical flow's neurological foundations [192, 193]. Today, the technique is still sometimes used in dense crowd situations when individual objects cannot be segmented [10].

### 2.1.2 Feature tracking

Moving up the scale of feature size, we may attempt to track 2D features such as edge sections [142, 189, 24] or corners [151]. These features are extracted from each frame, and then inter-frame correspondence is established between the features. To overcome noise and detection ambiguity issues, a network of motion constraints is applied, usually in the form of a system of non-linear equations [5]. This approach was introduced by Ullman (1979) [174] and developed by many researchers through the 1980's [150, 129]. An inherent flaw in the method is its reliance on an explicit motion model and thus it

breaks down in the presence of multiple objects [5] or non-rigid objects [3, 120] common in surveillance.

### 2.1.3 Location prediction

As object models become more complex, it becomes necessary to avoid global search and to predict the object motion and then to limit the search to that area. The first successful predictor was the Kalman filter [72, 181], a linear filter borrowed from the world of radar and control theory. However, it was quickly discovered that non-linearity was extremely advantageous in visual tracking, and the Extended Kalman (EKF) [111, 40], Unscented Kalman (UKF) [11, 178], Sequential Monte Carlo [88], Markov Chain Monte Carlo [87], Condensation/particle filter [89] and EKF [50] and UKF directed particle filters [153, 176] have been tried. The above papers often do not make sufficiently clear that the filter stage serves only as a location predictor in order to direct the search. A separate appearance model, already known or learned, must be tested at each candidate location suggested by the predictor.

A Kalman filter is a type of on-line least-squares analysis system and thus gives a linear response [181, 19, 72]. Harris (1992) [81] and Blake *et al.* (1993) [33] were among the first to employ the Kalman to predict probable object locations or search candidates in visual tracking applications [37, 13], although it had been used for many years for object tracking in radar. The algorithms are initiated with a target model and location, search for this

model in the next frame, and use this position change to predict the next location. Search is begun at the candidate location and worked outwards until a good match is found. The error (difference between predicted location and actual location) is fed back to the tracker to improve the next prediction. The Kalman fails when the data (object location over time) is non-linear or when the noise is non-Gaussian – which translates into poor performance in cluttered scenes [89]. Results can be improved with the EKF which uses Taylor series expansion to linearize non-linear functions [111, 40] and the UKF which uses the ‘unscented transform’ [11, 178] to delinearise predictions [97, 166]. These improved versions of Kalman still react poorly in cluttered conditions due to their mono-modality, tending to get stuck on local minima.

In 1996, Isard and Blake [88] proposed a tracker using Sequential Monte Carlo methods. Imperfections of the motion model would be overcome by introducing a random component into the search strategy. Each search candidate area (which would later become known as a particle) was assigned a weight based on confidence. Multiple particles of different weights can be maintained simultaneously to model a multimodal data field and thus ‘multiple hypotheses’ of the target location, resulting in better performance in cluttered scenes. The particle location and weighting are altered using results fed back from the appearance matcher. Particle filters generally give very good results but fail in cases of sudden change in motion. They are, of course, only as good as the appearance model used in the search stage.

Isard and Blake (1998) extended their earlier work with the ‘ICondensa-



tion' algorithm [90] to include Importance Sampling in order to allow combination of multiple separate appearance models (in this case, shape and color). Other work has involved directing each particle using an EKF [50] or UKF [153, 176] filter to improve location estimation. Particle filtering with Sampling Importance Resampling has also been explored [130, 12]. Choo and Fleet (2001) [43] reported a particle filter extended for tracking very complex target models, calling it a 'Hybrid Monte Carlo'. Hue *et al.* (2000) [87] gave a detailed discussion of tracking multiple objects with particle filters.

## 2.2 Appearance models

Appearance models, or observation models, used in visual tracking are very similar to techniques for object recognition in static images. The most basic appearance models are those of optical flow and feature tracking techniques (which consider single greylevel pixel values or corners plus relational constraints). Early Kalman filters generally used template matching [55] or color histograms [188]. Other reported possibilities include Harris 'Interest Points' [73]. Interestingly, some techniques in this area seem not to have changed greatly in recent years.

The appearance model used by Isard and Blake (1996) [88] was based on curved segments, while their 1998 work [90] combined this with a colour matcher. Other methods used corners or line segments but tend to be more clutter sensitive. Rui and Chen (2001) [153] and others [136, 130] used colour histogram matching with particle filtering. This has the advantage over shape

matchers of being rotation invariant. Nummiaro *et al.* (2002) [136] also claimed that colour histograms are robust to partial occlusion. However, this is clearly conditional on the target's structure. Nummiaro's application was face tracking where the target was of approximately uniform colour. Zhou *et al.* (2004) [196] discussed techniques of adaptive appearance models for use with particle filtering.

The choice of model is highly data and target dependent. Non-rigid pedestrians are generally harder to track than rigid vehicles for various reasons, but notably because of deformations. Histogram matching gives more robust results for pedestrians, while template matching is more suitable for vehicles.

Selinger and Wixson (1998) [156] used this very change in the appearance of an object, i.e. the failure of their static model to match the true appearance of some targets, to cheaply distinguish between rigid objects such as vehicles and non-rigid pedestrians.

Boosting [68] is a general machine learning method for improving poor recognition methods. Boosting occurs in stages, incrementally adding to the current learned function. At every stage a *weak learner* (one that has an accuracy only slightly greater than chance) is trained with the data. The output of the weak learner is then added to the learned function, with some strength proportional to the accuracy of the weak learner. The data is then reweighted. Incorrectly matched examples are *boosted* in importance. In the context of tracking, boosting can be used to improve appearance models given poor data. Okuma *et al.* (2004) [138] boosted color histogram matching in the context of particle filtering. Fan *et al.* (2006) [63] compared the ben-

efits of this method with those achieved by improving the prediction stage, concluding that boosting the appearance model gives better results.

Random sample consensus (*RANSAC*) [67] attempts to overcome the imperfect model problem and noise by working through a search space to find the global minimum. The search is stopped after a certain number of tries; this number may be calculated on-line or predetermined. The scheme has been used for feature detection in noisy images [44].

### 2.2.1 Illumination

Belhumeur and Kreigman (1996 and 1998) [21, 22] explored the key difficulty of image and video processing; that truly illumination invariant image features do not exist. They noted that in the case of face recognition the variability in an image due to illumination may exceed that due to a change in the subject's identity. They proposed modeling appearance change as an 'illumination cone' as a way to predict appearance given a sparse illumination data set. These ideas were extended by Belhumeur *et al.* (1999) [23], Chen *et al.* (2000) [42] and Baker *et al.* (2003) [15].

Drew *et al.* (1998, 1999, 2002) [59, 60, 58] detailed work on illumination invariant colour image recognition. Drew found that invariance is improved by matching the DCT compressed images. Black *et al.* (2000) [32] attempted a direct model of appearance change due to illumination based on four causes – object or camera motion, lighting source changes, specular reflection and iconic changes.

The issue of shadows can be viewed as an aspect of illumination. Prati *et al.* (2001 and 2003) [147, 148] noted that while shadows will usually be detected by backgrounding techniques, they can cause incorrect object merging and shape distortions. An object which transits to or from a shadowed region may be lost due to incorrect segmentation. Even histogram matching may fail under some conditions. Numerous shadow detection algorithms have been proposed, mainly based on comparing motion detected regions for texture or colour similarity with the background. Bevilacqua (2003) [28] simply used ‘darkness’ and smoothness to select shadows in a traffic monitoring application.

## 2.3 $1D+2$ – Background modeling

### 2.3.1 Image differencing

While optical flow was at its peak of popularity in the 1980’s another, *ad hoc*, method of motion detection was emerging. The first paper to discuss it was Jain and Nagel (1979) [94], which proposed a method of ‘intraframe differencing operations’ (later known as ‘image’, ‘frame’ or ‘temporal’ differencing) for image segmentation based on motion. A similar method had been popular in the early 1970’s for detecting change in satellite imagery [106, 119]. Jain and Nagel reported that this had only a limited applicability in surveillance “because images [sic.] can be extracted only when they have been displaced completely from their position in a specified reference frame, which is usually

the first frame of the sequence.” This problem of incomplete segmentation, which would arise continuously over the coming years, was addressed in a later paper; Jain and Aggarwal extended this work using partly segmented regions as a ‘motion cue’ to extract the rest of the object using edge profiles and morphological growing [93].

Many of the developments to image differencing explored in the 1980’s would find later use. Then, as now, traffic surveillance was a promising application which presented problems of robustness and speed [86]. In Jain’s (1981) paper [92] he extended his earlier work to a traffic monitoring application and applied contemporary understanding of motion attention in the HVS. Anderson (1985) [8] attempted a speed-up using “Image Pyramids”, i.e. multiscale analysis. Tsukiyama (1985) [173] used Jain’s method coupled with a motion model to detect single people in video, and Lee (1988) [105] extended this for groups of people. Dinstein (1989) [53] proposed a similar technique for a visual motion alarm and Brofferio (1989) [36] explained its use for video compression of temporal redundancy. This simple technique is used to this day in the MPEG4 video compression standard.

Due to the computational limitations in the 1980’s, the above techniques were tested on relatively short video clips. With longer clips, complex illumination changes reduce robustness. In 1989, Jain [161] proposed using the difference of thresholded edge maps, rather than of the intensity images themselves. In his perceptive review of this early work, Rosin (1997) [152] noted that the key to this technique for motion detection is the choice of thresholding level, with methods using adaptive and local thresholds being

far superior to global, experimentally chosen levels. While using edge maps does increase robustness in some ways, it removes important illumination information and makes automatic selection of local threshold levels impossible.

### 2.3.2 Statistical filtering

Frame differencing can be viewed as the simplest form of statistical background modeling, where in this case, the model is composed of pixel values taken from a single earlier frame. Long and Yang's (1990) [110] influential paper set out a number of methods involving the computation of a running average of pixel values in order to achieve 'Stationary Background Generation'. They point out that the stationary parts of the image are present for the majority of the time, and can be detected statistically. Using statistical methods the background can be constructed even when some moving objects are present in the video. The paper also addresses the issue of an imperfect background (mentioning the effect later known as the 'transient background problem') and proposes a morphological solution. They tested the method on both indoor and outdoor scenes. However the videos were quite short (54–71 frames) and so it is likely that they didn't experience many problems using the 'mean' as their statistic. It is easy to understand why the seemingly obvious step of extending differencing to true statistical background modeling arose only in the 1990's by noting the comment from the final page of Long and Yang's paper:

"a complete run [of 54 frames] required approximately six hours

on a 3 Mbyte SUN 2/170”

On a modern computer this process would be completed in under one second. Many researchers continued to use the temporal mean filter due to its simplicity [29, 122, 85]. However, simple statistics are not robust in complex lighting situations, and thus require *ad hoc* post detection noise removal steps [28]. The median or mode of the pixel statistics can be used as a more robust measure of the background, because this is not so easily skewed by outliers. This is particularly important when the background frame must be calculated while objects are in motion in the scene.

As Kalivas (1990) [98] noted, “mean filtering is more effective in the case of white Gaussian noise and median filtering in the case of salt-and-pepper noise and burst noise.” Kalivas explored the use of temporal and spatio-temporal mean and median filtering for video noise reduction. Gloyer *et al.* (1995) [77] were perhaps the first to report use of a temporal median filter for background modeling. They noted that the median’s robustness to shot noise and outliers also made it robust in the presence of transient moving objects. They used it to bootstrap a background model in the presence of moving vehicles. McFarlane and Schofield (1995) [120] published a similar method for tracking piglets. Lo and Velastin (2001) [109] formalized the idea, calculated the median value of the last  $n$  frames as the background model, and compared it to other methods. Cucchiara *et al.* (2003) [47] argued that such a median value provides an adequate background model even if the  $n$  frames are subsampled with respect to the original frame rate by a factor of 10. In addition, they proposed to compute the median on a special set of values



containing the last  $n$  sub-sampled frames and  $w$  times the last computed median value. This combination increases the stability of the background model.

Wren's *Pfinder* (1997) [186] has been described [144, 47] as a form of mode filter, calculating a 'running Gaussian average' for each pixel. This has a memory advantage over temporal median filtering, in that only two parameters ( $\mu$  and  $\sigma$ ) of the Gaussian *pdf* are stored for each pixel, and are updated using the equation:

$$\mu_t = \alpha I_t + (1 - \alpha)\mu_{t-1} \quad (2.1)$$

The *Pfinder* method has proved influential and has frequently been imitated (see reviews in [47, 121, 146]). The influential  $W^4$  system reported by Haritaoglu (1998) [80] used a simplified variation on Wren's technique, finding only the maximum and minimum pixel intensities and the maximum inter-frame difference values, using a few seconds of sample scene video without moving objects.  $W^4$  has become a popular foundation of many reported systems [128, 175]. Lou (2002) [112] coupled  $W^4$  with homomorphic filtering to normalize for variations in illumination.

### 2.3.3 Other methods

An alternative to the usual statistics is to use a form of linear predictor to estimate the most probable pixel value at each location, thus constructing the background image. The first paper to propose this approach was Karmann



(1990) [99] who used a Kalman filter for each pixel. The Kalman has an advantage in that it directly provides the appropriate threshold level. Koller's influential 1994 paper [103] also used Kalman filtering for background modeling for the application of traffic monitoring. Toyama's *Wallflower* system (1999) [170] used a Wiener filter to to make probabilistic predictions of the expected background. This paper contains a very thorough comparison of eight common background techniques. *Wallflower* became an important benchmark method for detecting quickly moving cars [116] in traffic surveillance. It was later used in some commercial surveillance systems [46]. However, as McIvor (2000) noted in his review [121] "The *Wallflower* algorithm requires the storage of over 130 images, many of which are float valued. This requires significant statistical analysis per pixel per frame to adapt the coefficients."

Some recent publications explore the use of nonparametric kernel density estimation to calculate the pdf of intensity values for each pixel [61], giving a multimodal result similar to GMM.

#### 2.3.4 Maintenance

Cucchiara *et al.* (2003) [47] defined the principal difficulty of all statistical backgrounding techniques as the conflict between the "stationary background problem" (SBP) and the "transient background problem" (TBP). To accurately segment moving objects, the background must be updated regularly to reflect changing conditions (TBP). However, updating runs the risk of including some part of a foreground object in the background, and this must be

avoided (SBP). The result of failing SBP is false positive detections known as ghosts, while TBP failures result in false negatives or inaccurate segmentation. For a practical system, backgrounds must be updated to deal with changing illumination conditions and motion clutter such as moving tree branches. Some solutions to this difficulty will now be described.

As an extension to the mode filter methods mentioned above, Stauffer and Grimson (1999, 2000) [164, 165], developed a multimode Gaussian Mixture Model method (GMM). This has the advantage of allowing a pixel to oscillate between a few different frequent values. This gives greater robustness to motion clutter, such as oscillating tree branches. However, as Power and Schoonees (2002) [146] noted, the iterative Expectation-Maximization method for calculating GMM is computationally expensive and can become unstable. Stauffer required specialized hardware to achieve his reported real-time results. GMM methods have since become a standard part of the computer vision toolkit [187, 25, 146].

Koller *et al.* (1994) [103] used a similar scheme to Wren (1997) in part of their complex traffic monitoring system, with the addition of a selective update to improve speed. Only those areas in recent contact with foreground elements would be updated. However, the lower the update rate of the background model, the less a system will be able to quickly respond to the actual background dynamic.

A number of methods try to encode repeating movements into their background models in order to deal with motion clutter. Monnet (2003) [126] used "Dynamic Textures", blocks of video motion which are allowed to re-

peat without being classified as foreground. Pless (2005) [145] and Ohta (2001) [137] used similar PCA methods to achieve the same goal. Pic *et al.* (2004) [143] were concerned with the case of a moving or shaking camera. All these methods rely on breaking the frame into small regions which are assumed to be quasi-stable.

Toyama (1999) [170] also noted that “the difficult part of background subtraction is... maintenance of a background model”. His solution was a multiscale decision and update approach, involving pixel-level, region-level and frame-level detection of various illumination events.

### 2.3.5 Thresholding

Most background modeling techniques select a threshold level experimentally. Cavallaro and Ziliani (2000) noted [41]:

“The thresholds that are needed to extract the changed areas have to be tuned manually according to the sequence characteristics and they often need an update along the sequence itself. This main drawback limits this approach for automatic applications. Moreover, if the contrast of moving objects is not sufficiently high compared to the camera noise, there might not exist a unique threshold to get rid of noise and to preserve the motion information. In this case detected objects have not precise edges.”

Rosin (1997) [152] observed that experimentally selected threshold values for background models are not robust, and for practical surveillance systems they must be chosen automatically. He further noted that locally variable, rather than global, thresholds are often necessary. Rosin's technique was to pick a threshold level which maximizes "clumpiness" of change regions (an assumption of motion's spatial coherence), thus avoiding spatially random noise. Some recent publications have observed that if the threshold selection method is robust enough, results comparable to background modeling can be achieved by image differencing [27, 70].

### 2.3.6 Tracking after detection

After moving objects have been detected by background subtraction, they must be tracked. However, this task is greatly simpler than is the case for foreground tracking techniques because the tracking stage now has knowledge of object locations. The task is frequently described as 'maintaining object identity through occlusions' and "Blob Tracking". The task involves modeling of some discriminative feature of the moving object and making assumptions of motion inertia.

Inertial prediction, often in the form of a Kalman filter, is generally more successful in traffic monitoring [111] than with human tracking [132, 83] because human behaviour is less predictable [195]. It is often not made clear that the predictive step (i.e. the Kalman filter) and the identity check (i.e. histogram matching) are separate. When there is no ambiguity (isolated

objects with small interframe motion) a simple interframe blob overlap test is sufficient [71]. Jorge *et al.* (2004) [95] maintained object identity using a list of logical rules governing merges and splits of multiple objects. This was formalized as a Bayesian Network. Wang *et al.* (2000) [180] carried out appearance based blob matching with motion constraints – in this case a piecewise assumption of constant acceleration. Bascle *et al.* (1994) [18] used post-detection Kalman filter to predict both location and shape. Lou *et al.* (2002) [111] used EKF for vehicle tracking. Niu *et al.* (2003) [132] and Lei and Xu (2005) [107] used second order Kalman filters. The *BraMBLe* system of Isard and MacCormick (2001) [91] used a particle filter to track an *a priori* unknown and changing number of blobs.

Zhou and Aggarwal (2001) [195] explored blob matching using a number of metrics including PCA selected color features, compactness and shape matching, the latter two being sensitive to accurate segmentation. Xu (2004) [187] used a mixture of speed, size, color and the ratio of the major axis to the minor axis of the best-fit ellipse. Collins (2003) [45] used multiple scales to ‘zoom in’ on the target, using the previous positions as a starting point. This used an implicit assumption of small interframe movement. Fuentes and Velastin (2001) [70] use color histogram blob matching with location prediction while allowing for group formation and dissolution. Xu (2004) [187] follows a GMM based detection stage with blob matching based on a combination of shape, size and velocity information. Naftel and Khalid (2004) [128] used RANSAC to plot paths through noisy detection locations.

## 2.4 3D – Spatio-temporal techniques

It has been noted for some time that the Human Brain uses spatio-temporal ( $s-t$ ) processing at the heart of its visual system. Tsotsos *et al.* (1988) [172] presented results from convolving video with  $s-t$  Difference of Gaussian filters for velocity specific feature detection. Peng and Medioni (1989) [141] and Peng (1991) [140] use  $x-t$  slices of video to detect paths and speeds of objects. Kalivas and Sawchuk (1990) [98] investigated  $s-t$  mean, median and mode filters for video noise removal and compression.

Researchers at the Computer Vision Lab at Linköping University, Sweden have published a number of papers covering many aspects the  $s-t$  approach [9, 16, 62, 65, 66, 78, 100, 163, 194]. Bårman *et al.* (1991) [16] achieved similar results to Peng and Medioni using a complex tensor representation of  $s-t$  ‘cones’. Bårman managed to extract both velocity and acceleration information. Knutsson *et al.* (1992) [100] compared these conic filters to Gabor filters for 3D medical imagery. Parts of Granlund and Knutsson’s 1995 book [78] cover these topics in detail. More recently, Farnebäck (2000, 2001) [65, 66] used similar filters to segment video into regions of coherent motion, and Andersson and Knutsson (2003) [9] presented work on non-regular sampling in 3D data. The Linköping research involved both 3D  $s-t$  video, 3D medical imagery and time-varying 3D imagery, noting that the same processing techniques are applicable to both. With  $s-t$  data, slope along the time axis is due to velocity, and curves are due to acceleration. Other papers which make this connection include Sohn *et al.* (2004) [163] and Faas and

van Vliet (2003) [62]. Yu *et al.* (2001) [194] compared the Linköping conic filter with the Hough transform and the Difference of Gaussians filter.

Few  $s-t$  papers are concerned with visual tracking and surveillance. Sato and Aggarwal (2004) [154] used a  $s-t$  Hough transform based method to track objects after they have been detected by background subtraction. Object interactions are detected as crossed paths in  $s-t$  space. Niyogi and Adelson (1993) [135] at the MIT Media Lab reported a gait recognition system which analysed the distinctive braided pattern in  $y-t$  slices of videos of walking pedestrians. Kobayashi and Otsu (2006) [101] followed background subtraction by convolution with hundreds of  $s-t$  filters (called *CHLAC*) each of which is sensitive to different  $s-t$  angles. The output is fed into a Discriminant Analysis stage with the aim of gait recognition.

Also of interest is the recent work of Michal Irani of the Weizmann Institute of Israel. Her work has included video completion [183] and behaviour recognition [159] using  $s-t$  textures. Similarly, Pless (2005) [145] used  $s-t$  filters to build a background model which was tolerant to some motion clutter.

## 2.5 Behaviour analysis

After objects have been detected and tracked a large number of capabilities are open to us. This final stage of processing is actually the real goal of this whole process. We may wish to distinguish cars from people, groups of people, different activities such as bicycle riding, or detect abstract behaviours.



Alternatively, we may wish to log volumes of traffic (either vehicular or pedestrian) and derive statistics based on this. We may divide surveillance systems into three broad categories. The first uses only the object track and path information while the second focuses on other available information. The third is only interested in identifying individuals using gait information.

Malik *et al.* (1995) [116] developed a traffic monitoring system which detects crashes and congestion based on vehicle behaviour. Makris and Ellis [115, 114] developed a system which learns common pedestrian routes by merging paths together. Scene entry and exit points can be detected. Dee and Hogg (2004) [51] used a scene model to break tracks into a tree of goals and subgoals. If a pedestrian deviates from a likely path, his behavior is flagged as suspicious. Owens *et al.* (2002) [139] used the techniques of Novelty Detection [117, 118] to automatically pick out abnormal pedestrian routes. Niu *et al.* (2004) [133] used HMM to detect complex interactions such as stalking.

Heikkilä and Silvén (2004) [83] combined templates and speed information to distinguish pedestrians from cyclists. Zhou and Aggarwal (2001) [195] used nearest neighbour with multiple features to distinguish four categories of object (vehicle, bicycle, single person and people group), and reported a 5% misclassification rate. Selinger and Wixson (1998) [156] and Curio *et al.* (2000) [48] distinguished people from cars using the periodic deformations of people's blobs. Bobick and Davis (1998) [34] integrated the object's motion into a 'motion history image' and then applied template matching (referred to as 'temporal templates') to detect specific behaviors.



Once objects have been classified, their interactions may be understood. Collins *et al.* (2000) [46] used Hidden Markov Models to classify events such as 'human entered a vehicle' and 'human rendezvous'. Kojima *et al.* (2002) [102] sought to develop a system for automatically describing human behaviour after these behaviours have been successfully extracted.

Gait analysis is an important area of surveillance research. Vega and Sarkar (2003) [177] used gait information to distinguish between people who are walking, jogging or running. However, most work on gait uses videos produced in indoor labs or under highly controlled conditions. Yoo and Nixon (2003) [191] discussed the need for feature extraction and the distinctive near-sinusoidal motion patterns of human gait. Many approaches rely on extracting a 3D human body model first, before gait can be analysed. Bharatkumar *et al.* (1994) [30] used the Medial Axis Transformation to apply a skeleton model. Dockstader and Tekalp (2002) [56] tackled this as a tracking problem, where models are applied at coarse and fine resolutions with kinematic constants. Lee (2003) [104] extracted gait information from multiple binary silhouettes of pedestrians and used this information to identify particular people. Kobayashi and Otsu (2006) [101] used spatio-temporal filters to extract gait information for human identification. Benedek *et al.* (2005) [25] turned gait analysis on its head, using simple gait information to match multi-camera views of the same person and thus build a multi-camera surveillance system.

Surveillance approaches have been extensively reviewed in Moeslund and Granum (2001) [123] and Aggarwal and Cai (1999) [2].

## 2.6 Human Visual System

Aggarwal noted in 1988 that “There are two groups of scientists studying vision. One group is studying human/animal vision with the goal of understanding the operation of biological vision systems... [The other] includes computer scientists and engineers... with the objective of developing vision systems.” Farah’s 2000 book [64] traced the current state of knowledge of the HVS. Visual information is initially collected and preliminarily processed by the retina. It is then passed on to the Lateral Geniculate Nucleus (LGN) where information for two eyes is combined. Subsequently, this data is transmitted to the Visual Cortex.

The Visual Cortex is composed of five subregions, V1 through V5, which are complexly interconnected. Sekuler *et al.* (2003) [155] gave an overview of what is known of motion processing in the Middle Temporal (MT) area which lies near V5. Although much motion processing has been resolved to the MT area, it is also known that there are neurons in all regions of the HVS which are sensitive to combinations of particular features and motion.

Overall, motion detection is a direct experience, uniquely specified by the visual system. This can be seen most clearly in the ‘blindsight’ condition, where a patient with neurological damage may be blind to stationary forms and objects but may still be aware of motion, while being unable to identify the source. Humans have been shown to be better at detecting relative rather than absolute motion. Other interesting points include MT neurons sensitive to changes in object direction. Werthein (1994) [182] reviewed ego-motion

in the HVS and the need for a reference signal (i.e. assumed fixed points in image space) as opposed to extraretinal signals (a non-image knowledge of motion and eye position).

Friston and Büchel (2000) [69] discussed the processing connections between early V2 and the suspected site of motion processing in V5/MT area. Britten and Heuer (1999) [35] explored what happens when one motion sensitive cell in the MT region is faced with overlapping motion data for two objects. Priebe and Lisberger (2004) [149] explored how visual tracking is related to the contrast of the object. Unsurprisingly, higher contrast objects are easier to track. Bair *et al.* (2001) [14] studied the connection between MT response and stimulus duration. Thornton *et al.* (2002) [169] reported on a study for biological motion using light point displays and discovered that recognition is highly attention dependent – suggesting high level processing.

Most of our understanding of the HVS is derived from experiments on animals. Shadlen and Newsome (1996) [158] presented a study on how the monkey visual system detects and tracks small moving dots. Dittrich and Lea (2001) [54] provided a very accessible introduction to motion processing in birds, noting that many birds have the ability to define and distinguish patterns and objects using only motion information. The oft-repeated ability of hawks to spot their prey at great distances is only true if the prey is in motion. Conversely, that most small creatures have a ‘freezing’ instinct in case of danger shows that motion is a key detection and recognition cue.

Li (2002) [108] discussed the need for on-retina visual bandwidth reduction, noting that the retina strips perhaps 80% of redundant information

before transmission to the optic nerve. Nadenau *et al.* (2000) [127] and Momiji (2003) [125, 124] provided detailed reviews of how models of the HVS and retina have been used to optimise image compression. Martin and Aggarwal (1978) [119] also proposed a dual-channel approach whereby the *peripheral* region detects motion and directs an *attentive* tracker which incorporates object detail. Simoncelli and Heeger (1998) [160] presented a model of the MT area which is believed to be the motion processing center of the HVS.

Grossberg *et al.* (2001) [79] noted that visual motion perception requires the solution of two competing problems of ‘motion integration’ and ‘motion segmentation’: “The former joins nearby motion signals into a single object, while the latter keeps them separate as belonging to different objects.” Grossberg suggests a neural model to explain this ability.

Neri *et al.* (1998) [131] attempted to explain the HVS’s remarkable ability to detect ‘biological motion’, i.e. to distinguish the cyclical motion of living things from that of rigid inanimate objects. Giese and Poggio (2003) [75] model biological motion recognition using a dual form and motion channel processing architecture. They conclude with a number of open questions: “*How is the information from the two pathways combined?*”, “*Does form or optic flow-based recognition dominate for certain stimulus classes?*”

Heeger and Simoncelli (1992) [82] presented a mathematical model of how they believe the HVS uses *s-t* filters to compute optical flow. Young *et al.* (2001) [192, 193] gave a detailed exploration of *s-t* filters in the visual cortex and reports on the Difference-of-Offset-Gaussians (DoOG) response

of some neurons. He noted that “Next to the simple detection of light and dark, the ability to see motion may be the oldest and most basic of visual capabilities.” Most research points to a *hardwired* motion detection system centered on *s-t* cells in the MT area. However, there are dissenting views. Sengpiel (2006) [157] reported evidence which suggests that even low level motion must be learned and is not innate.

In 1942 anatomist Gordon Lynn Walls observed “If asked what aspect of vision means the most to them, a watchmaker may answer ‘acuity’, a night flier ‘sensitivity’, and an artist ‘colour’. But to animals which invented the vertebrate eye and hold the patents on most features of the human model, the visual registration of motion was of the greatest importance.” [155]

## 2.7 Summary

Video surveillance has a very practical aim: to automate the task of detecting and tracking objects and to derive specific information from these movements and behaviours. The specific information required depends on the application, and ranges from automatic detection of crashes and jams in traffic monitoring to crime detection in public areas.

Visual surveillance systems can be divided into four constituent parts or stages that exist in varying proportions in every reported system:

- MD – Motion Detection
- LP – Location Prediction (or search constraints)

- AM – Appearance Matching and modelling
- BA – Behaviour Analysis

Two principal paradigms also exist:

- $1D+2$  – Background modeling, where motion detection takes precedence over LP and AM.
- $2D+1$  – Foreground tracking methods (i.e. particle filtering), where emphasis is placed on LP and AM. Here MD might only be used for bootstrapping or may not be present at all.

Either paradigm, if successful, simply outputs the location and path of the object. The final behaviour analysis step must either rely solely on this path information, as many do, or process the video data again to extract extra information.

The Human Visual System incorporates a strong spatio-temporal motion detection stage followed by, and integrated with, highly robust location prediction and appearance modeling. However, there is evidence that the HVS tackles the final surveillance stage using a rich data field extracted directly from the initial detection stage. ‘Biological motion’ has been shown to be detected at this initial stage, rather than by some post-tracking processing. Troscianko *et al.* (2004) [171] studied the reactions of human CCTV operators and showed that their ability to detect crime or suspicious behaviour had little connection with the path taken by the target, but focused mainly on the intra-body movements and pose.

The following chapters present new approaches to the stages of video surveillance. Chapter 3 describes a motion detection technique, called Motion Distillation, that follows a *3D* paradigm. Chapter 4 covers the LP and AM stages and describes a post detection tracking system which emulates the dual-channel form-motion scheme of the HVS. Chapter 5 describes a BA technique which uses motion information produced by the MD stage.

## Chapter 3

# Motion Detection

Motion detection is a component of many tracking systems. There are broadly three categories of motion detection:

1. *Pixel or feature optical flow approach.*

Every instance of a chosen feature type is detected in each frame and matched to the next. Moving objects are detected as connected regions of moving features. As frames will normally contain many features of very similar appearance, this is possible only by using constraint models to reduce ambiguity. These techniques are not well suited for surveillance applications because objects are small with respect to the scene and there are many motion discontinuities [120].

2. *Pixelwise statistical background modelling and subtraction.*

Here we suppose a model of the stationary scene (background) is being prepared. Regions of the current frame that are significantly different



from this background are tracked as objects. When calculating the background model, each pixel is computed separately, using the statistical assumption that the most common value is that of the stationary background. The choice of the particular statistical method used is where the various published methods differ. This choice depends on the application, the noise properties of the video, and the available computational resources.

### 3. *Motion 'Distillation'*.

Spatio-temporal edge-based detection. The subject of this work: see particularly Section 3.3 below.

## 3.1 Filtering for noise reduction

Perhaps the most basic task in signal processing is filtering for noise reduction. In general terms this can be explained in terms of the result of applying a point spread function  $g$  to all points of a function  $f$  and accumulating the contribution at every point – a process known as convolution:

$$f * g = \int_{-\infty}^{\infty} f(u)g(x - u)du \quad (3.1)$$

For a discrete 1D signal of length  $k_{max}$ , this becomes in practice:

$$F(t) = f * g = \sum_k^{k_{max}} f(k)g(t - k) \quad (3.2)$$

For 2D discrete spatial images, of width and height  $i_{max}$  and  $j_{max}$ :

$$\begin{aligned} F(x, y) &= f(x, y) * g(x, y) \\ &= \sum_i^{i_{max}} \sum_j^{j_{max}} f(i, j) g(x - i, y - j) \end{aligned} \quad (3.3)$$

Video can be viewed as a form of 3D, spatio-temporal signal. Convolution can be applied using the form:

$$\begin{aligned} F(x, y, t) &= f(x, y, t) * g(x, y, t) \\ &= \sum_i^{i_{max}} \sum_j^{j_{max}} \sum_k^{k_{max}} f(i, j, k) g(x - i, y - j, t - k) \end{aligned} \quad (3.4)$$

The nature of the  $g(x, y, t)$  function defines how the data will be effected. Choices include the filter size, whether it is high or low pass, symmetrical, smooth, Gaussian, etc.

In image processing, filters are commonly used and easily understood. An example of a square low-pass averaging function in 2D with a width of 3 is:

$$g(x, y) = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (3.5)$$

This filter will blur regions of high frequency in the data. For example, a single isolated noise pixel will be smoothed, but neighbouring pixels will become 'contaminated'. The usefulness of such an operation depends on the noise and signal qualities of the data. White Gaussian noise will be

removed by this filter without disrupting the signal. However, the important structures in most images involve high frequency components, such as edges.

Kalivas and Sawchuk (1990) studied the application of 3D, spatio-temporal filters for video noise reduction. They noted that a mean, low pass filter applied using equation 3.4 will blur moving edges over time.

A common alternative to the mean (while not a convolution) is a median filter. The median (discussed in more detail below) is the mid value of the ordered distribution. This allows outliers to be excluded, thereby removing noise points. 2D median filtering does not blur an image but does remove some fine detail, resulting in a 'softened' appearance.

Mode filters are also sometimes used. There are difficulties involved in calculating the mode accurately because of the small number of discrete samples generated by common filter sizes of  $3 \times 3$  or  $5 \times 5$  pixels. This means that instead of the smooth distribution whose mode is easily located, we are presented with a multimodal distribution whose highest point does not indicate the position of the *underlying* mode. This is discussed in detail by Davies (2005) [49].

## 3.2 Background modelling

In noise reduction, the current pixel is replaced with the 'central tendency' value, either mean, median or mode. To detect motion the aim is reversed. Now it is assumed that the central tendency is the background and the

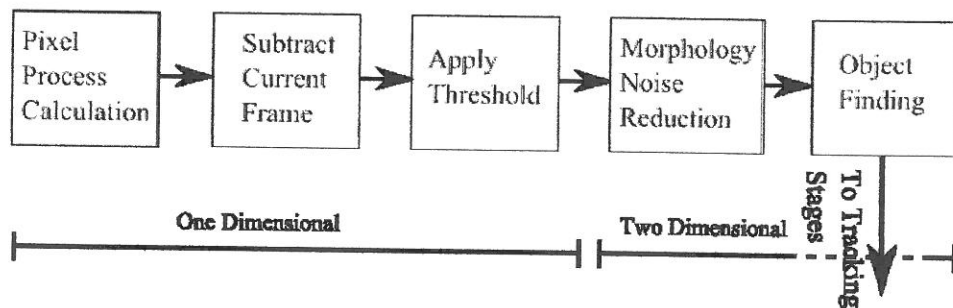


Figure 3.1: The logical scheme of Motion Detection by background modelling methods. The first three steps are 1D pixelwise processes. 'Pixel Process' refers to the chosen statistical model, such as Median filtering or Gaussian Mixture Model. The final two steps are 2D. After Motion Detection is completed, the system moves on to tracking and behaviour analysis stages.

important value is the outlier, as this may be due to either a moving object or or noise.

Background modelling starts with the computation for each pixel of the central tendency of the 'pixel process', followed by its subtraction from the current pixel value. If the difference is greater than some threshold the pixel is assigned a 'true', otherwise a 'false', in a binary detection mask. In contrast to the spatial and spatio-temporal analysis described above, background modelling only deals with 1D statistics. This technique is based on the following assumptions: that the moving object is small, with respect to the scene; that the object moves quickly, with respect to the filter size or 'temporal window'; and that the background is non-moving.

Background modelling schemes face two inherent and antagonistic problems, called the 'Stationary Background Problem' and the 'Transient Background Problem'. First, the background model must reflect the stationary

part of the scene to allow accurate segmentation of moving objects. This problem requires a low difference threshold in the subtraction stage and a large filter size so that slowly moving objects do not merge with the background. Conflicting with this is that the background model must update to appearance changes in the scene, such as changed lighting conditions, requiring a smaller filter size.

No compromise gives perfect results, with a common failure being partial segmentation when the 'object depth' (object overlap in previous frames, equal to length/speed) is greater than half the filter size. Similarly, a tracked object that then stops, such as some abandoned luggage, will merge with the background, vanishing from the tracking stage. For a background model to successfully segment a moving object:

$$\text{filter size} > 2 \frac{\text{width}}{\text{speed}} \quad (3.6)$$

where filter size is measured in frames, width in pixels and speed in pixels per frame. This evaluation is frame rate and resolution dependent. The choice of the central tendency statistic (temporal mean, temporal median or temporal mode filtering), depends on knowledge of the lighting and scene characteristics. The mean can give acceptable results in indoor environments with constant and diffuse lighting whereas outdoor 'open-world' scenes require the greater robustness to lighting fluctuations offered by the median and the mode. The Gaussian Mixture Model approach has become widely used as it has the advantages of allowing multimodal pixel processes, and thus a quick recovery time, and also a wide temporal window for robustness.

The temporal mean filter has an advantage of speed and simplicity. It is sometimes used in indoor surveillance application where lighting is relatively constant, and where a motion free bootstrap is available. If there are moving objects present during the bootstrap sequence these will cause large blurry regions in the background model.

### 3.2.1 Median filter

Mathematically, the median is described in terms of a bijective map,  $X \rightarrow Y$ , such that  $Y_1 < Y_2 < \dots < Y_N$ . The median is calculated by:

$$\text{Median}(X) = \tilde{x} = \begin{cases} Y_{(N+1)/2} & \text{if } N \text{ is odd} \\ \frac{1}{2}(Y_{\frac{N}{2}} + Y_{1+\frac{N}{2}}) & \text{if } N \text{ is even} \end{cases} \quad (3.7)$$

To implement this method, a histogram of pixel values is prepared. For a neighbourhood with  $n$  pixels, the median value lies with  $\frac{n}{2}$  on either side. The median filter removes outliers with much less blurring, but at greatly increased computational cost.

Median background modelling is based on 1D median statistics. At each pixel location the pixel values over the temporal window (the pixel columns represented in Figure 3.2) are placed in a histogram (256 levels for 8 bit pixels). As discussed earlier, there is no temporal window size that perfectly solves the Stationary Background Problem and the Transient Background Problem. By experiment I determined that a value of between 20 and 40

frames gave best results for the videos tested (see Appendix A). This number is not critical but video frame rate and content are important factors.

The background model is constructed from the median values of each pixel. These median values are compared to the current frame using a difference threshold to create a binary 'motion mask' (see Section 3.2.2). This must then be analysed using an object labelling algorithm (see Appendix section B) and a tracking algorithm (see Chapter 4).

This version of the temporal median filter was implemented for later performance comparison with the spatio-temporal filtering method presented below.

### 3.2.2 Thresholding and video noise

After the background model has been prepared it is subtracted from the current scene. The task then becomes to decide, by means of a difference threshold, which parts of the difference map are due to noise and which are due to motion. From a signal processing approach, each pixel can be viewed as being composed of the static background value ('carrier wave'), on top of which is placed noise and the motion signal we wish to detect. The signal qualities and noise levels will be quite different for pixels in different parts of the image, due largely to lighting effects. Further, the strength of the motion signal depends not on motion, but on the intensity contrast between the background and that part of the moving object passing over the pixel. However, for a given pixel, at a given time, the presence of a motion signal



is a rare event and there is no other source of information on whether there is motion or not.

The first step to considering automatic threshold selection is to examine the noise distribution of the signal. Rosin (1997) modeled image difference noise using a Rayleigh distribution, but conceded that a Gaussian centered on zero was an acceptable approximation.

Rosin showed that the probability of incorrectly classifying a pixel as motion, for a given threshold  $\tau = x\sigma$  is:

$$P = \operatorname{erfc}\left(\frac{x}{\sqrt{2}\sigma}\right) \quad (3.8)$$

where 'erfc' is the complementary error function. Coupled with an experimental knowledge of  $\sigma$ , equation 3.8 allows a choice of threshold for a given proportion of false positives.

Another approach is to rely on the assumption that there should be only a small number of large moving objects in the scene. Here, the full range of threshold levels is tried and the number of motion regions is counted. It is also assumed that true objects will remain stable over a range of threshold levels. Rosin describes this as a 'clumpiness' test. This causes problems with noisy videos as the constant region may not exist.

Similar results can be achieved more directly by setting a constant threshold value and using morphological dilation and erosion to join separated regions of the same object. An object size threshold can be applied to remove scattered noise points at the object labelling stage.



### 3.2.3 Gaussian Mixture Model

Stauffer and Grimson (1999) realised that pixel processes are often multi-modal in nature. This can be due to a static object, such as a CRT screen, flickering, or a small object oscillating over the pixel and interrupting its view of the static background. They also wished to solve the problems of adaptive and non-global threshold selection. Their solution is to model each pixel process as a mixture of Gaussian distributions in colour space. They describe the probability of observing the current pixel intensity value as:

$$P(X_t) = \sum_{i=1}^k \omega_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \quad (3.9)$$

where  $k$  is the number of Gaussians chosen for the distribution,  $\eta$  is the Gaussian pdf,  $\mu$  is the central mean point of each Gaussian and  $\Sigma$  is the covariance matrix of each Gaussian. The covariance matrix,  $\Sigma_{K,t} = \sigma_K^2 I$ , is simplified to save computation using an assumption that each colour channel is independent and has the same variance. Equation 3.9 can be solved using a k-means algorithm (among other approaches), solving first equation 3.10 followed by equation 3.11:

$$\mu_t = (1 - \rho)\mu_{t-1} + \rho X_t \quad (3.10)$$

$$\sigma_t^2 = (1 - \rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T(X_t - \mu_t) \quad (3.11)$$

where  $\rho$  is a learning rate. The differencing threshold is automatically determined for each pixel using the standard deviation of the Gaussian distri-

bution. If the current pixel value is more than 2.5 standard deviations from any Gaussian mean it is recorded as foreground.

This version of the GMM algorithm was implemented for later performance comparison with the spatio-temporal filtering method presented below. While the GMM method does give improved background modelling results over median filtering, these come at extra computational cost due to the algorithm's complexity. In operation it was noted that sometimes the EM algorithm for a particular pixel would become unstable, causing the Gaussian pdf to collapse to a point. It was not possible to determine whether this problem was due to some error in implementation or some property of the data. An *ad hoc* solution of re-initialisation of these pixels was applied.

### 3.2.4 Criticism of background modelling

Figure 3.1 provides a general background modelling scheme. An often overlooked, yet vital and expensive, stage of background modelling and subtraction follows the production of the binary foreground map, but precedes the tracking stage. Here it is necessary to search the foreground map for objects using connectivity and to eliminate 'small' objects as probable noise. It is very common that pixel-wise detection methods will result in 'holes' or incorrect splits in the object silhouette, where parts of the object are similar in colour or intensity to the background. This may be tackled using a series of morphological dilation and erosion steps on the foreground mask.

The background modelling approach can be thought of as an indirect

route to spatio-temporal motion detection. The first step is a 1D pixel-wise statistical process. This is followed by a 2D object detection stage that uses morphology. The 1D step can only detect *change* rather than true motion. The reason for this can be seen in Figure 3.2: when only one pixel is considered it is impossible to distinguish the case of localised random noise from the case of an object moving across the background. Only reference to the 2D data can allow this distinction. (Section 3.4.2 and Figure 3.16 illustrate this point from a different perspective.)

### 3.3 Motion Distillation

Motion Distillation is the name I have given to direct motion detection for visual tracking. This method uses spatio-temporal edge detecting filters to achieve this goal with greater robustness to noise and lower computational costs.

#### 3.3.1 Temporal edge detection

Can we detect motion directly, without statistics? Video is usually thought of as a sequence of frames, as with a traditional roll of film. A different approach is to model video as a 3D spatio-temporal structure, notionally 'stacking the frames' into an  $x$ - $y$ - $t$  column as in Figure 3.3. This reveals some interesting and useful aspects. Stationary image features will form straight lines parallel to the  $t$ -axis, while features in motion will form lines

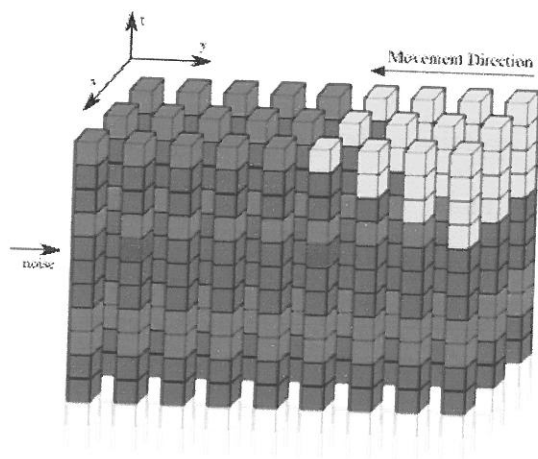


Figure 3.2: Video viewed as 1D pixel statistics. On a pixelwise basis it is impossible to distinguish the isolated noise points (left) from a spatially coherent motion (right). Only subsequent 2D processes can achieve this. An alternative approach is to apply 3D, spatio-temporal motion detection from the outset.

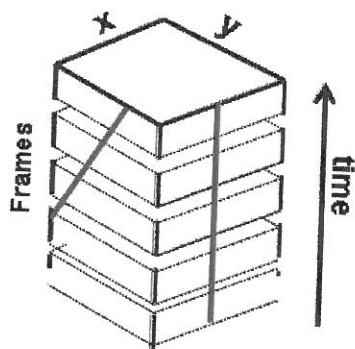


Figure 3.3: The result of 'stacking the frames' into a  $x$ - $y$ - $t$  column or 'video cube'. In this view, edges parallel to the  $t$  axis are stationary while edges with a non-parallel component are in motion. A spatio-temporal edge detector tuned to detect these edges will thus also detect motion.

with a non-parallel component.

An ordinary edge detector can be used to enhance these 'temporal' edges. In the past, several researchers have reported a variation of simple image differencing where the difference of edge-maps is taken. This approach has improved illumination invariance but has disadvantages when non-global threshold levels are required [152].

A slightly different approach is to use a Sobel or other edge detector in the  $x-t$  and  $y-t$  planes. Using a filter orientated perpendicular to the  $t$ -axis highlights features in motion. Use of a  $3 \times 3$  edge detector on a surveillance video in this fashion quickly extracts the moving objects in a robust way [167], see Figure 3.5. A threshold level is required to binarise the output for blob extraction and tracking. The histograms in Figure 3.5 compare the outputs of image differencing and the Sobel method. With image differencing, as with all pixelwise statistical methods, the output histogram shows that pixel changes due to noise and motion are grouped together in a large peak near the zero point; the spatial properties of the Sobel give greater weight to pixel change due to motion, resulting in a peak at a much larger value. A threshold level may be chosen robustly (in the sense that the threshold is far from being critical) in the large gap between the noise peak centered near zero and the motion peak at larger values, while for image differencing, morphology and size filtering will be required to extract the target.

### 3.3.2 Wavelet decomposition

Rather than use two 2D edge detectors, we can construct a single 3D spatio-temporal filter. The wavelet transform is a particularly useful mathematical technique for analysing signals which can be described as aperiodic, noisy and discontinuous. The transform is composed of two parts, the 'scaling' function low-pass filter and the 'wavelet' function high-pass filter. By repeated application of the scaling and wavelet filters, the transform can simultaneously examine the signal in both position and frequency. In its continuous, general form the wavelet function is represented by:

$$\Psi_{a,b}(t) = \frac{1}{\sqrt{a}} \Psi \left( \frac{t-b}{a} \right) \quad (3.12)$$

where  $a$  represents the wavelet scale and  $b$  the location in the signal. This may be translated into a discrete wavelet transform by setting a discrete bounding *frame* and using a type of logarithmic scaling known as a *dyadic grid*. The result is a general discrete scaling function  $\phi$ , and a wavelet function  $\psi$ , each in turn dependent on the separately chosen *mother wavelet*  $\Psi$ .

The choice of  $\Psi$  depends on the application. Commonly used functions include the Gaussian, the 'Mexican Hat' function (a second derivative Gaussian) and the series of Daubechies wavelets. The JPEG 2000 standard, for example, uses one of the Daubechies wavelets in a 2D fashion to separate images into a series of feature maps at different scales, which may then be efficiently encoded and compressed.

One way of viewing the behaviour of the wavelet transform is as an edge



Figure 3.4: Shows the scale pyramid output of wavelet image decomposition. The top row represents the output of the horizontal wavelet, the side column of the vertical wavelet and the diagonal of the diagonal wavelet. The image in the top left column is the result of the final scaling function.

detector [1], because a high-pass filter detects discontinuities. To decompose a 2D signal such as an image, 1D low-pass filter  $\phi$  and high-pass filter  $\psi$  are combined using the tensor product to produce four filters, each sensitive to edge features at different orientations.

$$\begin{aligned}
 \text{2-D scaling fn: } & \phi(x, y) = \phi(x)\phi(y) \\
 \text{2-D horizontal wavelet: } & \psi_h(x, y) = \phi(x)\psi(y) \\
 \text{2-D vertical wavelet: } & \psi_v(x, y) = \psi(x)\phi(y) \\
 \text{2-D diagonal wavelet: } & \psi_d(x, y) = \psi(x)\psi(y)
 \end{aligned} \tag{3.13}$$

These filters are applied sequentially to the image. First, the three orientated wavelet filters are used to extract features at the highest detail scale. Next,



the scaling filter is used to reduce the scale of the image, followed by another pass of the wavelet filters. This is repeated to the desired scale, producing a 'feature pyramid' (Figure 3.4).

This system can be extended to 3D spatio-temporal wavelet decomposition using tensor products. The result is eight spatio-temporal filters, ranging from the  $s$ - $t$  scaling function  $\phi(x)\phi(y)\phi(t)$  to the  $s$ - $t$  diagonal wavelet,  $\psi(x)\psi(y)\psi(t)$ .

### 3.3.3 The spatio-temporal Haar wavelet

To apply these wavelets to real data, a specific mother wavelet function must be chosen. The requirements of the mother wavelet are application-specific while bearing in mind practical considerations such as computational cost. In this case, the goal is not to encode or record the image data, but merely to detect temporal discontinuities at particular  $s$ - $t$  orientations. The widely used Daubechies category of wavelet has the ability to detect polynomial signal patterns. The wavelet order is linked to the polynomial order to be detected. Daubechies 4 (D4) detects second-order polynomials. D6 is sensitive to third-order polynomials, and so on. The higher the wavelet order, the greater the computational cost of implementation. The simplest Daubechies wavelet is D2, and is also known as the Haar wavelet. Here we will demonstrate that this simplest wavelet, extended to three spatio-temporal dimensions, is a very powerful and efficient motion-detection tool.

The 1D Haar mother wavelet is composed of just two coefficients. Tensor



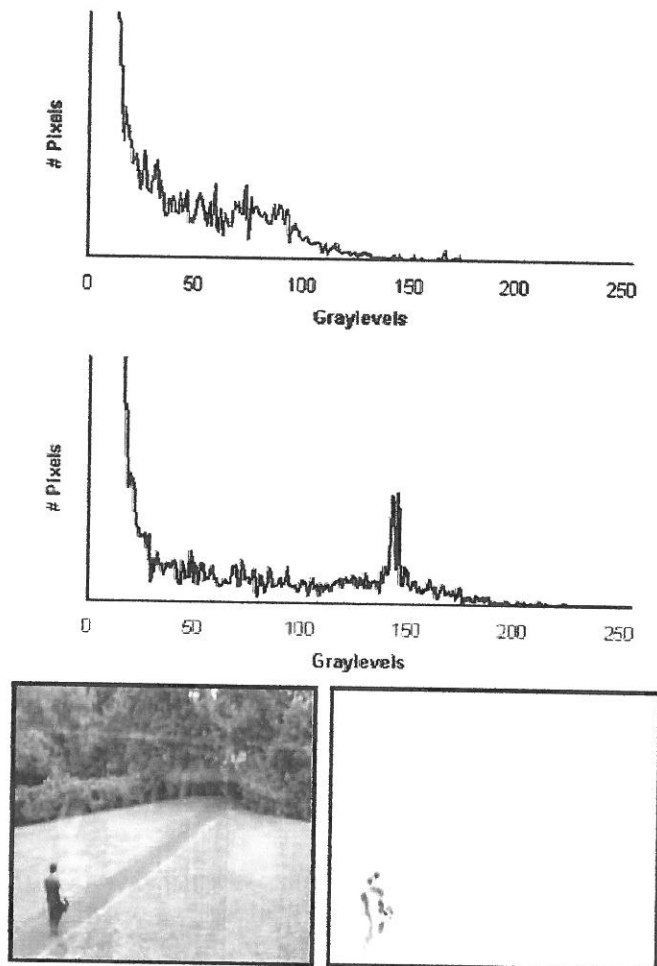


Figure 3.5: Top – the image difference histogram. Noise and motion pixels grouped together. Lower histogram – Sobel edge detection histogram. Motion pixels are shifted out and can be easily detected. Bottom – An example of motion detection using Sobel temporal edge detection.

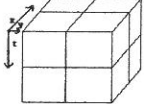
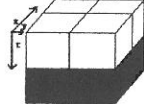
	Scaling fn	Wavelet fn
1D	$[1 \ 1]$	$[1 \ -1]$
2D	$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$
3D		

Figure 3.6: Table showing the coefficients for the Haar Wavelet Transform for different dimensions. For 3D, white is used to represent a value of '1' and black '-1'. Wavelet decomposition is carried out by sequentially convolving the signal data with first the scaling fn, and then the wavelet fn for each desired scale level.

products of 1D wavelets can produce a series of 2D and 3D scaling functions and wavelet functions. The 3D row of Figure 3.6 shows an example of two of the eight spatio-temporal Haar wavelets, where black represents '-1' and white '+1'. This wavelet function has its step-function orientated perpendicular to the  $t$ -axis.

Spatio-temporal wavelet decomposition of a signal produces a feature pyramid as with 2D. The input signal is convolved with both the scaling filter and the wavelet filter. The output of the scaling filter is reduced by a factor of two in each dimension. This output is again convolved with the scaling and wavelet filters.

The output of each wavelet convolution is a product of the speed and contrast of the edge feature. The equation for computing a single wavelet

filter is:

$$W = \sum_{t=T_0}^{T_1} \sum_{(i,j) \in W} x_{tij} - \sum_{t=T_1}^{T_2} \sum_{(i,j) \in W} x_{tij} \quad (3.14)$$

where  $W$  represents the filter size in  $i$  and  $j$  dimensions,  $x_{tij}$  represents the video pixel data at the point in spatio-temporal space  $(t, i, j)$ .  $T_0$  and  $T_2$  are the lower and upper bounds of the filter respectively along the time axis, and  $T_1$  is the position of the discontinuity. If part  $T_0 \leq t < T_1$  is equal to part  $T_1 \leq t < T_2$  then the wavelet output  $W$ , will be zero. This condition may occur if there is no movement within the data analysed, i.e. if the  $s$ - $t$  orientation of any edge is parallel to the  $t$ -axis. The filter output,  $W$ , is approximately proportional to<sup>1</sup>:

$$\left(\frac{\pi}{2} - \theta\right) \times \text{edge contrast} \quad (3.15)$$

where  $\theta$  is angle of edge to the spatial plane. Because of the fact that at a local level it is impossible to know whether a dark object is moving against a light background, or visa versa, motion direction information is ambiguous. For detection, where only binary motion data is required,  $W$  can be normalised to remove this ambiguity. In Chapter 5 we use the raw motion information for behaviour classification.

### 3.3.4 Detection comparison

Table 3.7 details a comparison of segmentation results for several videos using three motion-detection methods – two traditional background modelling

<sup>1</sup>A more exact form is developed in Section 3.4.2

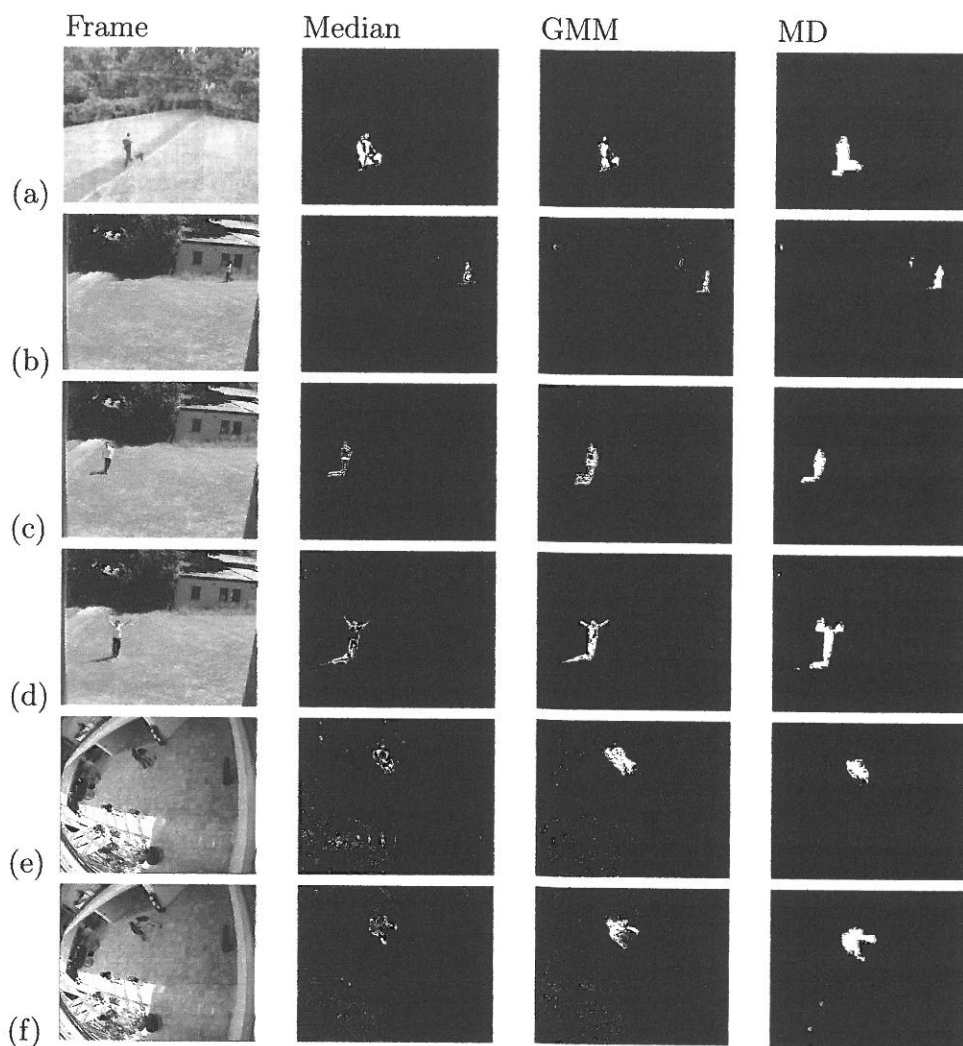


Figure 3.7: Figure shows comparisons of thresholded detection results for Haar wavelet and traditional background modelling. Three videos with different noise qualities are analysed. The first (a) low noise, is captured using a webcam on an overcast day. The second (b, c, d) is a long higher resolution camcorder video on a sunny day with changing lighting and shadows. The target pedestrian walks and runs in different directions. Frame 5187 (d) shows the pedestrian walking slowly towards the camera and waving his hands. The final video (e, f) is taken from the CAVIAR project and has considerably higher lighting noise than the other two. Note: MD segmentation results shown use edge zoom discussed in Section 3.3.6.

techniques, median filtering and GMM, and the  $s$ - $t$  Haar method. For comparison, we have chosen the temporal median filter because it has comparable algorithmic complexity to our own method; it may be set up to operate over a small number of frames and it has a simple bootstrap. The GMM approach is chosen as it is an accepted and particularly widely used background modelling method. Here both the median filter and the new method are computed over eight frames (the new method is computed to the third decomposition level, for which the number of frames is  $2^3$ ), and both also have an 8-frame bootstrap. The GMM requires a larger number of frames in order for the Gaussians to stabilise on a particular distribution, and in our implementation we use a 20-frame bootstrap, although this number is not critical. The output from all three methods is presented without any subsequent morphological or noise reduction steps, so as to be sure of comparing like with like. (Naturally, any method can be enhanced to improve performance, but here we focus on *intrinsic* performance for clarity.)

The videos are surveillance style and have been chosen for their differing degrees of noise, characterised by the median value of pixel variance over time, and they depict behaviours of pedestrian targets. The first case presented here (Figure 3.7, Video 1) is a simple motion segmentation task of an outdoor scene with diffuse lighting and low noise. The median pixel intensity variance is 1.65. The median filter results clearly show the difficulties of that method. The target is incorrectly segmented with leading and trailing edges separated. This problem is due to the slow speed of the target with respect to the temporal window size of 8 frames; the middle of the target has been absorbed

into the background model. Both the GMM and new method give more accurate results.

Video 2 in Figure 3.7 shows a variety of target pedestrian behaviours. The median pixel variance is 3.05. In frame 664, the target is walking directly across the frame. There are slight shadows that interfere with segmentation and more random image noise than Video 1. This noise is shown clearly in both median and GMM background subtraction results because these methods behave as change detectors. Pixels are segmented if the contrast with the background model is above some threshold. (In median filtering this threshold is global; in GMM it is pixelwise and adaptive.) Morphological closing (which would have to be anisotropic, and would to a fair extent be an *ad hoc* measure) could improve the background subtraction results, but this somewhat expensive step is unnecessary for the new method. However, it is commonly necessary for many implementations, such as [164].

The random, structureless nature of the video noise means that the edge detector of the new method reacts less strongly, and is automatically removed by the scaling process. In this frame there are also two other small regions of motion. Median filtering detects only one of these, while the GMM catches both, but in neither of these cases is a strong signal obtained. However, neither method is capable of cleanly distinguishing these objects from noise and it is likely that subsequent noise reduction steps will remove them entirely. The new method clearly detects both small moving regions while robustly eliminating all noise.

Frame 5136 is of a target pedestrian walking slowly towards the camera

– a case rarely dealt with in the background modelling literature. Because of the slow motion relative to the image frame, the centre of the target becomes absorbed into the background (to a large degree in the median filter, but less so in the GMM), and the background subtraction results show large gaps. In Frame 5187 the pedestrian is waving his arms; this is discussed further in the future work section below. The new method results for Frame 5187 show slight smearing of the arms because of their rapid motion, which is a characteristic by-product of this method. (It must be emphasised that motion analysis is necessarily carried out over time, and thus refers to a range of positions: the complete picture at any moment can therefore only be ascertained by combining form and motion information.) Again, the new method also detects a small moving object in the top left corner which is completely missed by both the median and GMM methods.

The final frames are from the “fights & runs away 1” sequence in the CAVIAR database. This video shows the highest degree of noise of these examples with pixel variance as high as 10 in the brightly lit bottom left quadrant of the video. Again, the new method has a cleaner response because this noise is random – and thus changing but not moving. In all cases the new method demonstrates a far greater robustness to noise than either median filtering or GMM.

Table 3.1 presents quantitative object detection results for the  $s-t$  Haar method. Object detection rates for each video were compared with the manually established ground truth. In Table 3.1, TB stands for ‘True Blobs’, which is the true number of moving objects in each frame. Blobs are formed



Table 3.1: Performance of motion channel with three videos from Figure 3.7. 'Frames' indicates the length of each test video in frames. 'TB' indicates the total number of real moving objects in each frame (ground truth) and 'DB' is the number of detected blobs. 'FP' is False Positives, objects detected where there are none. False positives are mainly due to motion clutter. 'FN' is false negatives, moving objects not detected. The precision results presented compare favourably with those for background modelling and at considerably less computational cost.

	Frames	TB	DB	FP	FN	accuracy
Video 1	538	378	380	2	0	99%
Video 2	5458	2712	2850	138	0	95%
CAVIAR	550	938	934	20	24	95.30%

by grouping together connected pixels. Any isolated pixel or isolated group of connected pixels are counted as individual blobs. If a blob exists at the location of a true object, or touches on the location of part of a true object, that is counted as a match. The size of the object is not considered here. 'FP' and 'FN' stand for 'False Positive' and 'False Negative'. FP indicates a blob (one pixel or more) was in a location where no true object was moving while FN means no blob was in the correct location. If two separate blobs are detected at a location which contains only as single true object, this is counted as a single true positive.

In test videos 1 and 2, all false positives were observed to be due to moving tree branches and all real moving objects were detected. In the CAVIAR video a small number of false positives were caused by shadows and a poster moving in air currents. The 24 false negatives were due to a single person moving slowly in a dark region of the video (though there is clearly some argument whether these false negatives should be counted as

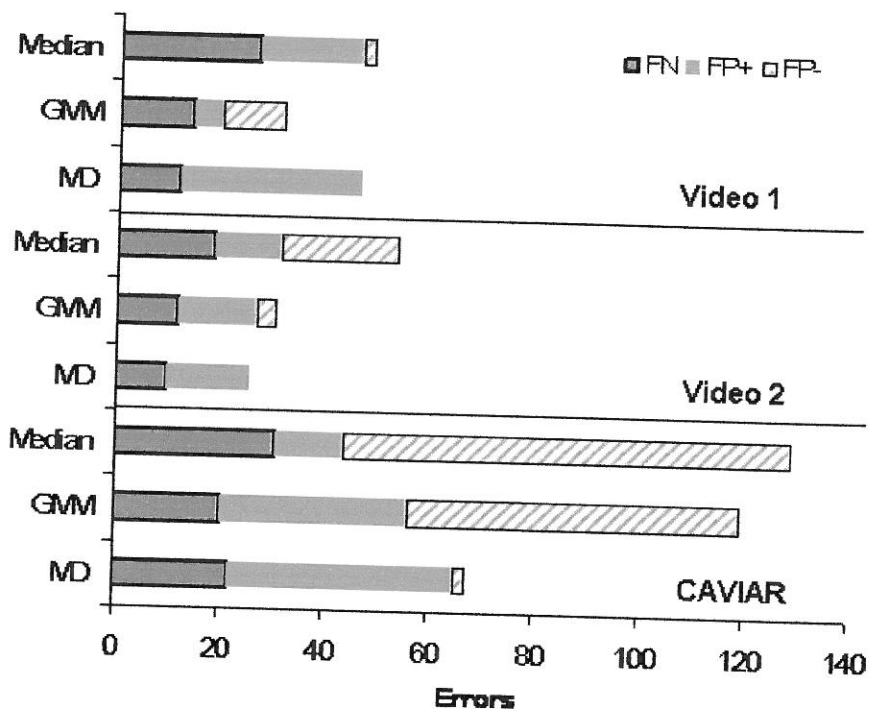


true positives).

The numerical results for the median and GMM filters would be very poor without some post-detection noise-reduction step, and are thus not fairly comparable. This comparison is better made with reference to Table 3.2, which shows pixelwise detection accuracy for the median filter, GMM and MD methods for the three videos presented in Figure 3.7. These results were prepared with a random selection of 10 frames from each video using manually groundtruthing of moving pixels. The results show three categories of error, FN (false negative) – moving pixels which were not detected, FP<sup>+</sup> (false positive plus) – nonmoving pixels falsely detected, but connected to a correctly detected blob, and FP<sup>-</sup> (false positive minus) – nonmoving pixels falsely detected and unconnected to a true moving object. The x-axis values are denominated in thousands of pixels and these values have been normalised for frame size to allow comparisons between videos.

The total error level for MD are generally lower than the comparison methods, beating median filtering in every case and GMM in 70% of cases. Of particular note is that the instance of unconnected false positives is at or near zero for MD, while connected noise points are higher. This is due to the scaling effect of the method which, while removing noise also reduces pixelwise precision.

Table 3.2: A pixelwise detection accuracy comparison for the three videos from Figure 3.7 using the Median, GMM and MD methods. The bars show the average number of error pixels for each method and video divided into three categories, FN (false negative pixels), FP<sup>+</sup> (false positive pixels connected to a correctly detected object) and on the right, FP<sup>-</sup> (false positive pixels unconnected to a true object).



### 3.3.5 Computational cost

Here we offer a precise derivation of the computational costs of the system measured in operations per pixel. For comparison, the computational requirement of the temporal median filter method of background modelling is  $\sim 256$  operations per pixel (because of the need to analyse the intensity histogram). The GMM technique is more expensive, requiring a lengthy initialisation step, followed by evaluation of an exponential function for each

Gaussian and each pixel. For this system, the temporal window concept is replaced by the idea of a decomposition level  $D$ . The number of starting frames for this decomposition level is  $2D$ . On the first iteration, two frames are convolved with two filters – the scaling function and the wavelet function – to produce one scaled frame and one motion frame (each of size  $\frac{n}{2} \times \frac{n}{2}$ , where  $n$  is the frame width). The next stage repeats this for two scaled frames from stage one, resulting in output frames of size  $\frac{n}{4} \times \frac{n}{4}$ . The third decomposition stage uses two second-level scaled frames and produces frames of size  $\frac{n}{8} \times \frac{n}{8}$ . The total number of pixels in the system is given by:

$$N = \sum_{i=0}^D \left(\frac{n}{2^i}\right)^2 \frac{2^D}{2^i} = 2^D n^2 \sum_{i=0}^D 2^{-3i} \quad (3.16)$$

For the 3D Haar Wavelet Transform, the number of operations required to decompose the signal to level  $D$  is given by equation 3.16 but with the expression summed to  $D - 1$ . The number of operations per pixel is:

$$\frac{1}{2^D n^2} \sum_{i=0}^{D-1} \left(\frac{n}{2^i}\right)^2 \frac{2^D}{2^i} = \sum_{i=0}^{D-1} 2^{-3i} \quad (3.17)$$

This has a minimum value of 1 operation per pixel when  $D = 1$  and the maximum is found using the limit as  $D$  goes to infinity:

$$\lim_{D \rightarrow \infty} \left( \sum_{i=0}^{D-1} 2^{-3i} \right) = \frac{8}{7} \simeq 1.14 \quad (3.18)$$

This measure of computational load, of less than about 1.14 operations per pixel per filter, is close to the minimum possible, and is a major speed improvement when compared with that of the median filter or other methods.

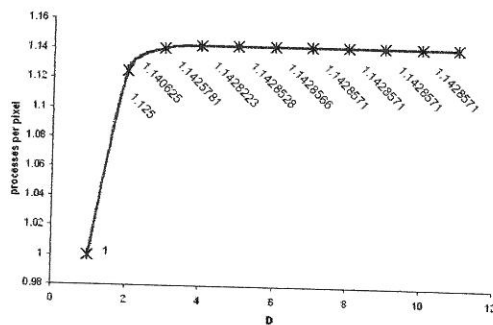


Figure 3.8: Graph of Decomposition level,  $D$ , vs. processes per pixel using the relationship in equation 3.18. This demonstrates how the computational cost of motion distillation has an upper limit. The value rises quickly towards the asymptote at  $\frac{8}{7}$ .

Figure 3.8 illustrates the relationship  $D$  vs. processes per pixel.

Frame rate comparisons of techniques are problematic because of differences in implementation, input and equipment. However, we can report that our implementation of the Motion Distillation scheme runs at 62 fps on a P4 machine, while median filtering and Gaussian Mixture Model (GMM) run approximately 10 and 80 times slower. Input frame size is  $720 \times 576$ .

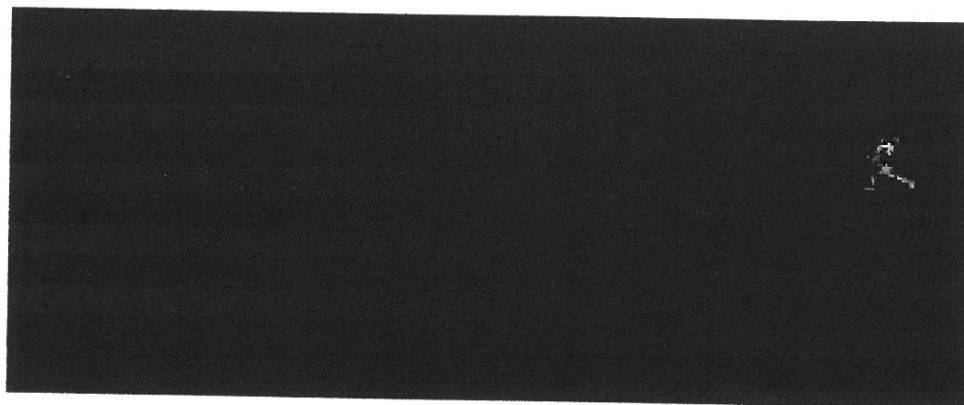
### 3.3.6 Edge zoom

A consequence of the reduction in the size of the motion map through spatio-temporal scaling (see Section 3.3.2) is imprecise segmentation (extra background pixels outside the edge of the moving object are taken with the object). This is partly due to a spatial uncertainty due to reduced resolution and partly due to temporal uncertainty, or motion blur, as discussed in Sec-

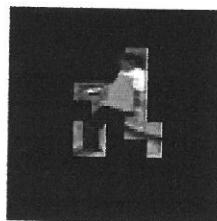
tion 3.3.4. Imperfect segmentation is not critical for the tasks of tracking and behaviour analysis, as discussed in the following chapters. However, there are times when improved segmentation is desirable.

This can be achieved by integrating the motion detection information derived during the wavelet decomposition stage. The system overlays the highest level decomposition mask over the lower level outputs. Each pixel in the higher level covers four pixels below. The silhouette boundary is refined at each stage using the higher level as a mask. The result is shown in Figure 3.9. The original image on the far left is cropped using motion information from a three-level decomposition to produce the 'low detail' image. This segmentation mask is laid over the second level wavelet decomposition output. A refined edge position is selected using thresholded motion information in the second level. This is repeated for the first level decomposition (right). The operation is computationally cheap as the motion data has already been produced and stored at the motion distillation stage and because the cropped images are small.

The question arises, why not use the first level of decomposition from the beginning? This is not used because lower levels have increased noise results. The motion signal has less noise at higher decomposition levels, but at the cost of reduced detail.



Original



High level  
Low detail



Medium level  
Medium detail



Low level  
High detail

Figure 3.9: This figure illustrates the task of integrating motion detection information from multiple decomposition levels to refine edge detail. The top figure shows the noisy output from the first decomposition level.

### 3.3.7 Shake protection

Camera shake is a frequent problem in practical CCTV systems. Shake causes the previously static background features to appear to move. This can fill the motion channel with spurious objects or, depending on video content, fill the field with one large object, which may disrupt and crash the tracker. This is a problem of all motion detection-based tracking systems.

In this implementation, we wish to detect the shaking frames and ignore them, skipping ahead to the next non-shaking frame. Occlusion reasoning

(Chapter 4) is used to recover object tracks following a shake event. Shake detection works by calculating the percentage of motion pixels in the motion mask. If this percentage is above a threshold the system discards the frame.

Figure 3.10 shows frames from two shaking videos along with the output of the motion channel. When the camera moves, all static edges in the moving background are picked up in the motion channel. The percentage of shaking motion detected depends on the number of edges in the image and, as featureless areas show no motion, the percentage does not reach 100%. The variation of line widths seen in the figure is due to feature contrast effects. When shaking stops, the system can quickly return to normal. Background modelling techniques require a longer recovery time because the interrupted pixels must pass out of the temporal window. In fact, there are really two types of camera shake: one where the camera returns to the original viewing angle and the other when the camera view is permanently changed. Background modelling techniques may have to reinitialise if the viewing angle has been permanently changed. The motion distillation approach is unaffected by a permanent change of view.

An alternative to simply discarding the frames would be to determine the direction of shake and compensate for it – eliminating the background while maintaining the real moving objects. This could be done using DoOG filters in the motion detection stage as discussed in Section 3.4.1. However, in many situations few frames are affected by shaking and the return would be minimal on this considerable computational investment. Simply discarding the frames is a more cost-effective solution.



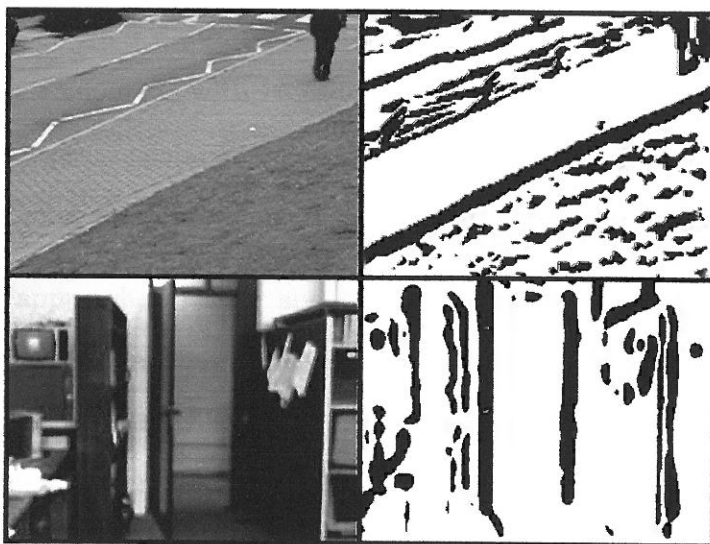


Figure 3.10: Result of shaking on motion detection. Motion masks (right) are produced from the camera shake in the videos on the left. A shake protector algorithm ignores these frames and waits for the shake to end. This protects the object finding and tracking stage, discussed in the next chapter, from overload.

### 3.3.8 Comparison to the Linköping s-t method

The system presented here provides a computationally cheap method of motion detection without attempting to generate optical flow information. In this its aim is to provide information useful for motion tracking and behaviour analysis. Only one filter (Haar) is used, which providing only one datum (colour) for each location, cannot be used alone to generate motion vector information.

In contrast, the work at Linköping University, discussed in Chapter 2, aims to generate full velocity vector and acceleration information akin to optical flow (e.g. [16]). In this task they generate a tensor representation of



video motion using a series of six or more quadrature filters. Global optical flow is then computed, with consideration of the aperture problem, from these local measurements. See also Bruhn *et al.* (2005) [39] for a discussion of the Linköping work which situates it within the optical flow literature.

For my approach, tracking and behaviour analysis, the extra data provided by the Linköping approach is not necessarily advantageous. Chapters 4 and 5 demonstrate that this single datum is adequate for object tracking and behaviour analysis, and so the method provides considerable computational efficiencies over the Linköping approach for this task.

### 3.3.9 Aperture problem

The goal of an optical flow algorithm is to output a unique vector field which describes the frame-to-frame motion apparent in a video sequence. Local, or even pixel-level, resolution is often desired [7]. Extensive research has been carried out into this area since the early work of Horn and Schunck [84]: references [113, 31, 17, 38, 39, 52, 7] exemplify this. In particular, there has been a continuous development of this topic from before 1981 right to the present day. Notable recent results include Díaz *et al.* (2006) who implement the classic Lucas and Kanade approach in real time using FPGA hardware. Bruhn *et al.* (2005) combine the Lucas/Kanade and Horn/Schunck methods into one system, and Amiaz *et al.* (2007) improve accuracy by 30% over benchmark results by adding a subpixel resolution step. However, it must be emphasised that the original intensity gradient based optical flow work

has been considerably augmented by use of higher order intensity variations, largely in the form of corner and other feature detectors. See, for example, Willis *et al.* (2006) [185].

The aperture problem refers to the difficulty of determining accurate region and object level motion information when using local ('aperture'-based) measurements of motion. The problem has been most thoroughly explored in the context of optical flow where motion is measured at the pixel level and the task is to determine a complete velocity vector field of all motion in the scene.

This task is quite a challenge as edges with intensity profiles that do not vary along the direction of motion prevent the local measurement of motion; only edges with a normal component to the direction of motion carry information about motion. In addition, there is ambiguity in the direction and magnitude of the velocity vector due to the limited aperture of local detection.

From the intensity function,  $I(x, y, t)$ , an equation for velocity field,  $\mathbf{v}(x, y)$  can be derived (see [49] for full derivation). Expanding  $I$  using Taylor series and assuming the image has shifted an amount  $(dx, dy)$  in time  $dt$ :

$$I(x + dx, y + dy, t + dt) = I(x, y, t) \quad (3.19)$$

Allowing for uncertainty due to the aperture problem, we can determine that the components of the velocity vector,  $\mathbf{v}$ , lie along the line described by:

$$I_x v_x + I_y v_y + I_t = 0 \quad (3.20)$$

and that this line is normal to the direction  $(I_x, I_y)$  and has a distance from the velocity origin equal to:

$$|\mathbf{v}| = -I_t / [I_x^2 + I_y^2]^{\frac{1}{2}} \quad (3.21)$$

Figure 3.11 shows the above equations plotted in  $v_x$ - $v_y$  space.  $I_x$  and  $I_y$  are the partial derivatives with respect to  $x$  and  $y$  of the image intensity function, and the direction  $(I_x, I_y)$  can be determined using a Sobel operator. All that is known about  $\mathbf{v}$  is that its components lie along the line described by equation 3.20. There is no way to determine  $\mathbf{v}$  uniquely using only local, gradient-only, information. (Note, however, that  $\mathbf{v}$  can be determined if higher derivatives of  $I$  are taken into account: use of corner features provides a common example of this. [185]) The most common global solution (Horn and Schunck (1981) [84]) uses a smoothness assumption and iterative relaxation labelling to arrive at a self-consistent solution which minimises global error. However, the problem of ambiguity in textureless regions remains.

The aperture problem is evident in the output of local Haar motion filters as used in this thesis. Each filter outputs a value for local speed (a scalar value, there is no direction information) which is representative of local apparent speed only. Motion at regions of zero contrast or where an edge moves parallel to the pixel grid will not be detected locally.

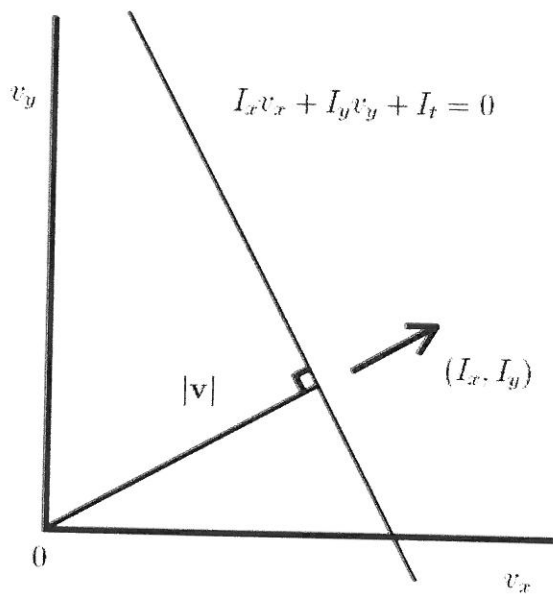


Figure 3.11: Determining the velocity vector,  $\mathbf{v}$ . It is known that  $\mathbf{v}$  must lie along a line perpendicular to  $(I_x, I_y)$  and its distance from the origin is  $|\mathbf{v}|$ . Figure from Davies 2005 [49]

In the visual surveillance application, the aperture problem will arise whenever textureless objects move rigidly parallel to the camera pixel grid. This will occur rarely for some vehicle movements. The detection and tracking results presented in this thesis were achieved without implementing a solution for the aperture problem. This demonstrates that, for surveillance applications, the aperture problem is not a serious impediment to the task of object detection and tracking. Chapter 5 presents an object level behaviour analysis method, which amounts to an application specific remedy for the aperture problem. Local motion data, derived from the output of local filter measurements, are combined at the object level into a single ratio measure. See Chapter 5 for further details.

Some applications may require that a more accurate measure of object level motion be developed to account for the aperture problem. Parts of some tracked objects will incorrectly be measured as having zero motion at the local level. This may be due to low local contrast or due to their motion being aligned with the pixel grid. As the whole object, and its motion speed and direction, are detected by the methods described in Chapter 4, this information can be efficiently fed back to the detection stage, assigning this speed information to the missing sections. However, if full optical flow information is required, standard optical flow algorithms would be necessary.

### 3.4 General spatio-temporal wavelets

The above study of Motion Distillation (Section 3.3) explores the most direct and efficient route to motion detection developed. It is a robust and extremely computationally cheap method and, as will be shown in Chapter 5, can also be used directly to achieve many behaviour analysis tasks.

Detection is sufficient for object tracking. In some applications, goals other than basic detection may be required. Below, new theoretical studies of other wavelets are presented, along with potential applications. Section 3.4.2 explores general cuboid filters, presents a mathematical study, and concludes with a practical method for speed detection independent of edge contrast.

Section 3.4.1 discusses large wavelets with a Gaussian profile and wavelets with a discontinuity plane which is rotated with respect to the time axis. Advantages of this include greater precision of detection and speed selectivity.

This method uses of multiple filters which have rotated discontinuity planes to derive optic flow information.

### 3.4.1 Difference of Offset Gaussians

Increasing the size of the filter allows a smoother filter profile, which in turn results in greater precision of detection and fine motion detail. Computational costs increase with the power of filter dimension.

Larger filters also make speed and direction sensitivity practical. The Haar filter has its discontinuity plane perpendicular to the time axis. The filter output is proportional to speed and insensitive to direction of motion (equation 3.15). Changing the angle of the discontinuity plane with respect to the time axis alters the response of the filter to particular combinations of speed and direction. For example, if the discontinuity is rotated  $+45^\circ$  about the  $y$ -axis, the filter's maximum response occurs when the detected edge is moving at one pixel per frame from left to right parallel to the  $x$ -axis. Minimum output is now when the edge moves at this speed in the opposite direction. Equation 3.15 becomes:

$$W \propto \left(\frac{\pi}{2} - \theta + \lambda_x\right) \times \text{edge contrast} \quad (3.22)$$

where  $\theta$  is the angle of the spatio-temporal edge and  $\lambda_x$  is the angle of the discontinuity plane of the filter. Both angles are measured perpendicular to the  $t$ -axis of the video data. (For example, for the cubic Haar, where the filter discontinuity is perpendicular to the  $t$ -axis,  $\lambda_x$  is zero and the above

equation reduces to equation 3.15.)

Young et al. (2001) [192, 193] propose that neurons in the HVS motion channel have a spatio-temporal Difference of Offset Gaussian (DoOG) profile. To construct this the Gaussian equation is used in each dimension:

$$f(x, \sigma_x, \mu) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma_x^2}} \quad (3.23)$$

$$F_{x,y,t} = f(x, \sigma_x, \mu) f(y, \sigma_y, \mu) f(t, \sigma_t, \mu) \quad (3.24)$$

where in this case  $\mu$  is the center point of the filter and  $\sigma$  is the filter size. In the time direction the sign of the Gaussian is reversed at the midpoint in order to form the discontinuity plane. For a DoOG profile:

$$f(t, \sigma_t, \mu) = \frac{1}{\sigma_t\sqrt{2\pi}} \left[ e^{-\frac{(t-\mu-\text{offset})^2}{2\sigma_t^2}} - e^{-\frac{(t-\mu+\text{offset})^2}{2\sigma_t^2}} \right] \quad (3.25)$$

However, for discrete filters with a relatively low number of coefficients this can be simplified with little loss of accuracy (see Figure 3.12):

$$F'_{x,y,t} = \begin{cases} +1 \times f(x, y, t) & \text{if } t > \frac{\sigma_t}{2} \\ -1 \times f(x, y, t) & \text{if } t < \frac{\sigma_t}{2} \end{cases} \quad (3.26)$$

A filter with an arbitrary discontinuity plane is constructed by rotating the coordinate system of the Gaussian function. Centering the function on zero in the coordinate system  $(x, y, t)$  the distribution is sampled at points in the coordinate system  $(x', y', t')$  which is rotated by  $(\delta, \epsilon, \gamma)$ . The angles  $\delta, \epsilon, \gamma$  represent rotations about the  $x, y, t$  axes respectively. These transforms

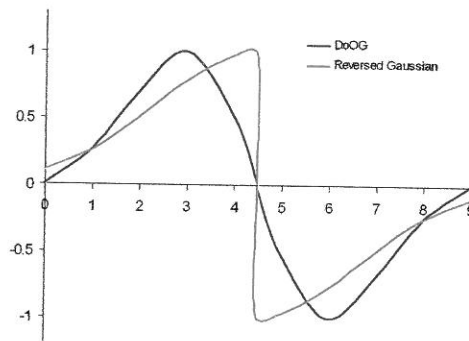


Figure 3.12: Graph compares a true Difference of Offset Gaussians profile (equation 3.25) with the reversed Gaussian profile (equation 3.26).

should be applied in the order presented below:

$$\begin{aligned}
 x' &= x(\cos \delta \cos \gamma - \sin \delta \cos \epsilon \sin \gamma) \\
 &+ y(\cos \delta \sin \gamma + \sin \delta \cos \epsilon \cos \gamma) \\
 &+ t(\sin \delta \sin \epsilon) \\
 y' &= x(-\sin \delta \cos \gamma - \cos \delta \cos \epsilon \sin \gamma) \\
 &+ y(-\sin \delta \sin \gamma + \cos \delta \cos \epsilon \cos \gamma) \\
 &+ t(\cos \delta \sin \epsilon) \\
 t' &= x(\sin \epsilon \sin \gamma) + y(-\sin \epsilon \cos \gamma) \\
 &+ t(\cos \epsilon)
 \end{aligned} \tag{3.27}$$

The constructed filter will have a maximum response to edges parallel to the discontinuity plane and zero response to edges perpendicular to it.  $\delta$  controls the speed response to motion along the  $x$  axis and  $\epsilon$  controls the speed response along the  $y$  axis.  $\gamma$  rotates the filter around the  $t$  axis and



so has no effect on detection. Filter response is also dependent on edge contrast, as shown in equation 3.15. We would like to transform these into heading ( $\phi$ ) and speed ( $\theta$ ) information and to eliminate contrast dependence. This requires combining the outputs of three filters at each point. The filters should be arranged as in Figure 3.13, with one filter aligned along each axis. Using trigonometry:

$$\phi = \arctan \frac{W_x}{W_y} \quad (3.28)$$

$$\theta = \arctan \frac{W_t}{\sqrt{W_x^2 + W_y^2}} \quad (3.29)$$

Figure 3.14 shows a video of two people moving towards each other, along with a colour-coded analysis of video using the wavelet decomposition method described above. The filters used were  $8 \times 8 \times 8$  DoOG, with sigma of 4 in each dimension. The hue of each pixel represents heading,  $\phi$ , and pixel intensity represents speed,  $\theta$ . The black lines occur when the filter is positioned between leading and trailing edges of the object. This method is very computationally expensive, involving the convolution of three  $8^3$  filters at each point. To cut down on unnecessary computation the video was first passed through an ordinary  $s-t$  Haar filter. The DoOG process was then performed only at points where motion was detected by the Haar process. This has the unintended effect of some blockiness around image boundaries.

### 3.4.2 General cuboid filters

The method described in Section 3.4.1 uses three filters, two of which have discontinuity planes that are tilted with respect to the time axis and thus

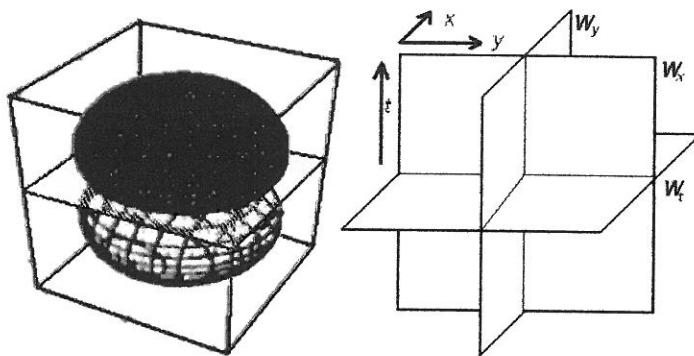


Figure 3.13: Left: Illustration of a Difference of Offset Gaussian spatio-temporal filter (modified from Young et al. (2001)[192, 193]) Right: Discontinuity planes of three DoOG filters arranged as used for determining contrast independent direction and speed detection.



Figure 3.14: Example of DoOG decomposition of video (left) showing two pedestrians walking towards each other. On the right, pixel hue represents motion direction and brightness represents speed. Each pedestrian is represented in a different colour due to their different direction of movement.

each is sensitive to a particular combination of speed and direction. This allows elimination of contrast dependence on the filter output but at the cost of using three filters.

Next, a theory of general rectangular parallelepiped, or cuboid, spatio-temporal wavelets is developed. This offers a way of eliminating contrast dependence using only two filters. This method is also insensitive to motion direction.

Cuboid filters are defined as ones where height is not necessarily equal to width, i.e. where  $t \neq x$  and  $x = y$ . Figure 3.15 shows an illustration of a general wavelet.  $t$  and  $x$  are the wavelet size in time and in the  $x$ -dimension respectively. For illustrative purposes the second spatial dimension,  $y$ , is temporarily ignored. The ratio,  $r$ , can be used to describe filter shape, such that:

$$r = \frac{t}{x} \quad (3.30)$$

'Tall' filters are described by  $r > 1$  ( $t > x$ ); 'wide' filters are described by  $r < 1$  ( $t < x$ ). The filter where  $x = t$  is the Haar filter with  $r = 1$  (see Figure 3.15).

As with the Haar shown in Figure 3.6 the top half of the filter is positive and the lower half is negative. When convolved with the video data, temporal edges with a nonparallel component to the time axis are detected as moving edges. The temporal edge to be detected is shown as a grey line in the figure. Its speed is described by the angle  $\theta$ . The areas  $B$  and  $b$  represent the sections of background covered by the filter and  $A$  and  $a$  are sections of

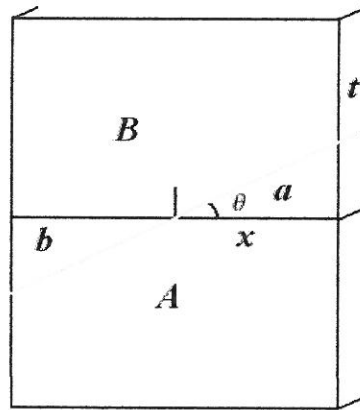


Figure 3.15: Diagram of a general cuboid  $s$ - $t$  filter simplified to 2D. The black horizontal line represents the filter's discontinuity plane; the grey sloped line is the moving edge in data;  $\theta$  represents speed of moving edge.  $x$  and  $t$  are filter dimensions.  $A$ ,  $a$ ,  $B$  and  $b$  are areas of filter. Equation 3.31 describes filter output.

the moving object. If the speed is high,  $\theta$  will be low and the areas  $a$  and  $b$  will be small. This means the filter output will be high. The output of the general cuboid filter is given by:

$$W = (BI_B + aI_A) - (bI_B + AI_A) \quad (3.31)$$

where  $I_A$  and  $I_B$  represent the pixel intensities of the  $A$  and  $B$  regions respectively. In the case where motion is at a constant speed, the motion edge will be straight and areas  $A = B$  and  $a = b$ . Eqn. 3.31 reduces to:

$$W = C(A - a) \quad (3.32)$$

where  $C$  is the edge contrast,  $C = I_B - I_A$ . The equation needed to describe

the area  $a$  depends on angle. If  $x \tan \theta < t$ ,  $a$  is a triangle, otherwise a more complex calculation is required:

$$a = b = \begin{cases} \frac{x^2}{2} \tan \theta & \text{if } \theta \leq \tan^{-1} r \\ xt - \frac{t^2}{2} \cot \theta & \text{if } \theta > \tan^{-1} r \end{cases} \quad (3.33)$$

$$A = B = 2xt - a \quad (3.34)$$

These equations show that the response of a general filter is a function of both edge speed and the width ( $x$ ) and height ( $t$ ) of the filter itself. Figure 3.16 demonstrates this relationship for a range of tall and wide filters. Filter output is recorded as the edge angle,  $\theta$ , is increased from  $0^\circ$  to  $90^\circ$ . Outputs are normalised to the maximum for each filter. The edge contrast is the same in each case.

The central grey line represents a cubic Haar filter ( $r = 1$ ). This has a near linear relationship with a maximum at the highest speed and dropping to zero at zero speed. Introducing asymmetry to the filter by making it tall or wide results in a concave or convex plot respectively. The dotted line divides the filter profiles into two regions according to equation 3.33; (i) when  $\theta \leq \tan^{-1} r$ , and (ii) when  $\theta > \tan^{-1} r$ . Greater asymmetry in the filter produces a sharper transition between these two regions. The extreme case of a 1D column of pixels is also shown ( $r=100$ ). Background modelling, which uses such 1D columns, is often described as a method of motion detection. This plot demonstrates why this is not the case, and in fact, *change* is being detected rather than motion. The plot remains at maximum until dropping

to zero suddenly at zero speed. The response is insensitive to speed and gives a binary output:

$$f(x = 1) = \begin{cases} 1 & \text{if } \theta < 90^\circ \\ 0 & \text{if } \theta = 90^\circ \end{cases} \quad (3.35)$$

Although the output of each asymmetrical filter is contrast dependant, the ratio for a pair of differently asymmetric filters with the same edge data is contrast independent. A filter pair can be defined as when the wide filter's  $r$  value,  $r_W = \frac{1}{r_T}$ , where  $r_T$  is the tall filter's  $r$  value. From equation 3.32:

$$\frac{f(r_W)}{f(r_T)} = \frac{C(A_W - a_W)}{C(A_T - a_T)} = \frac{x_W t_W - a_W}{x_T t_T - a_T} \quad (3.36)$$

as the filters form a pair,  $r_W = \frac{1}{r_T}$ . For convenience, we also make  $t_W = x_T$ , which also implies that  $x_W = t_T$ . Figure 3.17 shows the output profiles of the ratios of several filter pairs,  $r_W = (0.9, \dots, 0.5)$ . The pair ratio output,  $F(r)$ , has three regions: (i) where  $\theta \leq \tan^{-1} r$  for both filters; (ii) where  $\theta > \tan^{-1} r$  for both filters; (iii) the more complex case where the filters are in different output states. In all regions the ratio of filter pairs is independent of edge contrast:

$$F = \frac{f(r_W)}{f(\frac{1}{r_W})} \quad (3.37)$$

which is approximately proportional to  $(\frac{\pi}{2} - \theta)$ .

This calculation can be extended to 3D, as illustrated in Figure 3.18.

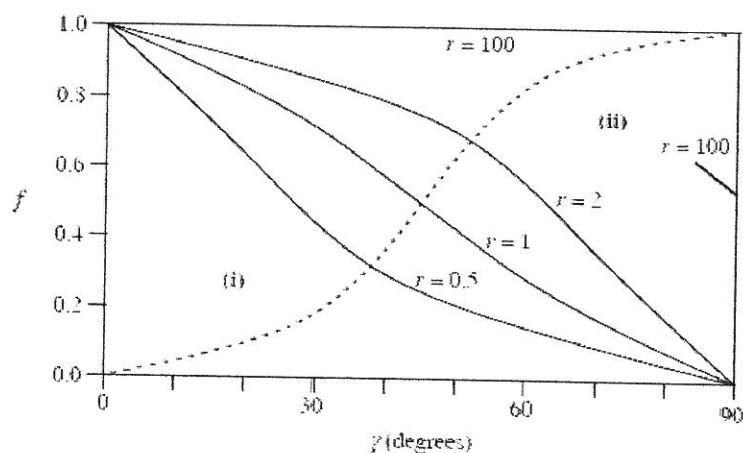


Figure 3.16: Graph shows the response to changing edge speed of four cuboid filters.  $r = 1$  is the cubic Haar and shows a near linear response.  $r = 100$  is a 1D column of pixels as used by background modeling.  $r = 0.5$  and  $r = 2$  are a *wide* and *tall* filter pair. Outputs are normalised.

The 3D case reduces to the 2D case when  $\beta$ , the directional component, is zero;  $\gamma$  also reduces to  $\theta$  in this case. The corners of the filter cause some sensitivity to changes in  $\beta$ , which manifests itself as an uncertainty in edge speed measurement. Using the filter model shown in Figure 3.18, the filter pair ratio output can be mapped. For an arbitrary pixel in the filter, the distance from a 3D moving edge is given by  $t$ :

$$d = x \cos \beta - y \sin \beta \quad (3.38)$$

$$t = d \tan \gamma \quad (3.39)$$

These equations can be used to calculate the continuous (non-pixelated) theoretical output profiles of arbitrary filters. Figure 3.19 shows the output maps for a cuboid filter pair ratio (left) and a spheroid filter pair ratio

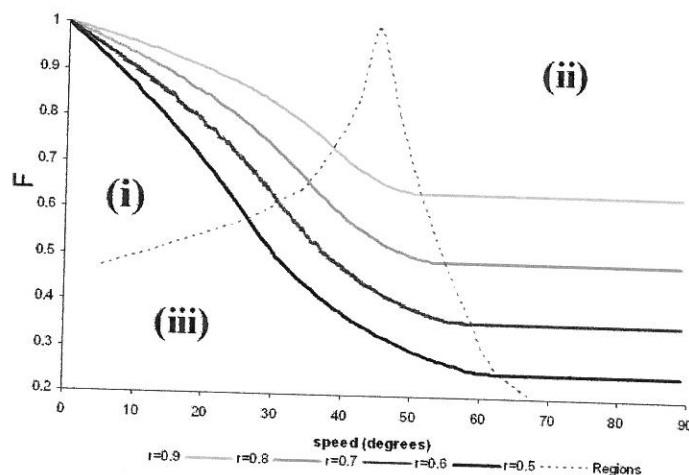


Figure 3.17: Contrast and direction insensitive speed detecting filters. Calculated from the ratio of two asymmetrical filters. Speed (temporal angles) range from  $0^\circ$  (maximum speed) to  $90^\circ$  (not moving).

(right) in  $\gamma$ - $\beta$  space. Filters were sampled over a wide range of angles from  $-90^\circ$  to  $+90^\circ$  to demonstrate output symmetry. Note the 'bumps' due to corner effects at  $\beta = \pm 45^\circ$  in the cuboid filter plot (a). This leads to some uncertainty in angular measurement (and thus speed measurement), as can be seen in (c). The spheroid output lacks this uncertainty (b, d); however perfectly spheroid filters cannot be produced for small sized filters due to pixelisation. Except for slowly moving edges where uncertainty is large, edge speeds can be determined with a precision of approximately  $\pm 5^\circ$ .

The profile in Figure 3.19 (c) and (d), where  $r = 0.5$ , can be described by the following equation:

$$F(r) = \frac{\tan^{-1}(r \cot \gamma)}{\tan^{-1}(\frac{1}{r} \cot \gamma)} \quad (3.40)$$



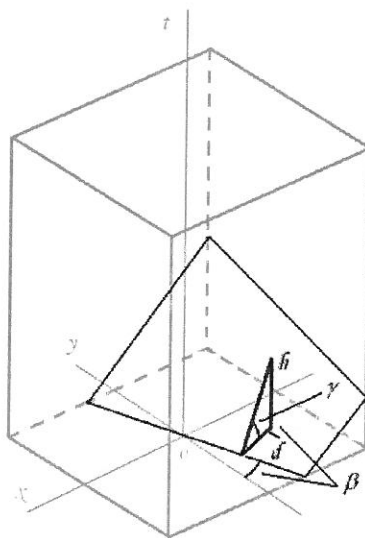


Figure 3.18: Diagram of a general cuboid  $s$ - $t$  filter in 3D. The figure shows only the top half of the filter; the surface bounded by the black line is the moving edge of the data;  $\gamma$  represents the edge speed and  $\beta$  the motion direction;  $d$  is the distance of an arbitrary pixel  $(x, y)$  from the line described by  $(O, \beta)$ ;  $h$  is the vertical distance of that point from the plane of the moving edge.

where, as  $\gamma \rightarrow 90^\circ$ ,  $F(r) \rightarrow r^2$ , which explains the horizontal limit of 0.25.

Figure 3.20 shows motion information derived from real video using this method. Intensity in the left-hand image indicates edge speed only. Motion direction is not detected by the filters used and edge contrast has been removed by the cancelling process described above.

This speed selectivity can also be achieved using an ordinary Haar filter by altering the decomposition process. Wide filtering is achieved by inserting an extra spatial scaling step before each  $s$ - $t$  Haar decomposition step. Tall filtering results from inserting a temporal-only scaling step into the decom-

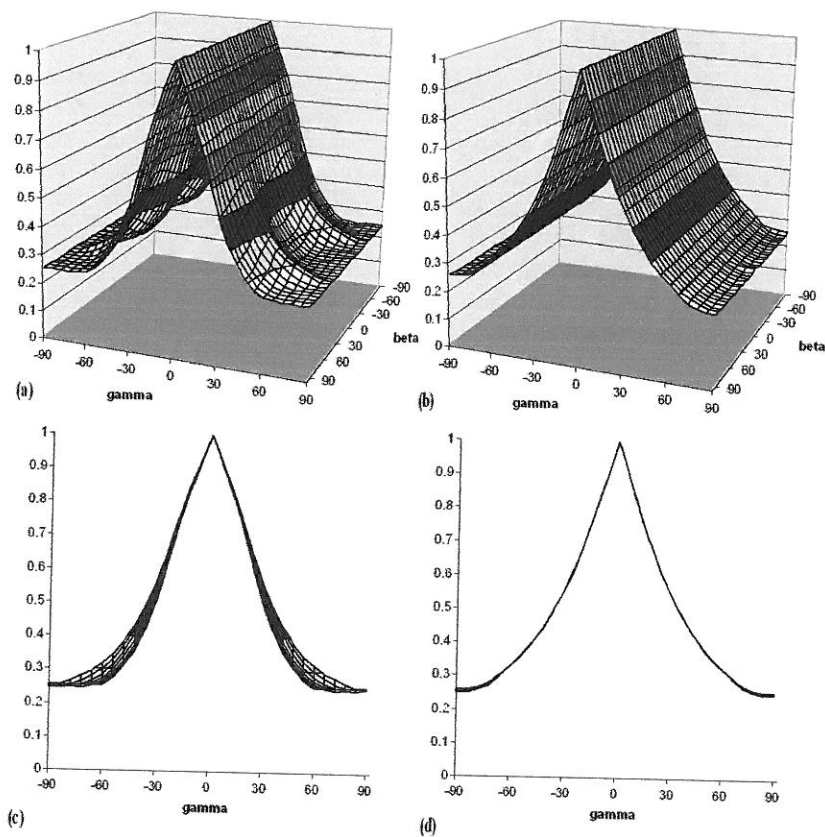


Figure 3.19: The ratio output maps of asymmetric cuboid filter pair (left) and spheroid filters (right) in  $\gamma$ - $\beta$  space. Top (a, b) are the 3D surface plot, bottom (c, d) are the respective edge-on profiles.

position process.

### 3.5 Summary

The most common approach to motion detection in video tracking applications is background modelling. This has been pursued by many earlier workers despite a lack of theoretical foundation which might define when, or if, this approach is valid.



Figure 3.20: Example of contrast free motion detection using non-symmetric filters, as described in Section 3.4.2. The video (left) shows two pedestrians walking towards each other. In the motion image on the right intensity indicates speed only. Edge contrast has been removed and the result is invariant to edge direction. See Figure 3.14 for comparison with DoOG derived motion. Blacked out areas within the pedestrian's motion profile are due to a local lack of edges.

This chapter has investigated background modelling and demonstrated its weakness. 1D statistical pixel processes can only detect change, rather than true motion, because motion is a spatio-temporal phenomenon. Pixel processes are unable to distinguish motion from noise, as shown in Figure 3.5. Further, Figure 3.16 shows that this method is insensitive to the speed of motion, producing only a binary, change/no-change output.

This chapter presents a powerful new paradigm for motion detection. This method, called Motion Distillation, uses spatio-temporal wavelet decomposition to detect motion as edges in space time. Figure 3.7 demonstrates motion detection by Motion Distillation and compares it to the background modelling techniques of temporal median filtering and Gaussian Mixture

Modelling. Table 3.1 provides quantitative detection results for a number of videos. The computational costs of the new method are extremely low at just 1.14 processes per pixel.

Finally, the chapter explores a number of variations on the Motion Distillation method using larger DoOG wavelets and general cuboid wavelets. Large wavelets allow a smooth Gaussian profile. Motion direction and speed information has been derived by combining the outputs of three filters of different orientations. Alternatively, two rectangular filters can be combined to achieve speed detection which is contrast and direction independent.

Motion Distillation provides a strong foundation for the later task of object tracking, as discussed in Chapter 4. The method also generates non-binary motion detection, which, in Chapter 5, will be used for direct analysis and classification of object behaviour.

## Main points

The main points and achievements of this chapter are:

- Background modelling is inefficient and does not output true motion detection.
- Motion is a spatio-temporal phenomenon and can be detected using spatio-temporal edge filters.
- The  $s$ - $t$  Haar wavelet decomposition is a very efficient and robust method of motion detection, but Haar output is edge contrast depen-

dent.

- Contrast can be removed by:
  - Using 3 orthogonally orientated DoOG wavelets to output contrast independent speed and direction information.
  - Using 2 different sized cuboid filters to output contrast independent speed information.
  - A third method for removing contrast for the purpose of behaviour analysis is presented in Chapter 5.

## Chapter 4

# Dual-Channel Tracking

Tracking is the task of maintaining object identity over time. Many tracking schemes reported in the literature include an inertia-based predictive mechanism, whereby past motion is used to predict a future location. Noise and uncertainty mean that the predicted location is rarely perfect. At the predicted location an appearance model is used to search for the true object location. The error, distance between predicted and true location, is fed back to improve the next prediction. Figure 4.1 illustrates this approach.

Location prediction serves to limit the search area, as a global scene search is prohibitively costly, and reduces the chance of mistakenly detecting clutter. Commonly, prediction uses the Kalman filter or the particle filter. Location prediction suffers from two fundamental flaws with regard to pedestrian tracking. First, it forces a total reliance on the appearance matching (AM) algorithm, even while the object is in motion; also, pedestrians actually change shape and appearance in order to move and an AM algorithm

sufficiently complex to predict this change may be also very costly. Second, prediction algorithms rely on limits and assumptions of expected object motion. Pedestrians frequently behave in a highly complex way, changing direction, starting, stopping, meeting other pedestrians, etc. Particle and Kalman filter based methods commonly lose track of their target when it makes an unusual and sudden movement or when the object rotates or deforms, even though this very motion information could be used to compensate for AM weakness.

This chapter discusses the operation and problems of the 'predict and match' paradigm of tracking. This is followed by a detailed discussion of the operation and basis of the tracking method used here. This thesis uses a dual channel 'form-motion' tracking architecture similar in approach to Jorge *et al.* 2004 [95, 96]. This scheme incorporates both the motion detection module presented in the previous chapter and an appearance model for use when motion detection is inappropriate. This chapter also discusses the choice of the appearance model and presents a quantitative comparison between two candidates. The chapter concludes with an extensive evaluation of the practical operation of the dual channel tracking method.

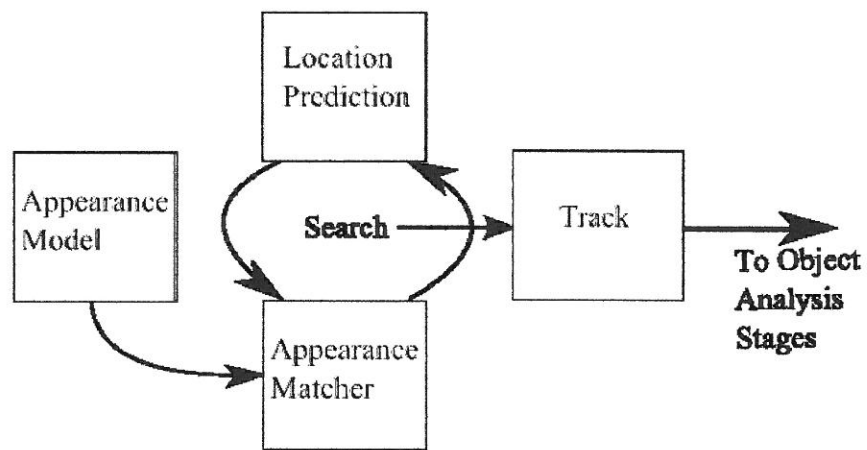


Figure 4.1: A general scheme for foreground, 'predict and match' tracking schemes. These methods use iteration between AM detection and future location prediction. The Appearance Model must be provided. The object track is passed on to the behaviour analysis modules.

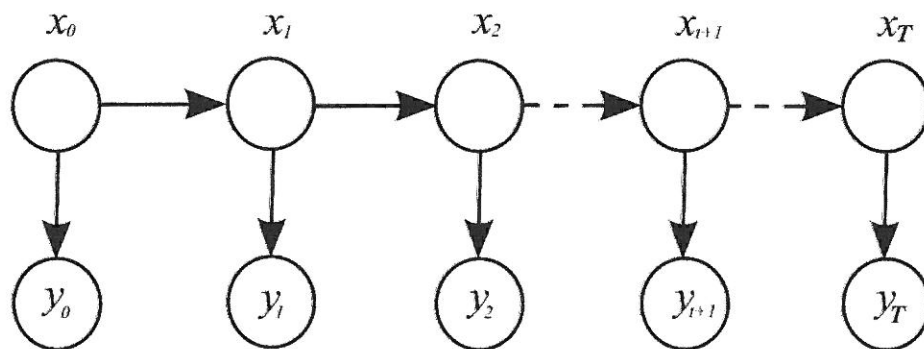


Figure 4.2: A scheme of the hidden state vectors,  $x$ , and their relation to future values,  $x_{t+1}$ , and current measurements,  $y$ .



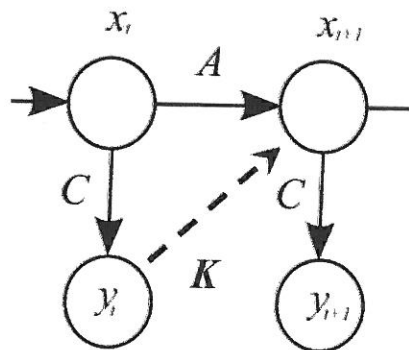


Figure 4.3: Illustration of the predictive process of the Kalman filter.  $x_{t+1}$  is generated from  $x_t$  using  $A$ , the *state transform matrix*, see equation 4.3. The state,  $x$ , is related to the measurement,  $y$ , through  $C$ , the *input transform matrix*, see equation 4.1. The Kalman gain matrix,  $K$ , allows the measurement error to be fed back to improve prediction.

## 4.1 Prediction

### 4.1.1 Kalman filter

In prediction terminology, the state vector, or true value, can be only indirectly detected through a noisy measurement. Figure 4.2 illustrates this relationship. Time runs from  $t = 0$  to the current time,  $t = T_c$ , through to the next, predicted, time step,  $t = T_c + 1$ , through to the final time,  $t = T_f$ . The current observable measurement,  $y_t$ , is related to the hidden 'true' state,  $x_t$ , using  $C$ , the *input transfer matrix*:

$$y_t = Cx_t + \nu_t \quad (4.1)$$

The measurement noise,  $\nu_t$ , is assumed to have a zero mean Gaussian *pdf*, with  $R$  as the *measurement noise covariance matrix*:

$$p(\nu) \sim N(0, R) \quad (4.2)$$

The Kalman filter has two stages; predict and update. The aim of the Kalman filter is to *predict* the future state,  $x_{t+1}$ , and thus the future location of the measurement (where to begin looking)  $y_{t+1}$ . Involved in this is the *state transition matrix*, which relates the current state to the future state using the physics underlying object motion, and the noise,  $\omega$ , which is a zero mean Gaussian *pdf* with *process noise covariance matrix*  $Q$ :

$$x_{t+1} = Ax_t + \omega_t \quad (4.3)$$

$$p(\omega) \sim N(0, Q) \quad (4.4)$$

The noise terms are used to determine the search area. After the object has been located, using an appearance model, the error (difference between the predicted location and the true location) is used to *update* the transfer matrices:

$$\hat{x}_{t+1} = x_t + K_{t+1}(y_{t+1} - Cx_{t+1}) \quad (4.5)$$

where  $K_{t+1}$  is the *Kalman gain matrix*:

$$\begin{aligned} K_{t+1} &= P_{t+1}C^T(CP_{t+1}C^T + R)^{-1} \\ P_{t+1} &= AP_tA^T + Q \end{aligned} \quad (4.6)$$

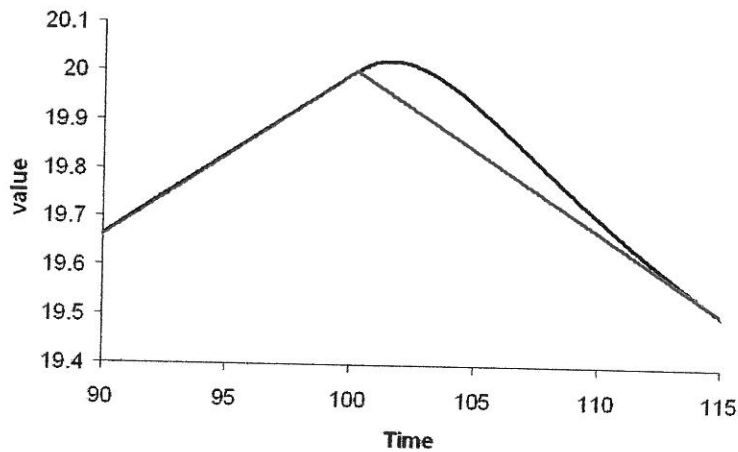


Figure 4.4: A demonstration of the inertial behaviour of the Kalman filter. The predicted position overshoots the true data position when the data suddenly changes direction.

Extra detail, along with derivations, can be found in [72].

The operation of the Kalman filter can be clearly demonstrated using a 1D tracking example. The Kalman uses an assumption of inertia, that the object will tend to continue to move in the same direction, speed and acceleration as it has in the past. Using this assumption the Kalman will filter out higher frequencies of the signal as noise. In Figure 4.4 the data increases linearly to the point (20, 100) and then suddenly reverses and starts to fall linearly. The Kalman tracks this data but loses track at the point where the inertial assumption fails. In this case, the true measurements are inputted directly, so the Kalman will use the prediction error to 'catch up' with the true signal eventually. In the visual tracking case, where the predicted location is used as the starting point for an appearance model search, such a failure of the inertial assumption may lead to a complete loss of the target. It is also

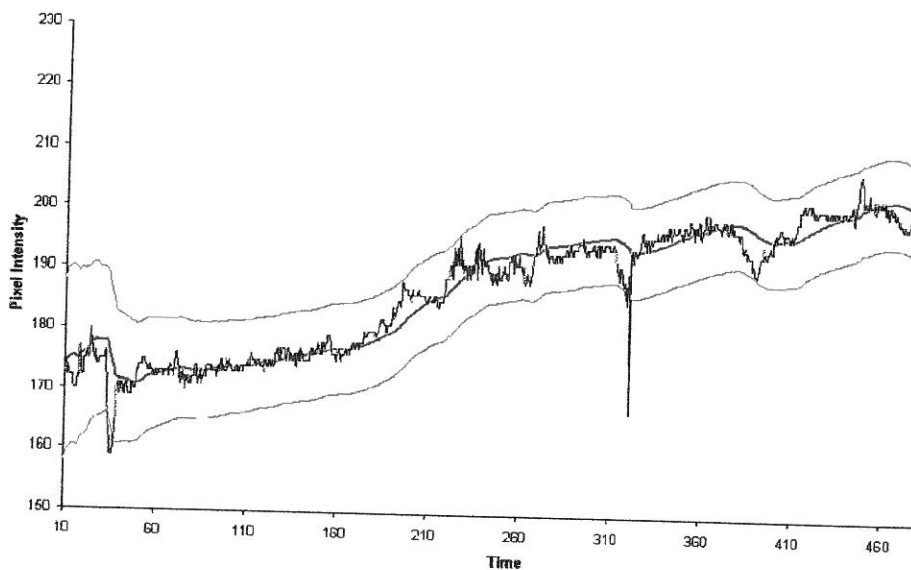


Figure 4.5: A pixel process modelled by a Kalman filter. The noisy input data from the pixel, along with central statistical trend line and error thresholds. This demonstrates the noise filtering properties of the Kalman.

worth noting that this is an assumption of apparent inertial motion, from the perspective of the camera frame, which is quite different from real-world inertial behaviour. Figure 4.5 shows a similar 1D task, this time using real data – a pixel value tracked by a Kalman filter, along with the automatically determined thresholds. Kalman predicted pixel processes are used in some background modelling methods [99].

#### 4.1.2 Particle filter

Particle filters represent an attempt to overcome the limitations of the deterministic Kalman filter, and the problem of an imperfect motion model, by introducing a random component to the search strategy. In cases where the

state space equations are non-linear particle filters can give better results than Kalman filters. The particle filter is an on-line version of the batch Markov chain Monte Carlo method.

Although there are many variations of particle filter, the core algorithm is Sequential Importance Sampling (SIS). The aim is to represent past measurement locations (samples) with weights and to compute new estimates based on these samples. As the number of samples approaches infinity the SIS filter approaches the optimal Bayesian estimate.

The procedure begins with a 'Random Measure' using weightings of previous states, that characterises the posterior *pdf*. The weights are normalised and chosen using 'Importance Sampling' of the results of testing those locations. This allows the motion model to control the search procedure. A posterior density is produced which is used to guide the search. The results of the search are then used to update the weights.

Diagrammatically, this can be seen in Figure 4.6, where the 'particles' at the top represent the initial weights. In visual tracking application, this motion model is used to guide the search procedure. The outputs of the appearance model tests are fed back into the system to re-weight the particles for the next iteration.

### 4.1.3 General problems

To predict the future location of an object, a prediction algorithm requires a model of the object's motion. This is commonly based on an assumption of

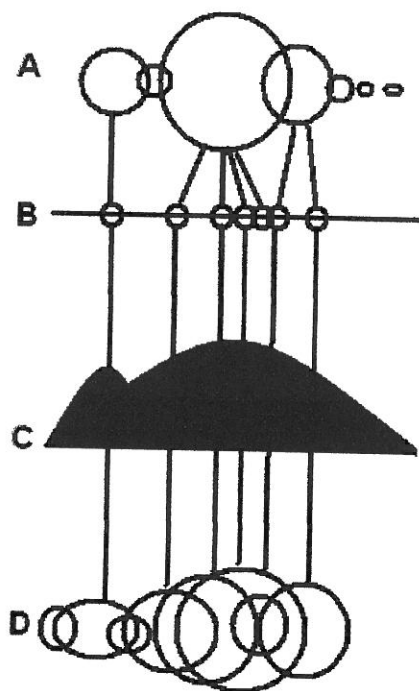


Figure 4.6: A schematic of one prediction cycle of the Particle Filter. A) shows the initial 'particles' with size representing weight. B) new search locations (unweighted particles) are chosen based on initial particle weights. C) Appearance model test returns a multimodal *pdf* of target location. D) This *pdf* is used to assign weights to the new particles generated in B.

inertial motion. That is, the object is likely to continue to move in the same direction and at the same speed or acceleration. For rigid objects such as cars this assumption generally holds. Cars move in straight lines along flat surfaces at approximately constant speeds (at least over the field of view of a CCTV camera).

Pedestrians are far less predictable. The scenes in which they move have staircases, ramps, hills, etc. which to a system with no scene model can make even continuous movement seem more complex and less predictable. Figure 4.7 illustrates this point with an example of a snowboarder being tracked in a snowy scene. Even though the snowboarder moves smoothly across the ground's surface, the unpredicted movement down a hill (a feature unknown to the system) causes the predictor to lose track. The track could have been maintained if the correct level for noise variance was set; however, this is difficult to determine in advance and a higher noise variance will increase the risk of detecting clutter.

Another key difficulty is the strong reliance on the appearance model test (as directed by the motion model). The predictor only tells the system where to look in the scene. Then the scene is tested using the appearance model for the target. Even if the location is correct, a poor result in the appearance test may cause the track to switch to a similarly poor match due to local clutter and the track may be ultimately lost. Pedestrians change shape and appearance as they move, due to swinging arms and legs and rotation as they change direction. As discussed in Section 4.2, this effect is highly dependent on the specifics of the appearance model used, whether it uses colour, shape,

feature points, etc. Figure 4.8 shows an example of a particle filter which has lost track of the person on the left. The appearance model here is based on detection of the eyes and mouth. The target walked into the scene from the left, and as he sat down, turned his head to look down, occluding the view of his face somewhat. The particle filter then lost track as it became stuck on the local minimum of the clutter on the wall.

The final example illustrates the difficulty of predicting pedestrian behaviour. Figure 4.9 shows two pedestrians meeting and interacting. The appearance model is based on a colour histogram and both pedestrians are wearing similarly coloured clothes. When they meet, they stop and shake hands. However, the Kalman filter does not predict this stop. Also, because the appearance model returns similar results for each person, the two tracks incorrectly swap onto the opposite targets.

## 4.2 Appearance model

As discussed in Section 2.2, there are a multitude of different appearance models described in the literature. Unfortunately, comparisons of the relative usefulness of different AM techniques for tracking scenarios are rare. The choice of appearance model is highly application and data dependent. Consideration must be made for the object to be tracked and how this object may be distinguished from others in the scene and from sections of the static background. It should further be considered when the AM will be used. In foreground methods the AM is used multiple times in each frame.





Figure 4.7: Tracking a target using an inertial motion assumption. The red ellipse (center) shows the last detected position of the snowboarder. The blue ellipse (bottom) shows the true location. (from Nummiaro *et al.* (2002) [136])



Figure 4.8: Example of face tracking using a particle filter. The person on the left turned his head as he sat down. The appearance model failed to detect his face properly and the tracker incorrectly fixed on a local minimum due to clutter. (Image courtesy of Dr. Hamadi Nait-Charif.)



Figure 4.9: Failure of both prediction and appearance model. The pedestrians suddenly stop moving. Their Kalman predicted motion continues and, because they are similarly dressed, the tracks swap onto the opposite targets.

In background methods the AM may be used only infrequently to resolve ambiguity, such as after an occlusion. This raises different requirements, as a non-rigid object may have changed significantly following an occlusion of several frames duration. The computational costs of the AM are, of course, another limitation.

For this application, two AM algorithms were tested for suitability. These were the colour histogram matching algorithm and template matching. Both methods were chosen due to their widespread use in the literature, their low computational complexity and the ease with which they can acquire and update models. It was anticipated that due to its insensitivity to rigidity, histogram matching would prove more robust than template matching for pedestrians. However, the results of this examination were somewhat surprising.

### 4.2.1 Template matching

Template matching aims to match the appearance model  $g(x, y)$  to the image,  $f(x, y)$  using a distance function. Each pixel is treated as a vector in colour space. Each pixel value in  $g$  is subtracted from the pixel at the same relative position in  $f$  using the Euclidean distance:

$$R(i, j) = \sum_{(x,y) \in Obj} \sqrt{\sum_c^{N_c} (f_c(i+x, j+y) - g_c(x, y))^2} \quad (4.7)$$

where  $c$  represents the colour axes; red, green and blue for colour images or just one intensity value for greyscale images ( $N_c$  is the number of colour axes). As template matching is highly dependent on correct alignment of the template over the image, the template must be raster scanned over the image, or a segment of the image (by varying  $i$  and  $j$ ) to find the best match. The minimum value of eqn 4.7 is the best match. A lower match threshold must be set to detect a match failure for the case where the object is not present in the image. More difficult problems include the template's sensitivity to rotation, deformation and scale. Objects detected by motion detection should be resized to a standard width and height before being matched using a template.

### 4.2.2 Histogram matching

Here the appearance model is constructed by generating a colour histogram of the segmented image. One implementation of a colour histogram is to

use three histograms, one for each colour channel. (An alternative is to use a single three dimensional histogram; however, this is problematic if small object will result in a sparsely filled histogram space.) Commonly, eight value 'bins' are used, rather than a full 256, as segmented object images usually contain insufficient pixels to fill a full histogram. Differences of image and model scale are dealt with by normalising each histogram; the sum of the values of all eight bins should equal 1. It is also possible to normalise the input color values to achieve colour constancy, although this is not attempted here.

To compare two colour histograms, the absolute difference of each *bin*, *b*, of the model, *g*, and the image, *f*, is summed for each colour, *c*. The three resulting histogram values form a vector in colour space. The length of this vector is used as the comparison between two colour histograms, a shorter vector being a better match:

$$R = \sqrt{\sum_c^{N_c} \left( \sum_b^{N_b} |g_c(b) - f_c(b)| \right)^2} \quad (4.8)$$

where  $N_c$  and  $N_b$  are the number of colours and bins respectively. Histogram matching is computationally cheaper (however, the time required to compute the histogram negates this advantage somewhat. This may only be significant for large objects). It is also less sensitive to alignment, avoiding the multiple tests needed for template matching. It also doesn't require the image to be resized. Instead, the histogram is normalised.



### 4.2.3 AM comparison test

The chosen AM for this application will be required to be robust to imperfect segmentation and give good results for rigid and non-rigid objects. The motion distillation system described in Section 3.3 was used to extract a series of moving objects from a video. Segmented images of a car and a pedestrian were tested against the first image segment of the sequence. Figure 4.10 shows the results of template and histogram tests for a car and Figure 4.11 shows results for a pedestrian. Example results for numerous other objects are included to indicate the 'discrimination quality' of the test method. It can be seen that the matching ability of each method is very similar, and that in most cases, either method is sufficient to identify the correct object from its neighbours.

Of note are the matching results which are smooth over time for the car but more erratic for the pedestrian. This is due to the rigid motion of the car and the deformations of the pedestrian as he moves. It might be expected that, while the template method would be sensitive to non-rigidity, the histogram should be invariant to the shape of the pedestrian, but this was not the case. On close examination of the frames, it was discovered that this was due to lighting and shadow effects which altered the colour of the pedestrian as he moved. This is expected to be a common effect in outdoor surveillance applications.

The AM's robustness to poor segmentation is also of interest. To test this a series of segmentations of an object was prepared, ranging from extreme

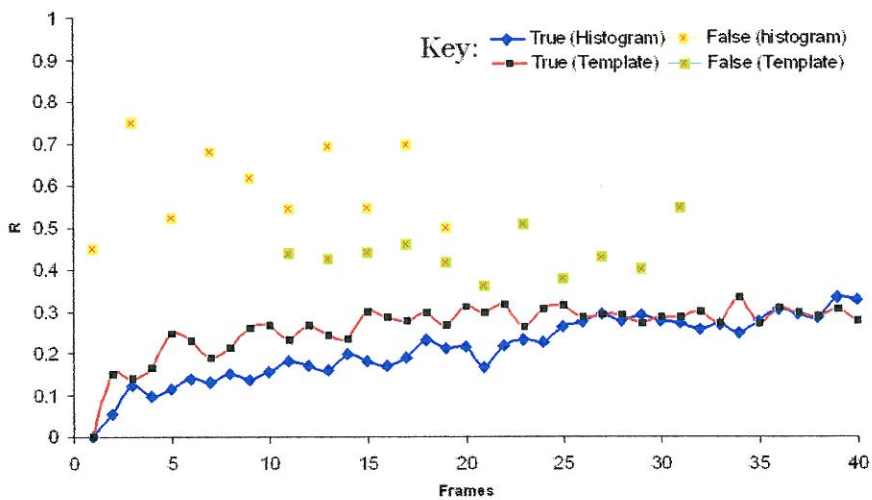


Figure 4.10: Comparative matching results for a car using a colour histogram model and a template model. The model is acquired in the first frame and not updated after this. The 'false' points are matching results for a set of non-target objects.

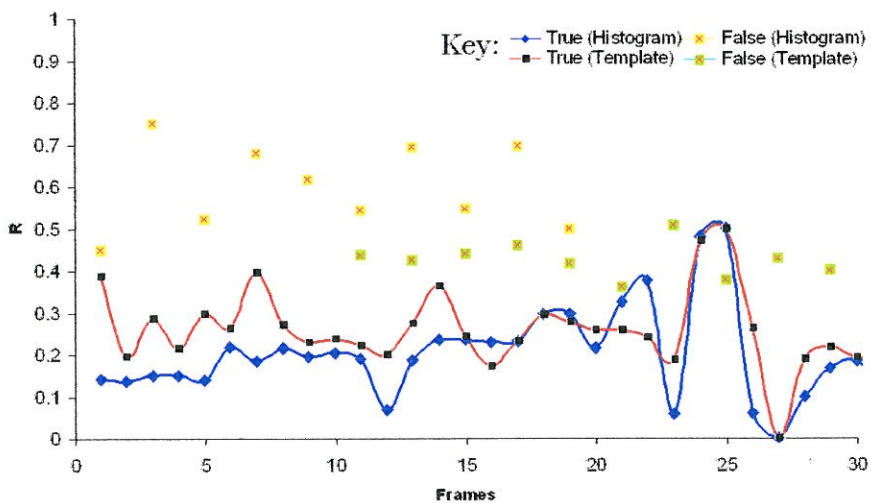


Figure 4.11: Comparative matching results for a pedestrian using a colour histogram model and a template model. The model is acquired in the first frame and not updated after this. The 'false' points are matching results for a set of non-target objects.

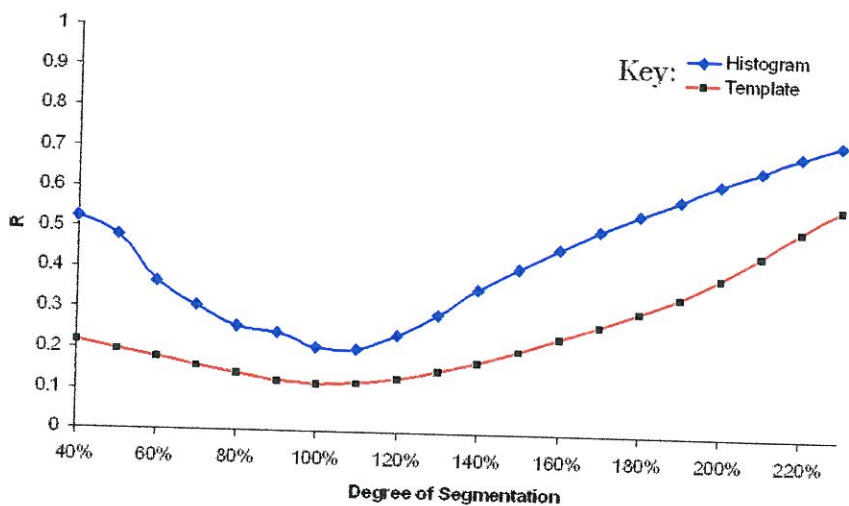


Figure 4.12: Test of sensitivity to segmentation. Using a properly segmented model in both cases, matching results for a histogram and template were produced for undersegmented and oversegmented target objects (oversegmentation is represented by less than 100% on the graph and undersegmentation is greater than 100%).

undersegmentation to oversegmentation<sup>1</sup>. Figure 4.12 shows results for the template and histogram matcher. This is computed for one example object from a single by manually varying the size of a cropping rectangle from over two times to under half the correct size. Template matching is shown to be better for poor segmentation. However, as the templates must be aligned through multiple sample matches, this method engenders greater computational costs.

These comparisons illustrate that the two methods give very similar matching results and both can distinguish the target object from neighbouring objects. Surprisingly, due to lighting effects, the histogram method is

<sup>1</sup>An AM's response to oversegmentation is also an indication of its response to partial occlusion of the object during tracking.

not much better at matching deforming objects. It was decided to use the histogram method due to its significantly lower computational cost and its slightly better distinguishing ability.

### 4.3 Dual-channel approach

The tracking schemes discussed above do not use a motion detection component. When a motion detection component is incorporated, this changes the nature of the tracking problem considerably. Now it can be assumed that any moving objects will be detected and the tracking task will be reduced to maintaining object identity through static and dynamic occlusions.

In this section we present an alternative solution to the post-detection tracking problem. We use the terms 'sprite'<sup>2</sup> and 'blob' to mean similar but distinct things. A blob is a connected region corresponding to the area of a motion in the current frame. A blob may represent a whole moving object, part of an object if only part of a object is moving, or several overlapping objects, in the case of groups of objects moving together. A blob has no knowledge of real objects, either spatially or in time. However, a sprite represents an attempt to integrate a notional 'object' from many blobs over many frames. This attempt to integrate blobs into sprites is the problem of post-detection tracking. Object refers to the real world target being tracked.

This problem can be divided into a series of questions. When a blob is detected, does that blob belong to a currently active sprite, an inactive but

---

<sup>2</sup>This is also referred to as a Video Object by the MPEG community.



visible sprite, an invisible or occluded sprite, or a new sprite? Conversely, if a sprite cannot be matched to any current blob, is this because the object has stopped moving, been occluded, or left the scene?

A further complication is due to the limits of motion detection. If an object stops moving then no blob will be detected. However, we may assume that a stationary object will not change its appearance radically from frame to frame, as such a sudden transformation would be detected by the motion-detection stage. Thus, in a missing blob situation, we may distinguish between the occluded object case and the stationary object case using an appearance matcher at the last known object location.

An alternative approach would be to use a layered background model, as in [179]. Layered background models store the past locations of stationary objects as a 'layer' of the background model, allowing their location to be maintained in memory.

This problem statement suggests the dual-channel tracking architecture described in the neurological literature, such as Giese and Poggio [75] (see also Section 2.6). As with the HVS, one channel contains motion information only and the other contains instantaneous form or appearance information. Neither on its own is sufficient.

The solution presented here relies primarily on the motion channel, as computed using  $s$ - $t$  Haar wavelet decomposition, for initialisation and detection. Detected blobs are sorted and matched using a two-stage rule-based matching scheme. Blobs detected in the motion channel are sorted first us-

Table 4.1: Qualitative sorting results of blob identity maintenance. Results show occurrence of particular event classes in the sample videos. The top row indicates matches which can be correctly resolved by an overlap test; other rows require an AM test. The final row indicates static occlusions and scene exits. These results are compiled from a sample of 1,935 events in three videos using software and manual ground truthing.

Event types	Video 2	CAVIAR	Video 3
unique match $\geq T_O$	96%	80%	78%
unique match $< T_O$	1%	3%	2%
Multiple matches	1%	6%	12%
AM split	0%	1%	0.5%
Match from memory	1%	1%	1%
No match	1%	9%	6.5%

ing an area Overlap Test, followed by an appearance match test to resolve ambiguity. Single blob to multiple sprite matches are treated as dynamic occlusions. Unmatched blobs are considered as new objects. Next, the sprite list is searched for unmatched sprites. These can be due to either occlusion or stopped or 'sleeping' sprites. The form channel is then accessed to resolve the ambiguity. Table 4.2 presents these rules as they are applied in the software and Figure 4.13 illustrates the program flow within the dual-channel approach.

Is this approach valid? Viewed from a Bayesian framework, equation 4.9 represents the matching task:

$$P(T|x) = \frac{P(x|T)P(T)}{P(x)} \quad (4.9)$$

where  $x$  is the result of some matching test.  $P(x)$  is the probability of this result in the video being analysed. We wish to know the probability that

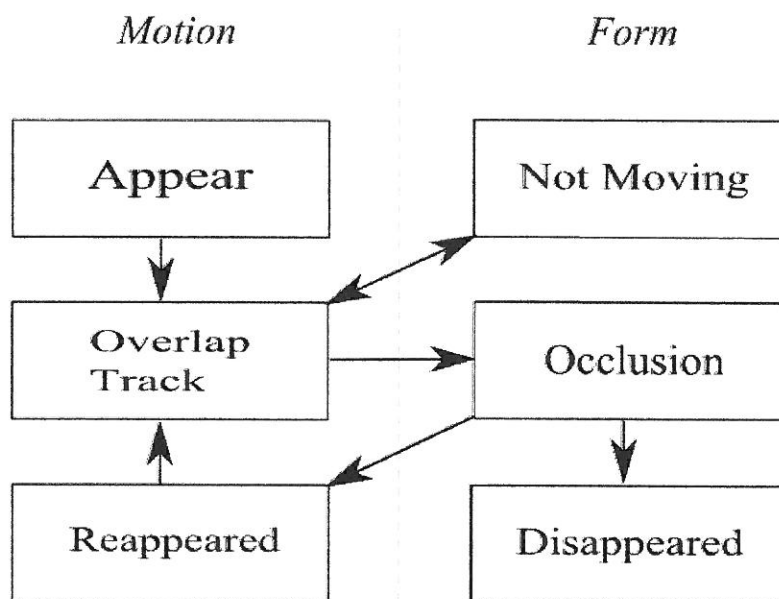


Figure 4.13: Illustration of algorithm flow within the dual-channel approach. New objects are detected and tracked primarily in the motion channel. Static occlusions and stopped objects are resolved by reference to the AM in the form channel. Dynamic occlusions are resolved in the motion channel. See also Table 4.2.

a measured positive match test result truly indicates that the tested blobs represent the same object. In equation 4.9,  $P(T)$  is the probability of a true match.  $P(x|T)$  is the probability of a positive match result in the case of a true match and  $P(T|x)$  is the probability of a true match given a positive match result.

These probability values must be calculated from the tracking data. This is a difficult task and the probability values are highly dependent on the content and events in the video and they change greatly from video to video. However, the question at hand can be answered with a study of a small

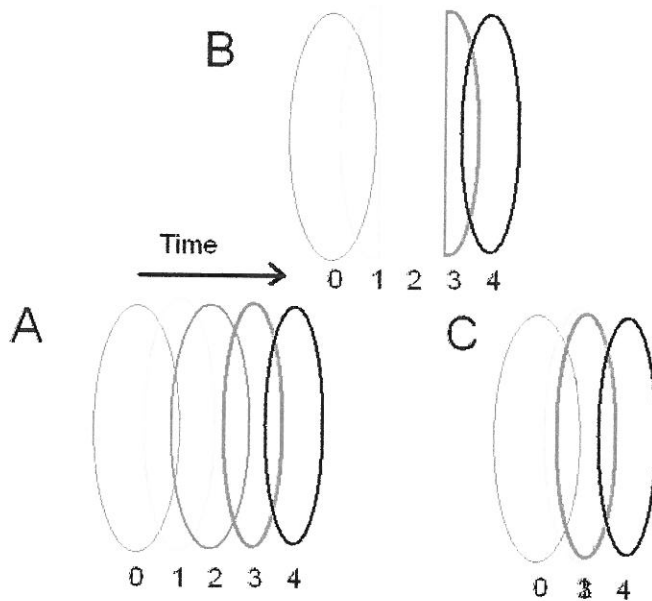


Figure 4.14: A schematic view of the motion channel output over time. Example A shows an object moving without stopping or being occluded. Examples B and C require reference to the form channel to decide whether the object has been occluded or stopped moving.

number of videos. Is the two-stage matching approach valid and is it correct to test blob overlap first (motion channel) and use the appearance model to resolve ambiguity (form channel). Table 4.1 provides quantitative results for the application of these rules for three videos and 1,935 matching events. The first row shows the percentage of matching events which can be correctly resolved with the overlap test alone. These results show that the quick overlap test is sufficient for a large majority of cases, with only a few percent of cases remaining ambiguous and requiring reference to the form channel.

Figure 4.14 shows a schematic diagram of this tracking system in action.

Each ellipse represents a detected blob in the motion channel. All objects are moving left to right according to the frame numbers underneath. In scenario A, the object moves without stopping or occlusion. In this case, the system matches incoming blobs to the sprite stored in memory using overlap alone when there is a unique match above an area threshold. Table 4.1 demonstrates that this initial, quick test is sufficient for between 78% and 96% of cases (depending on video content). In Figure 4.14, B shows the case where the tracked object passes behind a stationary object in frame 2 and C is the case where the object stops moving in frame 2. The system has no scene knowledge and so cannot predict when objects may be occluded or stopped; however, it can deal with object reappearance after several frames. In frame 2 the sprite will remain unmatched to a blob in both cases and the system must distinguish whether the sprite is occluded or sleeping. The sprite is sorted using the rules in the right-hand column of Table 4.2. The system uses the appearance model of each sprite, acquired during the tracking phase, and tests the current frame at the last known position of the sprite. In terms of dual-channel tracking, this step represents referencing the form channel.

### 4.3.1 Bayesian Networks

This section serves as a comparison of the tracking system described in this chapter to that of Jorge *et al.* 2004 [95, 96]. Jorge *et al.* proposed a two-step system where non-occluded objects are tracked using a simple algorithm and more complex interactions are resolved using a “data conflict” module. The

Table 4.2: Sorting rules of blob identity maintenance. First, blobs extracted from the motion channel are sorted and matched to sprites. Then the list of sprites in memory is updated and, in cases of ambiguity, checked against the form channel.  $T_O$  is the overlap threshold and  $T_{AM}$  is the appearance model matching threshold.

Motion Channel : Test Blobs	Form Channel : Update Sprites
<pre> if a blob was detected by MD stage   if OT gives a unique match <math>\geq T_O</math>     <b>match blob to sprite</b>   if OT gives a unique match <math>&lt; T_O</math>     perform AM test     if <math>\exists</math> a match <math>\geq T_{AM}</math>       <b>match blob to sprite</b>     else <b>register no match</b>   if OT gives multiple matches     perform AM test     if <math>\exists</math> one response <math>\geq T_{AM}</math>       <b>match blob to sprite</b>     else <b>register merged sprite</b>   else <math>\exists</math> no match     perform AM test     if <math>\exists</math> a response       <b>register occluded sprite</b>     else <b>register new sprite</b> </pre>	<pre> if the sprite was matched   if <math>\exists</math> a unique match     <b>update with blob data</b>   if <math>\exists</math> one sprite to multiple blobs     <b>split sprite</b>   if <math>\exists</math> multiple sprites to one blob     <b>merge sprites</b>   else <math>\exists</math> no blob match     AM test current frame     if object present       <b>register sleep sprite</b>     else <b>register occluded</b> </pre>

data conflict module deals with events such as occlusions, group merging and splitting.

As with the method proposed in this chapter, the first stage of the tracker attempts to match object motion silhouette, as output from the detection stage, from one frame to the next. Shape analysis is used as the matching method. In this chapter a simpler shape analysis method is used, in the form of an area overlap test.

The data conflict module combines an appearance test with global inference using assumptions on the movements of the tracked objects. The global inference feature requires that the method be carried out off-line or in batch mode. A modified on-line method is also proposed where the global inference step is greatly simplified; however, a delay of some seconds is still required to perform the step.

The method proposed in this chapter is similar to that of Jorge *et al.* without their global inference step, allowing it to track on-line and without a delay. The lack of a global inference step did not cause any noticeable tracking failures for the videos tested.

## 4.4 Results

This section provides qualitative results for tracking using the dual channel approach. The system was tested on a wide range of videos, both standard datasets and original video data. Appendix A provides details of the videos





Frame 80

Frame 161

Figure 4.15: Video 1: standard tracking scenario; single pedestrian outdoors in diffuse lighting.

used.

#### 4.4.1 Video 1

Video 1 is the simplest of the tracking cases that we will investigate. A single pedestrian moving at an approximately constant (real-world) speed (although this translates to an apparent deceleration and shrinking because of perspective effects). Lighting is constant and diffuse and there are no motion clutter objects, such as moving branches, etc. The system tracks described tracks easily in this video, see Figure 4.15.

#### 4.4.2 Video 2

Video 2 (Figure 4.16) is more complex. Lighting is strong direct sunlight. This leads to complex shadow and colour change problems which affect the AM stage (see Section 4.2.3). The pedestrian repeatedly enters and exits the scene, sometimes walking (frame 337), suddenly breaking into a run (frame



1560), changing direction suddenly (frame 1900), walking slowly towards the camera (frame 2196), etc.

For some applications it may be desirable to attempt to match new sprites entering the scene against those that have exited in the past (see frame 2368). However, for busy street scenes or long running systems this would require a large database and may produce many false positives. Here, this feature is generally turned off, meaning objects exiting from the boundary of the scene are deleted from the active memory.

A weakness of this system is shown in frame 1957. The pedestrian was tracked up to the point when he entered a deep shadow (frame 1900). Once under shadow, parts of the pedestrian's clothes have insufficient contrast and speed to be detected (see equation 3.15 in Section 3.3.3) leaving only the brightly coloured teeshirt. The new blob fails the overlap threshold test with the sprite of the previous frame because it is so reduced in size. The system attempts an AM test but this also fails to correctly match, because only the bright pixels of the teeshirt are recorded in the histogram, due to incorrect detection and segmentation. The pedestrian is still tracked by the system but it is incorrectly considered to be a new object.

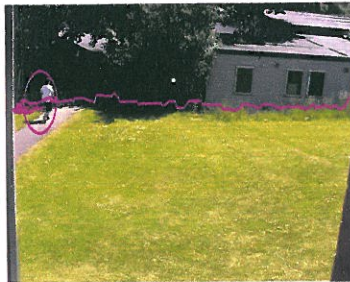
A human would know the pedestrian's legs could not disappear and would either assume it or look more closely for them (effectively lowering a threshold). To achieve this strength the system might require a structural model of the target or a knowledge of what can and can't happen to different classes of target.



Frame 337



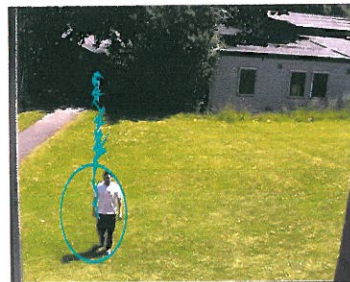
Frame 1560



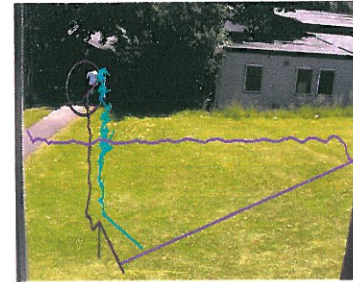
Frame 1900



Frame 1957



Frame 2196



Frame 2368

Figure 4.16: Video 2: a long video illustrating tracking results under a variety of pedestrian behaviours, including walking, stopping, running, and entering and exiting the scene.

### 4.4.3 Video 3

This video (Figure 4.17) illustrates how the system tracks in cases of dynamic occlusion and in the presence of severe motion clutter and noise. This video was captured during a minor storm, with the trees in the top left of the video shaking wildly. As the motion detection and AM system has no prior knowledge of what pedestrians are, it tracks these 'objects' just as it tracks pedestrians. Although the generality of the system is one of its major strengths, it causes a problem in this case which is only solvable using extra information on the objects of interest for a particular application. The behaviour analysis approach described in Chapter 5 can be used to categorise (and ignore if desired) certain classes of object.

The video contains two pedestrians who pass each other and interact three times. First, the two enter from opposite sides of the scene, meet, stop, shake hands, and then continue as before. During this, one pedestrian passes behind the other and is dynamically occluded. As shown in Figure 4.9, sudden stopping may cause problems for a predictive tracker. Second, the two meet, stop and then reverse direction. Finally they pass each other running without stopping.

The system successfully tracks these pedestrians as follows. When the motion detected blobs of two objects dynamically occlude (frame 395) it appears to the MD channel as a single merged blob. This is treated as a 'merged sprite' and the tracks and information of the separate sprites are combined in memory. While they move together, they are tracked as a single

object. When an object breaks away from the merged sprite the system will attempt to match it to one of the original components using the AM.

This merging behaviour can also be seen in an unintended case (frame 373 & 389). When the pedestrian on the left begins to enter the scene, initially the hand and foot are separately detected and tracked. Then, as the pedestrian moves fully into scene these separate blobs become one and the system merges them. Frames 389 and 395 show these merged tracks clearly. Merged sprites can also be correctly merged and demerged with other sprites (see 395 and 424). Although this behaviour does not effect the tracking goal, it could be easily hidden in a final system design by allowing special merges near the scene boundary.

During this video, the waving branches create several hundred spurious motion clutter objects which are all tracked. Despite this extra load and complication, the system still has no difficulty in tracking the pedestrians through occlusion and correctly maintaining their identity. Ideally, the system should be designed to ignore the moving branches. This would require a model of which object types to ignore or use of the motion information described in Chapter 5 to distinguish 'interesting' from 'uninteresting' moving objects.

#### **4.4.4 Other videos**

The dual-channel tracking algorithm was tested on the CAVIAR database and simulated CCTV security video supplied by the Home Office (Figure 4.18).





Figure 4.17: Video 3: Video illustrating tracking of dynamically occluding objects. The video also contains a high degree of motion clutter caused by the swaying tree branches and wind. The system successfully tracks the targets despite this.

The Home Office video shows a straightforward 'exclusion zone breach' scenario. The pedestrian is detected on entering the scene at the right and tracked as he walks to the fence on the left. There the pedestrian stops and so disappears from the motion channel. The system switches to the form channel and confirms that the target is still present. When he starts walking back, he reappears in the motion channel and is tracked until he exits the scene again.

CAVIAR contains more complex interactions and dynamic occlusions similar to Video 3 above. Here several pedestrians enter and exit the scene. Frame 182 shows the a fight scene where the participants are tracked as a single merged sprite. When they separate, as with Video 3, the two resulting objects will be matched with the original components of the merged sprite.

## 4.5 Summary

In  $2D+1$  approaches, such as particle filtering, the tracking task is approached from the methodology of *predict and match* without any motion detection stage. As shown in Section 4.1.3, this approach fails frequently when tracking pedestrians under real conditions. Pedestrians are simply not always predictable. Failure often occurs when the target changes velocity, despite the fact that this motion information could be used to overcome prediction problems.

However, motion information cannot detect a non-moving object. Nor can it match an emerging object to the track it had prior to an occlusion.



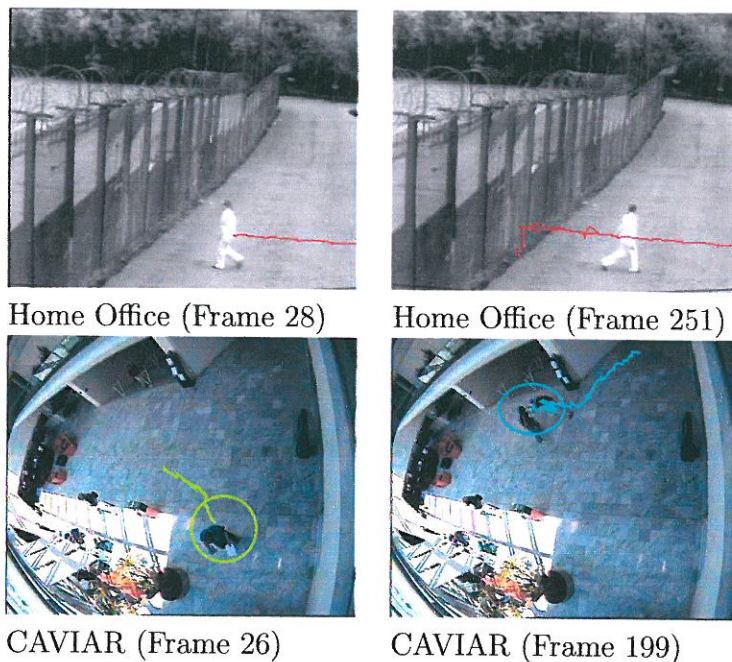


Figure 4.18: Tracking examples from the CAVIAR and Home Office datasets.

This can only be achieved using an appearance model.

This chapter has an approach to tracking, similar to Jorge *et al.* 2004 [95, 96], which combines the strengths of appearance matching with the powerful motion detection scheme described in Chapter 3. This tracking scheme maintains two separate information channels, motion and form, and switches from one to the other to maintain lock on the target through movement, stoppages and occlusions. Table 4.2 and Figure 4.13 describe the rules used to track objects while Table 4.1 defines quantitative results for tracking in a number of videos. The results section presents extensive qualitative results for a range of tracking scenarios.

Tracking need not be the final output of a surveillance system. We also

wish to classify moving objects in the scene and understand something about an object's behaviour. This task is explored in the next chapter.

## **Main points**

The main points and achievements of this chapter are:

- Prediction frequently fails when tracking pedestrians.
- Prediction, without motion detection, forces a reliance on the appearance model.
- Clutter is generated by false matches of the appearance model.
- Motion detection alone cannot track objects through occlusions.
- Dual-Channel form-motion tracking seems to provide an effective solution to these problems.



## Chapter 5

# Behaviour Analysis

In many ways behaviour analysis is both the ultimate goal and the most difficult part of an automatic surveillance system. The task is complicated by a lack of consensus on the requirements of CCTV systems. Several methods proposed in the literature aim only to catalog and chart pedestrian tracks in the scene. A few try to analyse the 'motion history' of the object.

The HVS incorporates a strong spatio-temporal motion detection stage and there is evidence that it tackles the behaviour analysis stage using the rich data field extracted directly from that initial detection stage. 'Biological motion', the characteristic organic movements of people and animals, has been shown [75] to be detected at this initial stage, rather than by some post-tracking processing.

There are few studies of future CCTV requirements. Troscianko et al. (2004) [171] studied the reactions of human CCTV operators and showed that

their ability to detect crime or suspicious behaviour had little connection with the path taken by the target, but focused mainly on intra-body movements and pose.

Categorisation of tracked objects may permit annotation of video data and later content-based search. What is required is a signal characteristic of particular behaviour, yet independent of a particular video, recording views, particular people, etc.

Here the output of the motion distillation stage is used directly, as is the case for the human visual system. A number of invariant *motion signals* are extracted and these are used to categorise moving objects into vehicles and pedestrians. Pedestrians are further categorised into a number of behaviours – walking, running, jumping, waving hands, etc. This information can ultimately be combined with the tracking and interaction data developed in Chapter 4.

This method is deterministic, whereas the human visual system is based on learning. The output of the motion distillation stage is a complex information field and it is presumed that an artificial neural network approach, trained on a large database of sample behaviours, might provide a more versatile categorisation ability. However, constructing such a training database is outside the scope of this thesis.

## 5.1 Biological models

The human visual system can recognise an extraordinary range of human behaviours from motion. Figure 5.1 shows the classic 'point light' experiment where lights are attached to the joints of actors. Different actions are filmed so that only the light points are visible. When a human subject is presented with still images of these lights, not even the human form can be detected. However, when the video is played, even complex actions such as dancing can be readily identified.

Giese and Poggio published a neurological model in 2003 which attempted to account for this. The model proposed that the dual-channel approach used for object detection and tracking be extended to behaviour detection. Learning using a neural network is important in the recognition of complex movements. Giese and Poggio, point out that learning is fundamental in the recognition of 3D stationary objects and the neural representation of objects seems to be based on learned 2D views. This supports the hypothesis that learning is involved in the recognition of complex movements [75].

The representation of motion in the model is based on a sets of learned patterns. These patterns are encoded as sequences of snapshots of body shapes by neurons in the form pathway, and by sequences of complex optical flow patterns in the motion pathway.

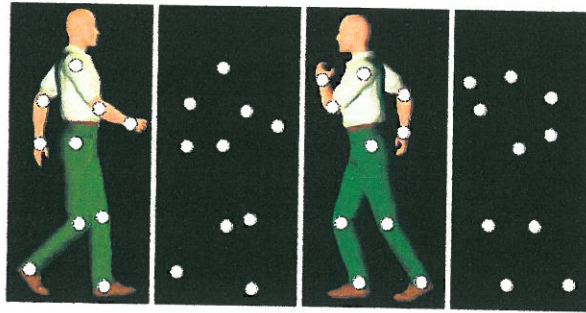


Figure 5.1: Point light representations are instantly recognisable, but only when in motion (From Giese and Poggio 2003 [75]).

## 5.2 Recognition approaches

### 5.2.1 Makris and Ellis

Whether tracking is achieved through foreground methods such as particle filtering or background modelling methods, when the system reaches the final behaviour analysis stage, only information on object position over time (i.e. the track) has been generated. Many approaches to behaviour analysis seek to highlight unusual object tracks. Makris and Ellis's influential work [115] uses a training video of the scene to build a map of common pedestrian paths, along with a measure of allowed variation at each point. During training, short tracks, or tracks of objects which change direction frequently, are eliminated. To counter the effects of perspective, paths are resampled at regular intervals in image space and velocity information is discarded. Resampling produces nodes at regular intervals. If a pedestrian moves sufficiently close to a path it is updated with this information, otherwise a new path is recorded. Distances are measured in image space as the separation between the nodes

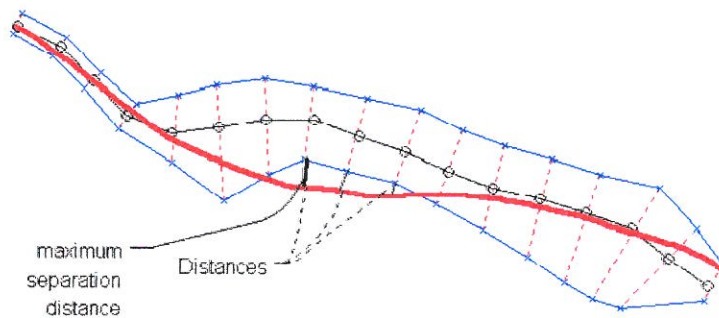


Figure 5.2: Tracks recorded in long videos are compiled into paths, composed of entry and exit points and equally spaced intermediate nodes. Unusual tracks are detected if the track is outside the bounding error bars. (From Makris and Ellis [115]).

of stored paths. Figure 5.2 shows the path nodes, variation envelope and the distance measurement to a new track. Path nodes are updated using the equation

$$\vec{x}^j = \frac{w}{w+1} * \vec{x} + \frac{1}{w+1} * \vec{x}_t \quad (5.1)$$

where  $w$  is the weighting of each node. This number is incremented with each update. If the new track is partly outside the variation envelope, that envelope is expanded to include it.  $\vec{x}$  and  $\vec{x}_t$  are the positions of the nodes of the path and new track respectively. Paths are resampled after update. When operational, the system matches new tracks to learned paths, highlighting those that are unmatched.

This approach has a number of disadvantages for practical CCTV applications. The learned routes are both scene and view dependant, meaning a long training phase would be required for each camera and location. If the layout changes during operation (i.e. a new obstacle is placed in a path,



forcing pedestrians to avoid it) the learned routes are invalidated.

The question also arises, is the approach valid? Can unusual behaviour be defined in terms of the path of the object centroid? Studies of CCTV operators reveal that characteristics such as pose and violence of action are used to a far greater extent than motion paths [171].

### 5.2.2 Dee and Hogg

Dee and Hogg developed [51] an interesting extension to track-based recognition. They reasoned that behaviour can be modelled as a series of goals and sub-goals through which each pedestrian must pass when walking through the scene. Goals are defined as exit points, whether at the scene borders or doorways within the scene. Sub-goals are at the corners of scene obstacles. Sub-goals allow a pedestrian to reach a goal indirectly when the direct route is not available due to obstacles. These goals and sub-goals must be manually added to the scene model as no automated method has been developed.

The aim is to determine whether a particular pedestrian track can be explained in terms of known goals or sub-goals, or whether it is *inexplicable*. At each point in the track the number of goals and sub-goals available to the pedestrian is calculated using line of sight determination. A cost function is calculated for each track, with low costs associated with direct movement. The pedestrian should walk in straight lines through these points (low cost). Inexplicable behaviour may be due to loitering or unknown goals (high cost). The system was evaluated by comparing automatically detected

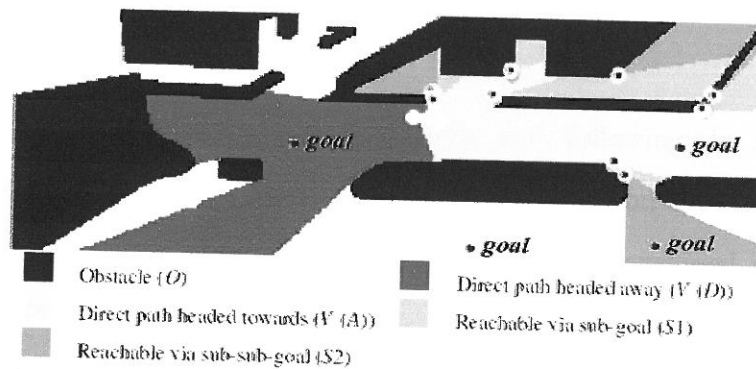


Figure 5.3: An example of the scene model used in Dee and Hogg. The pedestrian is represented by a white dot with a white arrow indicating motion direction: white dots with black centres are sub-goals. Obstacles are shown as black regions; areas invisible to the pedestrian are white. Areas shaded grey represent areas visible either directly or via sub-goals. From Dee and Hogg [51].

unusual routes to those chosen by a human operator.

The approach is quite different from others in that it is not based on statistics or novelty detection. Instead a psychological premise of goal-orientated behaviour is used to highlight inexplicable behaviour. This system would also allow an investigator to quickly query a video database with questions such as 'show me when someone enters that doorway, or gets into that car'.

Disadvantages of this approach are similar to those of Makris and Ellis. The layout of the particular scene is critical, meaning a good deal of work is involved each time the camera is moved. The question whether interesting or unusual behaviour can be defined by the centroid track of a pedestrian is also still an issue.



### 5.2.3 Bobick and Davis

A few approaches bypass the track entirely and, following the example of the human visual system, attempt behaviour recognition using the motion history of the target. Bobick and Davis (2001) [34] developed recognition based on *temporal* template matching. Using binary motion detection information ( $D(x, y, t)$ ) derived from background modelling, two templates are calculated. First, the binary Motion Energy Image (MEI) is the combination of a sequence of motion silhouettes taken over a temporal window,  $\tau$ . Second, a grayscale Motion-History Image (MHI) is the value-coded history of past silhouettes. If a pixel is in the current silhouette it is valued highest; pixels from previous frames are decremented:

$$\begin{aligned} MEI : E_{\tau}(x, y, t) &= \bigcup_{i=0}^{\tau-1} D(x, y, t - 1) \\ MHI : H_{\tau}(x, y, t) &= \begin{cases} \tau & \text{if } D = 1 \\ \max(0, H_{t-1} - 1) & \text{if } D \neq 1 \end{cases} \end{aligned} \quad (5.2)$$

Figure 5.4 shows the output of two aerobic exercises, with the binary MEI on the left and grayscale MHI on the right. These images are made scale- and view-invariant by computing a shape descriptor, and are matched to a database of standard actions.

Ultimately, this technique relies on binary detection information only, while studies of the HVS indicate that speed information is critical for behaviour detection. The motion templates are very specific to an exact be-

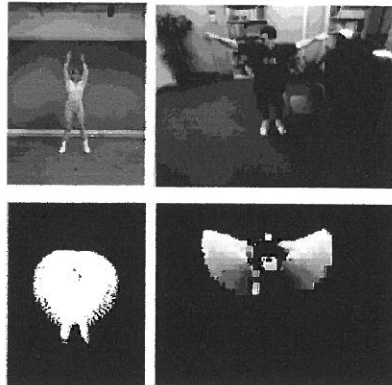


Figure 5.4: Aerobic performer and binary MEI (left); greyscale MHI (right). From Bobick and Davis (2001) [34].

haviour. The technique may work if someone is waving their arms, but if they are walking as well as waving their arms it will be unable to detect any similarity. This makes the approach unsuitable for CCTV applications.

#### 5.2.4 Stauffer and Grimson

Stauffer and Grimson [165], who developed the Gaussian mixture model for background modelling, also proposed a behaviour classification method. Each object is recorded with a list of data for each frame, including position, speed and direction of motion, and information on the motion silhouette such as size and aspect ratio. During training, all this information is then passed through an unsupervised hierarchical categorisation mechanism. During operation, objects are compared to these categories, with unusual events falling outside any category.

The authors note the importance of perspective on results, meaning that

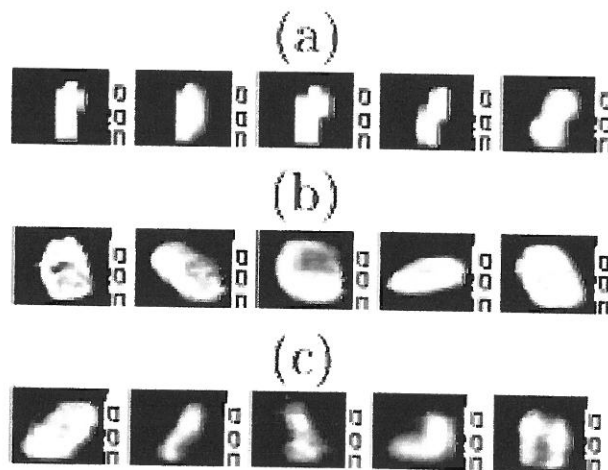


Figure 5.5: Stauffer and Grimson show the difficulties of categorising objects using shape. (a) Unambiguous pedestrians, (b) Vehicles. (c) Ambiguous cases. From Stauffer and Grimson (2000) [165].

a system trained on one view of a scene could not be expected to correctly categorise objects if the camera were moved. Also noted is the important case of pedestrians and vehicles. This approach did not consistently place each in a separate category (see Figure 5.5).

### 5.3 Motion signal analysis

Chapter 3 discussed the output of spatio-temporal filters. Equation 3.15 defined the filter output as being a function of both speed and contrast of a moving edge. The chapter described a number of ways to eliminate undesirable contrast dependence by combining the outputs of several different filters. An alternative to these methods is to normalise across the detected object. The filter equation can output either positive or negative values, but

in Section 3.3.3 only the absolute value was used for detection. Here, the original output values are utilised. Equation 5.3 calculates the ratio of the sums of positive and negative filter outputs across the whole detected object.

$$R_{obj} = \frac{\sum_{(i,j) \in Obj} |W_+(i,j)|}{\sum_{(i,j) \in Obj} |W_-(i,j)|} \quad (5.3)$$

The  $R_{obj}$  ratio is the *motion signal* of the object and has a number of useful properties. It is invariant to object size. It is also a function of the edge contrasts of the object and how it is moving. Thus, for a rigid object the signal remains approximately constant. For a non-rigid object such as a pedestrian, the signal will change as the pedestrian moves, as a direct result of the natural deformations involved in walking. In the figures below (Figures 5.7 and 5.8) the total motion amplitude is also computed. This number is a function of the size, speed and contrast of the object.

$$A_{obj} = \sum_{(i,j) \in Obj} |W_+(i,j)| + \sum_{(i,j) \in Obj} |W_-(i,j)| \quad (5.4)$$

$A$  is a sum of object motion intensity.  $R$  and  $A$  can be computed for the whole object or a subsection of it (see Section 5.4). This allows for computationally cheap categorisation of tracked objects.

Similar in concept to equation 5.3 is the video similarity measure developed in Syeda-Mahmood *et al.* (2005) [168]. This paper computed a threshold: the average velocity curves of an optical flow field of cardiac motion. The ratio of the number of vector components above the threshold to those below was then computed. The authors found that healthy hearts showed

sharp peaks in this ratio over time, while diseased hearts showed flattened peaks. The motion signal method differs in the use of scalar motion distillation inputs and the use of absolute values of inputs in the ratio, rather than the number of inputs exceeding a threshold.

The motion distillation stage filters video and generates motion channel data. These can be seen in the example in Figure 5.6 taken from a video of traffic. This figure demonstrates how the stationary features of the video, such as the road, trees, lampposts, are filtered out, leaving only the moving objects. In the motion channel images, positive and negative values are colour-coded as red and blue pixels, with pixel intensity indicating higher motion channel values. The black areas are due to a lack of motion, allowing for moving objects to be easily detected by thresholding. The object finding and tracking stages crop objects from these images, storing them, along with location information, as sprites<sup>1</sup>.

Figures 5.7 and 5.8 show examples of these cropped images. On top are object images cropped from a video sequence from the form and motion channels of the tracking systems. Note the size of the object changes as the object moves towards or away from the camera over time. The graphs show comparisons of  $A_{obj}$  (equation 5.4) and  $R_{obj}$  (equation 5.3) for two classes of object.  $A_{obj}$  of each object changes gradually with changing perspective as the object moves towards or away from the camera.  $R_{obj}$  of each object is radically different. The graph in Figure 5.7 shows an approximately constant value for  $R_{obj}$ . This data is taken from the rigid motion of a vehicle.

---

<sup>1</sup>A sprite was defined in Chapter 4.

Figure 5.8 is very different. Here the data is taken from a tracked pedestrian and  $R_{obj}$  oscillates with gait.

Analysis of the periodicity motion signal reveals whether the tracked object is moving rigidly as a vehicle or periodic non-rigidly as a pedestrian. Periodicity is determined by computing the number of times  $R_{obj}$  crosses '1' (i.e. when positive filter values become a majority). Other data computed are the rate of crossings (gait period), the amplitude of the signal peaks, and the standard deviations of these.

Table 5.1 shows quantitative results for classification of objects into rigid and periodic non-rigid using the periodicity of the motion signal. The test sample contained tracked objects from three different videos and a wide range of motion direction and viewing angles. Classification failed in only two cases giving an overall accuracy of 97%. One vehicle was mistakenly classified as a pedestrian. This was because the vehicle moved through a region of the scene with a number of deep shadows. This, combined with the dark colour of the vehicle resulted in over-segmentation and thus an oscillating motion signal. It should be noted that several other vehicles which passed through these shadows were correctly identified. The second failed classification was a pedestrian who walked briefly through one corner of the scene. The motion signal was too short to identify periodicity.

For vehicles little further information can be gleaned from the motion signal. A CCTV system might then use location or form information (vehicle colour or licence plate recognition perhaps) to further analyse the object. The motion signal for pedestrians contains a wealth of further information.

Table 5.1: Classification of objects into rigid cars and periodic non-rigid pedestrians showing then number of correctly matched objects using two tests: the periodicity of the motion signal and the width–height ratio test. Data was compiled from tracked objects in three videos.

Object	Sample Size	Width–Height	Motion Signal
Saloon	23	22	22
Van	2	1	2
Bus	1	0	1
M–Bike	2	0	2
Truck	1	1	1
Vehicle	29	83%	96%
Bike	1	1	1
Group	5	2	5
Walker	34	26	33
Runner	13	8	13
Strange	2	1	2
Pedestrian	55	69%	98%

Section 5.4 will demonstrate how this detailed information can be used to categorise the behaviour of the pedestrian.

These detection results can be compared to a simple discriminator based on the width to height ratio of a best fit bounding box around the object. By experiment, a ratio threshold of 1.5 was found to be optimum, with objects above this categorised as pedestrians and below as vehicles. This method correctly classified the vehicles from Table 5.1 with 83% accuracy and pedestrians with 69% accuracy. It was noted that the method failed for almost all buses, motorbikes and groups of pedestrians while the motion signal method was successful in all these cases.

Before we can move on to behaviour analysis a number of characteristics of the motion signal must be determined. Among these are the effects of viewing





Figure 5.6: Three sample frames from a traffic monitoring video (Top) along with the motion field output (Bottom) produced by the motion distillation stage. Figure 5.7 and Figure 5.8 below contain figures cropped from this data using the object finding and tracking stages.

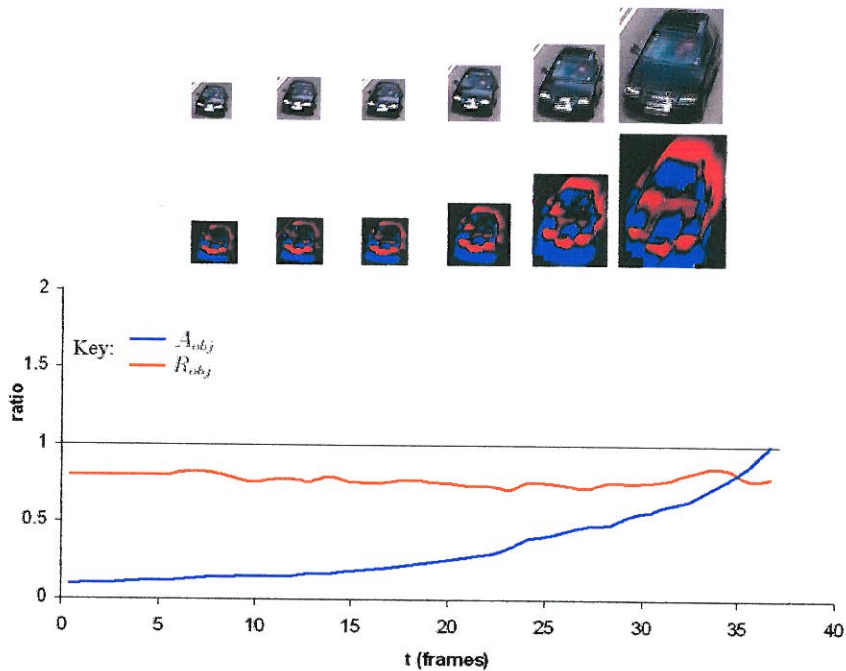


Figure 5.7: (Top) Cropped images of a rigid object (vehicle) from a sequence of frames of a traffic video along with motion output images. The graph shows the total motion amplitude ( $A_{obj}$  – equation 5.4) for this object over time, increasing because it is approaching the camera, while the *motion signal* ( $R_{obj}$  – equation 5.3) is approximately constant due to object rigidity.

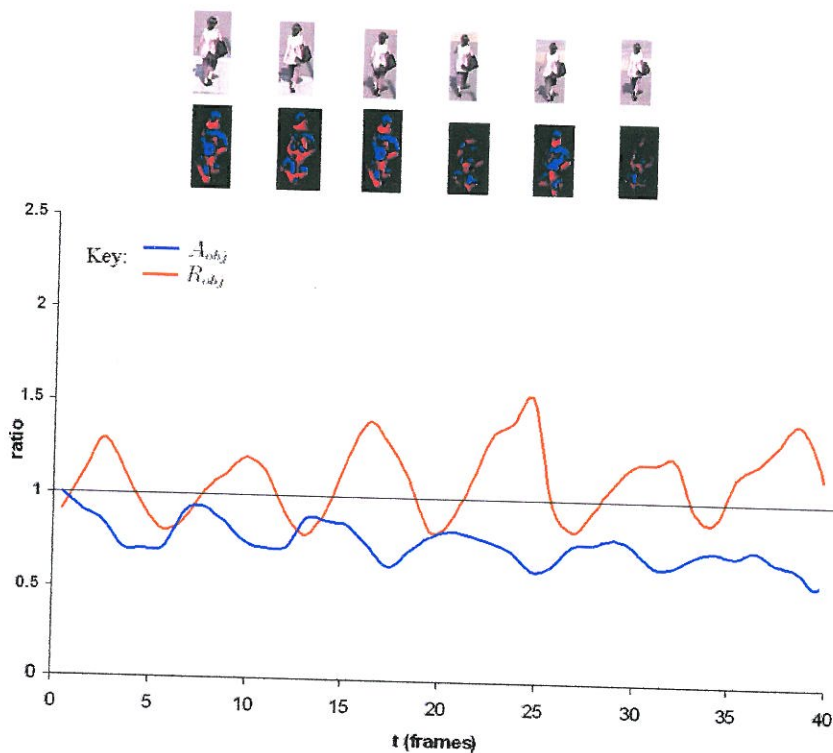


Figure 5.8: (Top) Cropped images of a non-rigid object (pedestrian) along with motion output images from a sequence of frames of a traffic video. The graph shows the total motion amplitude ( $A_{obj}$  – equation 5.4) for this object over time and the motion signal ( $R_{obj}$  – equation 5.3). The *motion signal* shows a pronounced cyclical response due to the gait of the pedestrian.

angle on the motion signal of pedestrians, the issues involved in accessing gait information, and the use of different filters in the motion distillation stage.

### 5.3.1 View independence

CCTV is recorded under largely uncontrolled conditions and may record pedestrian behaviour from any angle and distance. We would also like to establish the effects of changing viewing angles on the motion signal. Behaviour

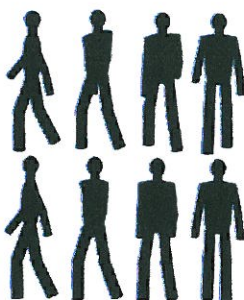


Figure 5.9: The animated pedestrian viewed from (left to right)  $\theta$  values of  $90^\circ, 60^\circ, 30^\circ, 0^\circ$ . Top row shows animation viewed from above at a vertical angle of  $\phi = 45^\circ$ , bottom row shows level view, i.e.  $\phi = 0^\circ$ .

analysis techniques should be independent of viewing angle and avoid lengthy learning or setup processes for different locations.

To test the method under controlled conditions, an animation of a walking pedestrian was prepared using the Blender software program.<sup>2</sup> This animation is purposefully simple and stripped of all texture and detail. It was then passed through the Motion Distillation process and the motion signal recorded for each frame. Animations were prepared at nine viewing angles. With the camera parallel to the ground ( $\phi = 0^\circ$ ), it was rotated around the walking figure and recorded at four angles,  $\theta = (0^\circ, 30^\circ, 60^\circ, 90^\circ)$ , where  $0^\circ$  is directly behind the figure and  $90^\circ$  is to one side. Four more animations were prepared at these horizontal angles but with the camera raised and looking down,  $\phi = 45^\circ$ . A final animation was prepared with the camera directly over the pedestrian's head,  $\phi = 90^\circ$ . Figure 5.9 shows detail of the animated pedestrian from each viewing angle.

<sup>2</sup>This subsection (5.3.1), along with Subsection 5.3.2 use simulated data. All other sections use real data.



Figure 5.10 shows graphs of the  $R_{obj}$  motion signal from the animated pedestrian, viewed from four different angles. The top graph is recorded from behind ( $\theta = 0^\circ$ ) on the flat and raised through vertical angles ( $\phi = 0^\circ, 45^\circ$ ). The bottom graph is from the side ( $\theta = 90^\circ$ ) on the flat and with raised angles. The change in vertical perspective has a strong effect on the amplitude of the motion signal due to different parts of the moving body being visible.

Figure 5.11 compares the  $R_{obj}$  motion signal from different horizontal viewing angles. It can clearly be seen that the horizontal viewing angle changes the phase of the motion signal by about one frame per  $30^\circ$ . This change in phase and the changing relative height of signal peaks are due to perspective effects.

Figure 5.12 shows how, even when viewed directly from above ( $\phi = 90^\circ$ ) detection of pedestrians by the motion signal is still clear and detectable.

The above test on simulated data shows that the motion signal is affected by view and direction of motion but that periodicity is still clear. This is true even for the case where the figure is moving directly away from the camera, when gait might be difficult to detect by traditional means. Analysis of real data (Table 5.1) proves that view does not inhibit categorisation into rigid and non-rigid objects.

View does have a strong effect on both the amplitude of signals and their phase. This complicates the recognition of gait and behaviour.

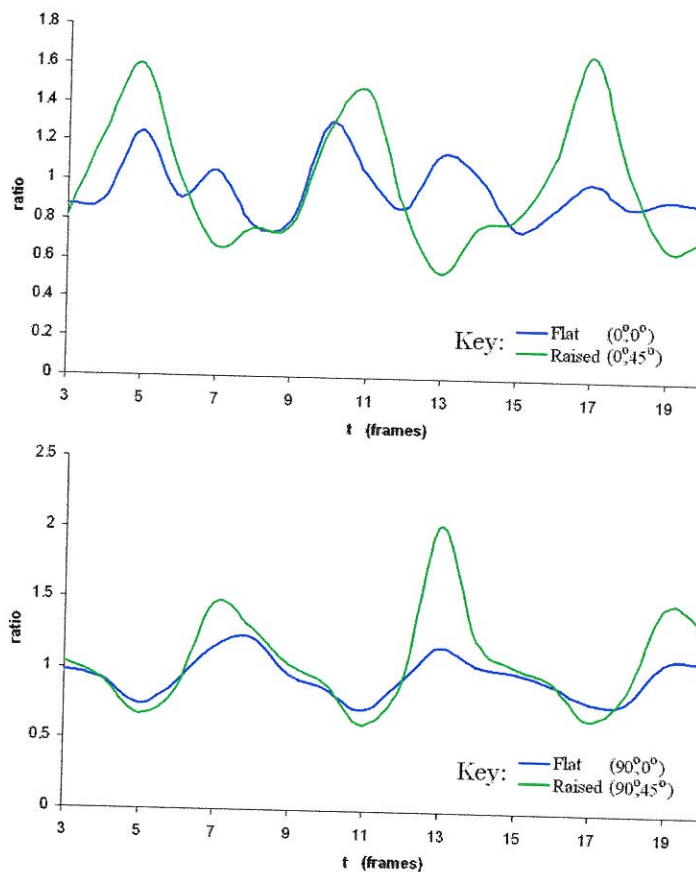


Figure 5.10: These graphs show the effects of vertical viewing angle on the motion signal. (Top) Signal from  $0^\circ$  view, horizontal and raised at  $45^\circ$ . (Bottom) Signal from  $90^\circ$ , horizontal and raised. (Using animation.)

### 5.3.2 Filter dependence

Does the choice of motion detection filter effect the motion signal? In Chapter 3 large filters with a Difference of Offset Gaussian (DoOG) profile were discussed. These filters have advantages of greater precision but at greater computational cost. The question arises whether the filters are better for motion signal analysis than the cheaper Haar wavelet decomposition. Fig-

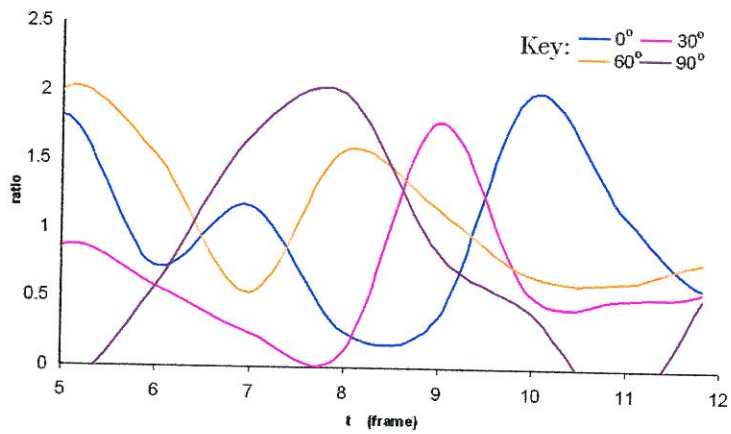


Figure 5.11: This graph shows how the phase of the motion signal changes with horizontal viewing angle. (Using animation.)

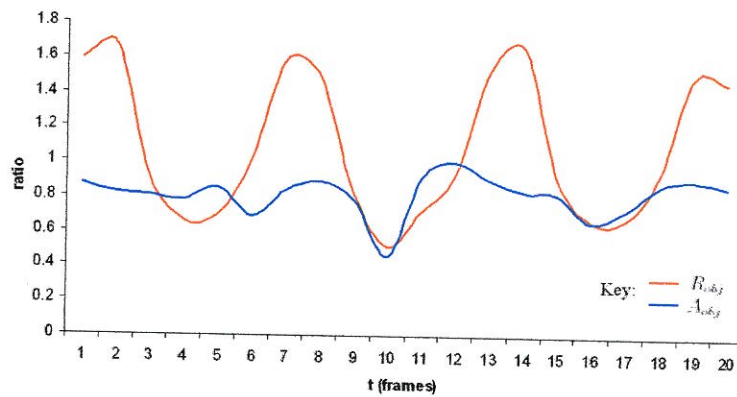


Figure 5.12: The  $R_{obj}$  motion signal of the animated pedestrian viewed from directly above ( $\phi = 90^\circ$ ).  $A_{obj}$  is shown for comparison. Gait information is still clearly visible. (Using animation.)

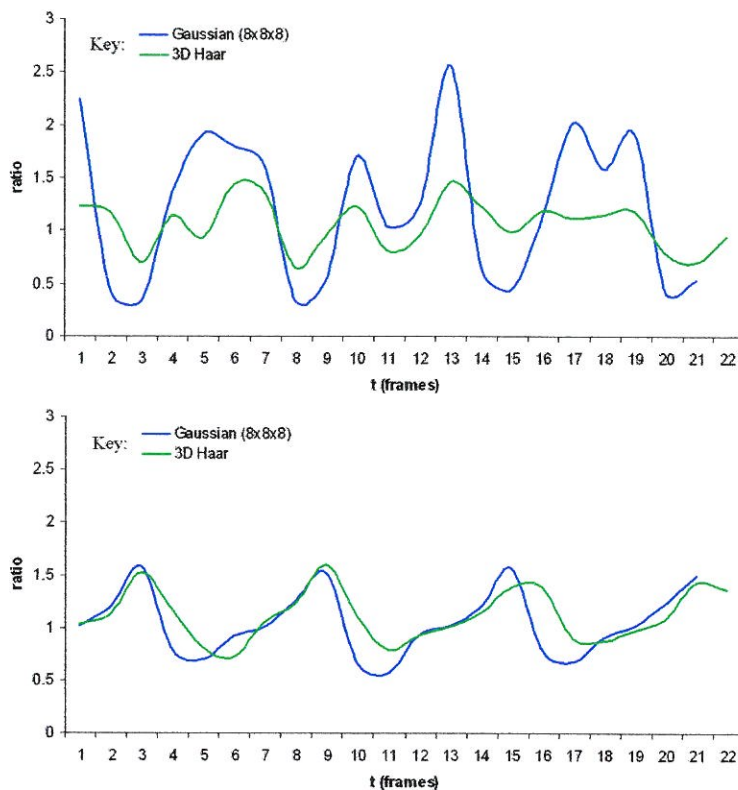


Figure 5.13: These graphs compare the motion signals generated by the  $s$ - $t$  Haar and DoOG filters in the motion distillation stage. The outputs of the two filters are presented for two different viewing angles, (top)  $\theta = 0^\circ$  and (bottom)  $90^\circ$ . (Using animation)

Figure 5.13 shows two graphs comparing the motion signals for the animated pedestrian viewed from two different angles; the top graph shows the motion signals as viewed from behind; the bottom graph shows the motion signals as viewed from the side.<sup>3</sup> The motion signals from Haar and Gaussian filters are quite similar, suggesting that the extra costs involved in the Gaussian filters give little extra benefit for behaviour analysis.

<sup>3</sup>This subsection (5.3.2), along with Subsection 5.3.1 use simulated data. All other sections use real data.



### 5.3.3 Gait

Classical gait detection uses silhouette extraction or model fitting to extract gait information. This is usually performed under controlled conditions with the subject walking across a prepared scene. Often different subjects must wear similar clothing to aid detection. Despite this, gait has been touted as a potential biometric for uses with CCTV [134]. The uncontrolled conditions in CCTV, and particularly problems of perspective and uncontrolled direction of walking with respect to the camera make classical gait detection techniques difficult or impossible. The motion signal proposed here might offer a practical alternative. The signal has a distinct advantage over other gait methods as it contains gait information and, while affected by viewing angle, has been shown to be accessible from any direction.

Figure 5.14 graphs the motion signal from two pedestrians, one walking near the front of the image and one running near the back. The top graph shows the uncalibrated signals, which are the raw input to the system. This is all that can be achieved without some knowledge of scene layout and perspective. It can be seen that there are few salient differences between the signals. In the lower graph, which has been scaled manually using scene information to account for perspective, the difference between the gait of the runner and the walker can clearly be seen. The effects of perspective and clothing would also have to be accounted for in a complete gait biometric recognition system.

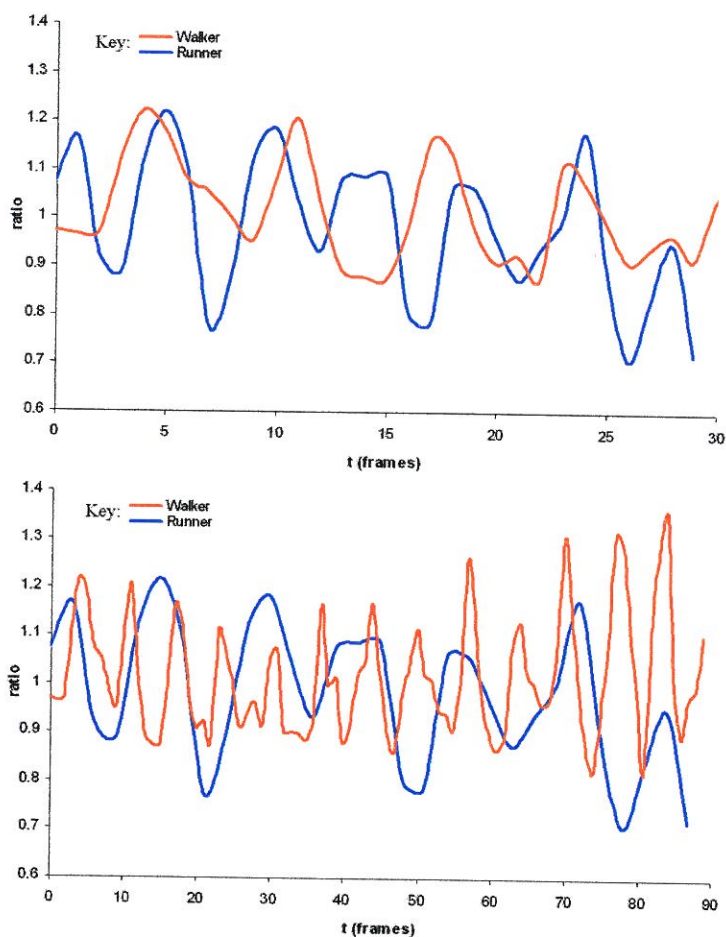


Figure 5.14: These graphs show the effect of scene perspective on gait information in the motion signal,  $R_{obj}$ . The runner was towards the back of the scene, while the walker was near the front. The top graph shows the original  $R_{obj}$  signal, while the lower graph has been manually calibrated using scene information. This data was produced using Video 2.

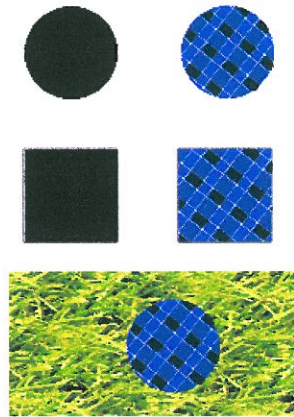


Figure 5.15: Top: untextured and textured moving circle. Middle: untextured and textured moving square. These four tests were simulated on a blank background. Bottom: Example of a textured circle on a textured background.

### 5.3.4 Shape and Texture

The method was also tested for sensitivity to shape and texture of the object. Four simulation videos were prepared; a monochrome circle, a textured circle, a monochrome square and a textured square. Each figure was made to move rigidly across a white background in a sinusoidal fashion, and processed for  $A$  and  $R$  values as before. The  $A$  values (normalised to account for slight size differences) are identical for the two untextured shapes ( $\pm 0.001\%$ ), while the two textured shapes are identical to each other, while showing a very slight difference ( $\pm 0.01\%$ ) to the untextured figures. (These slight differences are likely due to minor differences in pixelation of the simulated textured and untextured moving objects.) The  $R$  values for all simulations are identically 1 at all times as these are perfectly rigid moving objects.

The case of a textured background was also examined. A textured and untextured square and circle were simulated as above, but now a textured background was used. The difference observed between the textured and untextured circle on a textured background was  $\pm 1\%$ . The textured/untextured square and circle also showed a  $\pm 1\%$  difference.

The  $A$  and  $R$  values were compared with those from the untextured background test. The average values observed were similar in each test; however, a higher degree of noise was noted in textured background results, up to  $\pm 5\%$ . This is due to the output of the haar motion filter being a function of edge contrast, as stated earlier. The local edge contrast changes as the moving objects occlude different texel, resulting in some additional noise in the final  $A$  and  $R$  values. Figure 5.15 provides example images from the test simulations used.

## 5.4 Behaviour classification

Automatic classification of behaviours is a very difficult task. The human brain is so expert at this task that it is often startling to consider just how subtle the differences are between what might be thought of as distinct behaviours. This complexity means that automatic detection of a wide range of behaviours would almost certainly require a machine learning approach trained on a large database of sample behaviours. This sizeable task was deemed to be outside the scope of this thesis; however, much can still be achieved using deterministic analysis of motion distillation data.

This section presents techniques developed for analysing the motion signals which can correctly categorise the behaviour of more than 90% of pedestrians. Such a system could be used in a CCTV system to significantly reduce the workload for human operators.

The aim is to categorise pedestrian behaviours into the following types: *walking*, *running* and *unknown*. In addition, sometimes *groups* of people walking closely together are segmented as a single sprite<sup>4</sup>. Also, two special cases are detected: *waving hands* (or violent upper body movement) and sudden change in behaviour. These are included to demonstrate the rich detail available in the motion signal and its potential for full behaviour analysis using a machine learning approach.

The desired output of motion signal analysis is a signal characteristic of a particular abstract behaviour but independent of the identity of the subject pedestrian and of such complications as perspective, clothing, etc. One possible approach would be to analyse the motion channel output using moment approximations. Moments, which may be calculated using a series expansion of the equation below, are powerful shape descriptors, which may be independent of the size and largely insensitive to small rotations and distortions:

$$M_{pq} = \sum_{(x,y) \in I} x^p y^q I(x,y) \quad (5.5)$$

---

<sup>4</sup>As discussed in Chapter 3, it would be difficult to separate dynamically occluding objects without an object model.



where,  $I(x, y)$  is a function providing pixel data on the object and  $p$  and  $q$  are order number for the moment. However the key behavioural information of the motion signal lies not in the shape of the object silhouette but in the 'amplitude' of the motion signal. While the moments method might be modified for this task, research would be needed to determine how this might be achieved. It was decided to apply a more direct analysis approach as described below.

The initial rigid/non-rigid test identifies pedestrians (Section 5.3) and, under normal conditions, the vast majority of these will be walking. 'Walking' can be considered as a hypothesis to be tested. It was noticed during experimentation that the motion field of a walker is distinct in that it can usually be contained within a restricted area. To test the hypothesis, a rectangle is fitted onto the cropped motion channel data. The rectangle is scaled to the height of the object with a width equal to a fixed fraction of the height. The rectangle is positioned so that the sum of motion within its bounds is maximum. The motion sum outside the box,  $A_{ex}$ , is considered as a fraction of total motion:

$$\eta = \frac{A_{ex}}{A_{obj}} \quad (5.6)$$

Figure 5.16 shows the division into two regions; motion internal and external to the rectangle model. These regions are called *box* and *ex*.  $R$  and  $A$  signals are computed for each region.

To determine the optimum width of the rectangle, values of width ranging from 0.1 to 0.9 (as a fraction of rectangle height) were tested for a series of

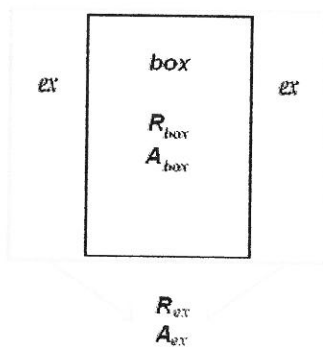


Figure 5.16: Applying the rectangle model to a pedestrian divides the sprite into two regions; motion inside the box is calculated as  $R_{box}$  and  $A_{box}$ . The areas at the left and right of the box are considered together; motion outside the box is calculated as  $R_{ex}$  and  $A_{ex}$ .  $\eta$  is calculated as the ratio of the  $A_{ex}$  to  $A_{box}$  region (see equation 5.6).

running and walking pedestrians. Data from a number of different videos and scenes were used (see Appendix A). Figure 5.17 shows  $\eta$  values for a series of pedestrians as the width of the bounding box is changed. The lower value solid line represents the average value for a set of walking pedestrians, with plus and minus standard deviations shown as broken lines. The higher valued lines show runners. There is a gap between the walkers and runners that allows the two categories to be distinguished. A rectangle width of half the height and a threshold value of  $\eta = 0.1$  was used to produce the classification results shown in Table 5.2.

As  $\eta$  and  $R$  are dimensionless, detection and recognition using these measures is independent of object scale and image resolution. Figure 5.18 shows how the motion distillation output for a walker will be enclosed to a high degree by the rectangle model.  $\eta$  is below the 0.1 threshold. This example is typical of walking pedestrians. This may be compared with the  $\eta$  value for a



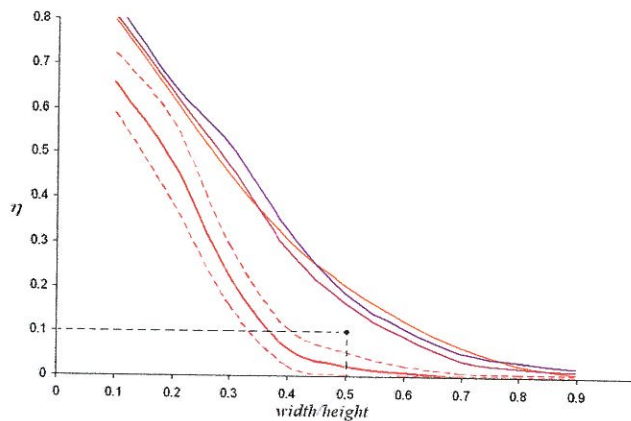


Figure 5.17: Graph of  $\eta$  vs. width for the rectangle pedestrian model. As the width decreases, the proportion of motion outside the model,  $\eta$ , increases.  $\eta$  is also a function of the type of motion. The lower solid line represents the mean value of a sample of typical walking pedestrians, the broken lines represent one standard deviation from the mean. These lines are separated from the higher value line which records runners. A threshold of  $\eta < 0.1$  with  $width = 0.5 \times height$  is used to distinguish walkers from runners.

running pedestrian in Figure 5.20 (top), where  $\eta$  is much greater than the 0.1 threshold. The  $\eta$  threshold test was evaluated for a number of videos of different scenes and a range of different pedestrian behaviours. Table 5.3 shows that 'walking' can be identified with a sensitivity of 96% and discriminability of 96% (see definitions on page 172).

Of special interest to CCTV operators may be cases where a pedestrian has been determined to be walking but the behaviour changes suddenly. Figure 5.19 shows such a case, where between frames 10-17, the  $\eta$  signal rises well above the 0.1 threshold. This was due in this case to the pedestrian suddenly jumping into the air. This unusual event can be distinguished from running, which also has a high  $\eta$  value, as the  $\eta$  signal for a runner is also

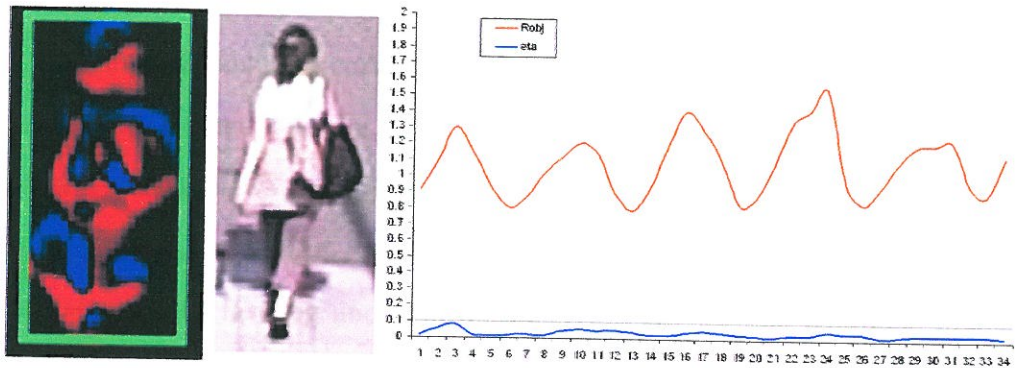


Figure 5.18: An example of the motion distillation output for a walking pedestrian (far left) beside the original frame (left). The pedestrian model rectangle is shown superimposed on the motion output. The graph compares  $R_{obj}$  (oscillating line) to  $\eta$  (low valued line).  $\eta$  is below the 0.1 threshold (shown as black line) allowing the system to categorise this pedestrian as a walker.

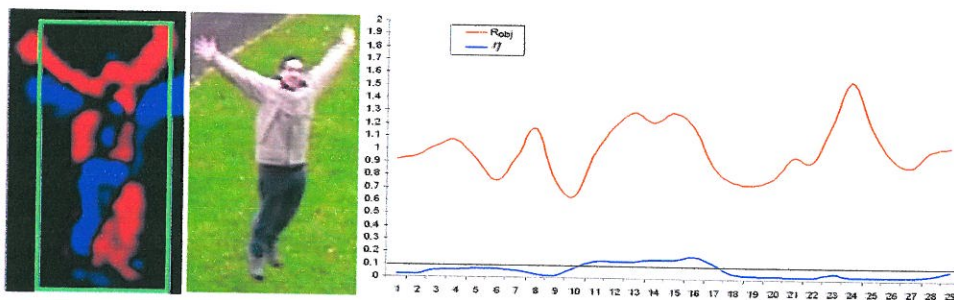


Figure 5.19: A pedestrian who jumps suddenly (between frames 10-17) causing  $\eta$  to increase above the 0.1 threshold (shown as black line).

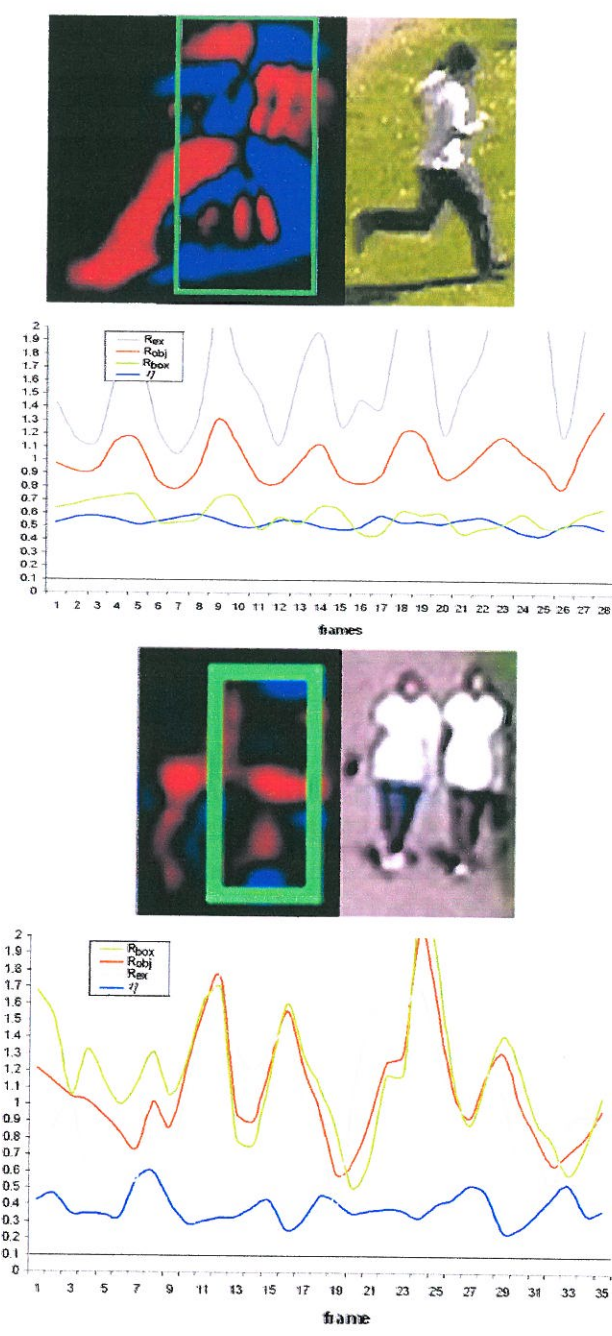


Figure 5.20: A comparison of the motion signals from a runner and a group. Both have a high and oscillating  $\eta$  signal. For the runner (top)  $R_{box}$  shows less variation in comparison to  $R_{ex}$  while they are similar for the group. Note that although the group shown here is more distant, and so of smaller apparent size, this has no effect on the recognition method.



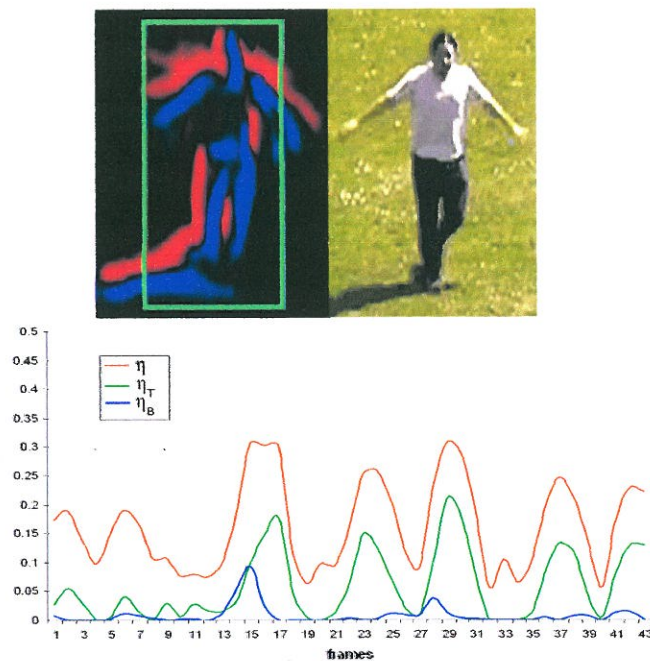


Figure 5.21: The motion signals from a walker who starts to wave.  $\eta_T$  shows large variation due to the movement of the pedestrian's upper body. Note: as this pedestrian is walking, the shadow is picked up in the motion channel. This does not unduly effect the final classification outcome as oscillations in the motion signal are sought and the shadow is relatively constant.

Pedestrians with a high  $\eta$  are tested for being runners or groups. Data was compiled from three different videos in which pedestrians moved in a wide range of directions with respect to the camera. View direction presented no problem for detecting groups or unusual behaviour. However the distinction between walking and running is difficult when the pedestrian is moving directly towards the camera. The majority of pedestrians were filmed naturally. However, for reasons of practicality, actors were used to provide all exemplars of running and unusual behaviour. Figure 5.22 shows the original raw data leading to Table 5.2, together with the decision tree paths that led

Table 5.2: A confusion matrix indicating classification of objects and behaviour using motion channel information. Columns represent true input types, rows are output classifications

	Walking	Unknown	Running	Group
Walking	23	1	0	0
Unknown	1	6	1	0
Running	0	0	14	1
Group	0	0	1	6

to the various classifications.

Table 5.3 details these results in terms of classification sensitivity and discriminability. Defining TP as true category population, FN as false negatives and FP as false positives, sensitivity is defined as  $TP/(TP + FN)$  and discriminability as  $TP/(TP + FP)$ .

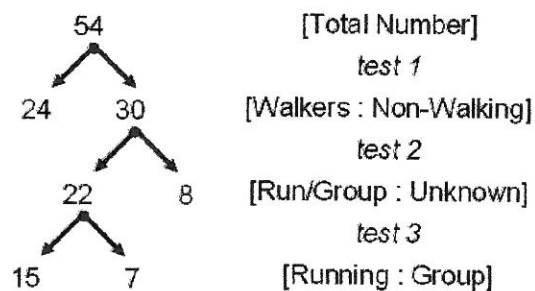


Figure 5.22: A decision tree of the behaviour analysis results. Test stages correspond to the pseudocode in Figure 5.23 (excluding Test 0: the rigidity test). Numbers correspond to the algorithm output (rows) shown in Table 5.2.

Figure 5.23 provides pseudocode for behaviour classification by motion signal analysis. There are three stages to the approach <sup>5</sup>. First, the motion

<sup>5</sup>Note: the last two stages of Figure 5.23 correspond on a one-to-one basis with the

Table 5.3: Classification results for the object types in Table 5.2 in terms of sensitivity and discriminability.

Activity	Sensitivity	Discriminability
Walking	96%	96%
Unknown	86%	75%
Running	88%	93%
Group	86%	86%

**Detect and track object**  
(see Chapters 3 and 4)

**Test 0: Rigidity test:**

Test  $R_{obj}$  for periodicity  
 if periodic, categorise as *Pedestrian*  
 else, categorise as *Vehicle*

**Test 1 : Behaviour classification:**

if Pedestrian, apply rectangle model  
 Test  $\eta < 0.1$   
 if true, categorise as *uninteresting pedestrian behaviour*  
 if false,  
     Test 2:  $\eta$  for peaks  
         if false, categorise as *unknown pedestrian behaviour*  
         if true, attempt advanced classification.

**Advanced classification:**

Test 3:  $R_{ex} > R_{box}$   
 if true, categorise as *runner*  
 if false, categorise as *group*

---

Figure 5.23: Pseudocode for the object classification using motion signal analysis. The method has three stages: a rigidity test to distinguish vehicles from pedestrians; the  $\eta$  threshold test to distinguish uninteresting walking from interesting unusual behaviour; the final stage serves to demonstrate the potential of the motion signal analysis approach for advanced behaviour classification.

signal  $R_{obj}$  is analysed for periodicity to distinguish rigid vehicles from non-rigid pedestrians. Second, a rectangular model is applied to pedestrians to detect any potentially unusual behaviour. In the videos analysed, about 90% of moving objects can be accurately categorised as either vehicles or uninteresting walking pedestrians. Such a determination could be used to direct the attention of a CCTV operator or to annotate video to facilitate easy search at a later date. The results for unusual behaviour detection show a relatively high false positive level. This would be acceptable in the context of a CCTV application where the avoidance of false negatives is more critical. The final stage, advanced classification, attempts to further categorise the remaining 10% of possibly interesting objects. The algorithm offered for this stage is intended to demonstrate the potential of motion signal analysis for full classification. However, a complete classification system would require a machine learning approach, coupled to a large database of behaviour data, due to the large number of behaviour types in real world CCTV. Such a system was judged to be outside the scope of this work.

## 5.5 Summary

This chapter covered one of the most important and difficult problems in visual surveillance – the task of classifying moving objects and identifying human behaviours. Although the HVS is expert at this task it is far from fully understood how this is achieved – but it is known that many behaviours

---

various stages of the decision tree of Figure 5.22



are detected directly from the motion channel information.

Several classification methods described in the literature focus on the object track alone. This procedure has a number of disadvantages, including perspective dependence and a requirement for lengthy set up and learning times for each camera location. Coupled to this, studies of human CCTV operators indicate that the track is rarely used in practice to determine behaviour (but see Dee and Hogg, 2004 [51]).

This chapter has proposed a new behaviour analysis method. Information is accessed from the motion channel directly and a number of motion signals are computed for tracked objects. These signals can be used to classify (rigid) vehicles and (non-rigid) pedestrians, which, when tested under a wide range of conditions and different videos, achieved a correct recognition rate of approximately 97%. Use of object level information in this way can be viewed as an application specific remedy for the aperture problem which arose from the use of local motion detection filters in Chapter 3.

Using an animated pedestrian, this method has been shown to be view- and perspective-independent and so requires no training or bootstrapping for different videos or locations. Further, pedestrians can be classified into five behavioural categories.

The visual surveillance system described in this thesis comprises three parts: motion detection using motion distillation, tracking using dual channel tracking and behaviour analysis as discussed here. In the past, visual tracking science has developed along *ad hoc* lines, using what works with little concern

as to why it works. The next chapter will present a new theoretical framework which integrates these components.

## Main points

The main points and achievements of this chapter are:

- *Motion signals*, extracted directly from the motion channel, can be used to classify behaviours.
- $R_{obj}$  is approximately constant for rigid vehicles and oscillatory for non-rigid pedestrians.
- As the values  $R_{obj}$ ,  $R_{ex}$  and  $\eta$  are dimensionless, detection and recognition are independent of the object scale and image resolution.
- Recognition by this method is view independent for vehicles, pedestrians and unusual behaviour.
- Gait information is present but calibration is needed.
- Assessing pedestrians using rectangles allows unusual behaviours to be detected.

## Chapter 6

### Discussion

This thesis has developed an automatic visual surveillance system consisting of three main components; motion detection through Motion Distillation, tracking through a dual-channel form-motion architecture, and behaviour analysis through the processing of motion signals derived from the motion distillation outputs. This chapter discusses general issues arising from this and the limitations of it.

#### 6.1 Motion Distillation

Chapter 3 describes how motion detection may be achieved through spatio-temporal wavelet decomposition, a process I have termed Motion Distillation. Distillation using the  $s$ - $t$  Haar wavelet is highly computationally efficient and detection is robust. While the output of statistical background modelling is

binary detection (Figure 3.16 shows why this is the case) the output of Motion Distillation is non-binary. This, coupled with increased robustness, are the chief advantages. The extra non-binary information can be used directly for behaviour analysis, as discussed in Chapter 5.

The output of a single  $s-t$  filter, such as the Haar, is a function of edge speed and contrast (See Subsection 3.3.3). Although this is sufficient for object detection, applications such as behaviour analysis require the contrast dependence to be removed. Three different methods for achieving this have been presented in this thesis: using three differently orientated symmetrical filters to derive local edge velocity (speed and direction) information (Subsection 3.4.1); using two non-symmetrical cuboid filters to derive local speed information (motion direction and contrast independent, see Subsection 3.4.2); and finally, in Chapter 5, normalising across an object to derive a contrast free motion signal for the object. The velocity or speed information produced by the first two methods may have a role in behaviour recognition but it is expected that this would probably only be practically utilised by a learning algorithm.

Here, Motion Distillation was implemented on a standard serial computer. In the HVS the motion channel is highly parallel with vast numbers of  $s-t$  responsive neurons operating simultaneously and locally in the visual field. One can envisage a parallel implementation of Motion Distillation in hardware where a separate parallel processor computes  $s-t$  wavelet decomposition for each pixel location. How best to feed this information to later stages and whether to tackle tracking and behaviour recognition in a parallel or serial

processor are questions for future research.

## 6.2 Tracking

Chapter 4 presented a tracking architecture using two information channels – the motion channel produced by motion distillation output and the form channel containing the current appearance of the tracked object. By using each channel when that data is most appropriate – the motion channel while the object is in motion and the form channel to resolve ambiguities – the system overcomes some of the limitations of other tracking paradigms. The algorithm was tested using a range of video types and scenarios and demonstrated to track robustly.

This approach cannot be applied to all situations. In dense crowd situations, where the moving crowd might fill the camera view, the system will cease to track (actually, in this case the shake protection system will switch off tracking to prevent overload). For crowds a modified approach is required, depending on the application. One goal might be to track the motion of the crowd as a whole, or to find pockets of unusual motion which might indicate that someone has fallen or has been crushed. Here, a solution should rely on optic flow information, perhaps generated by the DoOG method. Alternately, if the goal is to track individuals within the crowd, use of an appearance model and face tracker would be appropriate. A dense flowing crowd situation is a failure of the ‘small objects’ assumption stated in Chapter 1. Alternately, it could also be viewed as a failure of the static

camera assumption, because if the majority of the field of view is in motion the result will be the same as for a moving camera.

Aside from these niche applications, the Dual-Channel method has been shown to successfully track arbitrary objects in a wide variety of surveillance videos.

### 6.3 Behaviour

Behaviour Analysis in this system works directly on motion signals derived from the output of the motion distillation module. These signals are robust to changes in perspective and viewing angle and so avoid restrictive scene models and costly bootstrap routines.

Chapter 5 explains how motion signals can be computed and how they can be used to categorise objects based on their behaviour. Recognition rates range 75–97% for a selection of different behaviours. Motion signal analysis does not use shape analysis (as Stauffer and Grimson (2000) [165]), object tracks (as Malik *et al.* (1995) [116]) or spatio-temporal feature extraction (as for the recent work of Shah's group [6, 190]). Instead two dimensionless activity measures,  $\eta$  and  $R$ , are used which are independent of object scale and image resolution. This approach saves considerable computational expense when compared to other methods.

The motion signals are analysed in a deterministic fashion and this naturally limits the number of detectable behaviours to those which can be

programmed by hand. In this implementation, only Haar wavelet outputs are used in motion signals. Other wavelet systems which output optic flow and speed information may be more useful for behaviour analysis.

A learning algorithm, trained using a large database of different behaviours and actors can be expected to improve results and extend the number of behaviour categories. The more complex information produced by DoOG wavelets would be fully utilised only with an automatic learning algorithm. This approach was judged to be outside the scope of this work and is left as a suggestion for future research.

Giese and Poggio (2003) [75] postulate that behaviour recognition in the HVS uses form, as well as motion, information. Expansion of my algorithm to include this information, possibly a body pose database, would also be best achieved using a learning approach.

## 6.4 Context

Figure 2.1 organises visual surveillance methods according to how video data is processed to achieve motion detection; how tracking is achieved; and by which surveillance goal is pursued. In terms of this scheme, the Motion Distillation and Dual-Channel tracking paradigm follows the middle, 3D, route.

Motion Distillation is similar to Bårman et al. (1991) [16] and Knutsson et al. (1992) [100]. Differences include the choice of  $s-t$  filters (tensor cones



and Gabor filters as opposed to the Haar and DoOG used here) and the use here of wavelet decomposition. Also, Bårman and Knutsson were concerned with medical imagery, not visual surveillance and tracking.

The dual-channel tracking paradigm developed here is a novel approach with few parallels in the machine vision literature. The concept is derived from neurological models of the HVS. In 1978 Martin and Aggarwal [119] discussed the dual, peripheral versus attentive, structure of the eye, where peripheral vision emphasises motion detection and the attentive center emphasises appearance analysis. In 2003 Giese and Poggio [75] suggested a dual-channel model to explain the HVS's ability to recognise complex human activities. Neither appearance information nor motion analysis can account of this alone.

The approach to surveillance and behaviour analysis developed in this thesis is also new. Selinger and Wixson (1998) [156] use periodic deformations of object silhouettes to distinguish rigid vehicles from non-rigid pedestrians. This is similar in effect to my method, although here the periodic motion signal is derived directly from the motion channel, not through object shape information. It would be more difficult to extend Selinger and Wixson's method to pedestrian behaviours. Bobick and Davis (1998) [34], used the 'motion history image', which is also similar, but again, the non-binary nature of the motion channel grants greater flexibility. Bobick and Davis's is not size or view independent, for example. This system does not use the popular track modelling approach of Makris and Ellis [115, 114] but this might be profitably used in conjunction with my approach.

## 6.5 Theory and synthesis

In Chapter 1 a number of questions were raised as to the nature and function of various components of visual surveillance systems. Drawing from the work presented in previous chapters, some tentative answers can now be suggested:

*What is the appropriate role of motion detection in tracking? What is the best way to achieve it?*

Appearance-based techniques such as particle filters require an appearance model of the object to be tracked. Techniques that include a motion-detection stage can track without this model, allowing arbitrary objects to be tracked. Conversely, objects only become detectable while in motion; when they stop moving, only comparison with a stored appearance model can detect them. We can also surmise that in a surveillance application all new objects will enter the scene in motion, and continue to move for some time before they stop. This provides a window of opportunity to detect the new object and acquire its appearance model for later use in case of stopping or occlusion.

It is also known that primitive animals can only detect prey while it is in motion. Motion detection probably evolved from the most primitive light and dark sensing cells when animal brains became powerful enough to correlate spatially coherent changes.

Chapter 3 showed the deficiencies of the classic method of achieving this, namely background modelling. A new paradigm, Motion Distillation, was developed which may well prove to be the best way to achieve motion detec-

tion.

The pixel process calculations of background modelling cannot distinguish noise or isolated change from spatially coherent motion. This can only be achieved by an additional 2D object finding step. The spatio-temporal 3D technique discussed in Chapter 3 can achieve true motion detection in one step.

*Are approaches which avoid motion detection valid and what assumptions do they make?*

Those methods which forgo motion detection (frame-based methods) are deficient in that they require *a priori* appearance information. During tracking, these methods depend solely on location prediction and the clutter resistance of the appearance model. Clutter can be understood as detection noise. A perfect appearance model (AM), which will uniquely detect the target object, will return no clutter. In real-world systems clutter is a problem. Particle filters compensate for clutter-prone AMs by maintaining multiple hypotheses, but at significant computational cost. The work of this thesis deals with this problem by relying on the motion channel while objects are in motion and when the objects are likely to be subject to changes in appearance.

*Is location prediction necessary? What issues arise with appearance models in visual tracking and how does this relate to other aspects of the system?*

Visual tracking systems can be viewed as having three components in various proportions – a motion-detection stage, an appearance model stage, and a

prediction stage: if any two of these components work perfectly, the third is unnecessary. (Tracking systems often contain an update stage, here this is considered to be part of the appearance model.)

If the system has perfect motion detection, and perfect prediction while the object is occluded, there is no need for an appearance model to confirm the object's identity on reappearance. If the appearance model is perfect (i.e. produces no clutter or false positives) and the location prediction is perfect, then tracking will continue perfectly without any need for the crutch of a motion detection stage. This is the aim in particle filtering systems. Finally, if motion detection is perfect and the appearance model is perfect, then prediction is unnecessary as an occluded object which reappears elsewhere in the scene will be detected by its motion and perfectly matched by the appearance model to its original identity.

In the real world there are no perfect components. This answer can be used as a guide to direct improvement efforts. On which stage would efforts to improve the system as a whole be best employed? For a background modelling approach, should improvements be directed towards the appearance model to reduce clutter, or towards the prediction module to reduce search area? Some methods such as particle filtering contain no motion-detection stage. Would including one give more profitable returns than improving prediction or the appearance model?

The work of this thesis has emphasised motion detection and the appearance model, while limiting prediction to search constraints and a series of logical sorting rules. Results show the method to be quite robust for the fair

range of videos tested. The strength of our spatio-temporal motion detection method allows us to avoid using a complex appearance model. Work on this module may improve performance in longer videos with large numbers of objects.

Figure 6.1 illustrates a theoretically complete visual surveillance system. Information flows from both the motion channel on top and the form channel on the bottom. The blob sorting logic described in Chapter 4 is represented by the MD and AM components which together output a *Track*. Finally, the BA component accesses information directly from both the motion and the form channel, and the object track.

This figure represents an ideal system, which the work of this thesis aims to emulate. In this work, the BA component only accesses information from the MD component. Tracking information is also implicitly used by the BA as it maintains the object identity throughout analysis. Future work, possibly focused on learning-based BA algorithms, will be required to properly connect the BA to the other components as shown in the figure.

## 6.6 Suggestions for future research

This thesis has presented work on Motion Distillation and Dual-Channel tracking. However, much work remains to be done extending the operational envelope of the system. For now it seems that incorporating a behaviour learning algorithm to motion signal analysis would be most profitable. On the evidence presented in this thesis recommendations for future research

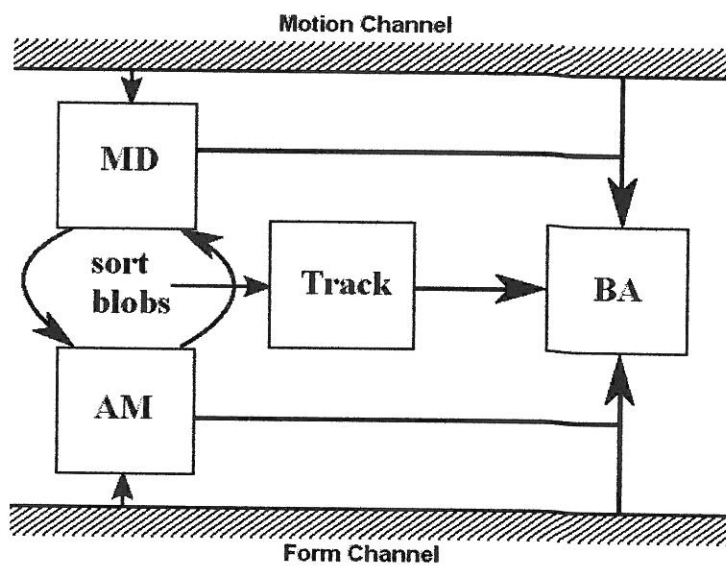


Figure 6.1: A schematic of a theoretically complete dual-channel tracking and behaviour analysis system. MD and AM are motion detection and appearance model respectively. BA is behaviour analysis.

are:

- Develop a hardware implementation of Motion Distillation. Parallel pixel processors, each performing local  $s$ - $t$  wavelet decomposition, would massively increase speed and performance.
- Extend the dual-channel approach using improved appearance models. Of particular interest would be performance for large numbers of objects and crowds.
- Develop a database of sample behaviours for training a behaviour analysis learning algorithm. The use of appearance information in behaviour recognition should also be explored.



## Chapter 7

### Conclusions

This thesis has explored the theory and practicalities of automatic visual surveillance. The research was inspired by neurological models of the Human Visual System.

Primary achievements of this work include a new motion detection paradigm, a new dual-channel tracking architecture and a novel behaviour analysis method. This thesis has also developed a new theory of visual tracking.

Motion Distillation achieves motion detection through spatio-temporal wavelet decomposition of video. This method is both more computationally efficient and more robust to noise than traditional methods such as background modelling.

The dual-channel tracking architecture provides a method of organising multiple information channels for tracking. The motion channel, the output of motion distillation, is used for detection. The form channel is used

to resolve ambiguities and occlusions. A combination of the two channels achieves tracking and overcomes some of the weaknesses of traditional methods. Detailed quantitative and qualitative results demonstrate the power and reliability of this approach.

Behaviour analysis is also approached in a novel manner. Information from the motion channel is accessed directly and *motion signals* are extracted and analysed. Object classification based on behaviour is demonstrated with a success rate from 75% to 97%, depending on the complexity of the behaviour to be detected.

# Appendix A

## Video data

### A.1 List of videos

The algorithms presented in this thesis were tested on a wide variety of video data. More than 20 videos were used. For brevity, the number presented here was limited to a smaller example set while including those which gave the most interesting and distinctive results.

A list of the videos discussed in the text is provided below. Videos are categorised by their dimensions, quality, content and level of noise.

#### Video 1

Outdoor in diffuse natural lighting; a pedestrian walks across the scene and back again. The video is recorded on a Logitech Webcam.

## **Video 2**

Outdoor with strong sunlight and shadows; a pedestrian repeatedly crosses the scene while walking, running, jumping, waving hands, starting and stopping. This video was recorded on an interlaced Canon MV700 Camcorder.

## **Video 3**

Outdoor with diffuse natural lighting; windy conditions create a lot of motion clutter. Two pedestrians walk and run several times, meeting and reversing direction, meeting and continuing and passing each other without stopping. This video was recorded on an interlaced Canon MV700 Camcorder.

## **Video 4**

Outdoor video of complex traffic scene with changing natural lighting, including strong sunlight and shadows. Scene includes vehicles and numerous pedestrians. This video was recorded on an interlaced Canon MV700 Camcorder.

## **Video 2**

Outdoor with strong sunlight and shadows; a pedestrian repeatedly crosses the scene while walking, running, jumping, waving hands, starting and stopping. This video was recorded on an interlaced Canon MV700 Camcorder.

## **Video 3**

Outdoor with diffuse natural lighting; windy conditions create a lot of motion clutter. Two pedestrians walk and run several times, meeting and reversing direction, meeting and continuing and passing each other without stopping. This video was recorded on an interlaced Canon MV700 Camcorder.

## **Video 4**

Outdoor video of complex traffic scene with changing natural lighting, including strong sunlight and shadows. Scene includes vehicles and numerous pedestrians. This video was recorded on an interlaced Canon MV700 Camcorder.

Name	Dim.	Noise	Type	Source
Video 1	320×240 px ×700 frames	Low Outdoor	Prog. Scan Colour	Webcam



Name	Dim.	Noise	Type	Source
Video 2	720×576 px ×5458 frames	Medium Outdoor	Interlace Colour	Camcorder



Name	Dim.	Noise	Type	Source
Video 3	720×576 px ×1623 frames	High Outdoor	Interlace Colour	Camcorder



Name	Dim.	Noise	Type	Source
Video 4	720×576 px ×8245 frames	Medium Outdoor	Interlace Colour	Camcorder





### **Shake Video 1**

Outdoor video with diffuse lighting for a pedestrian. Camera shakes intermittently. This video was recorded on an progressive scan Canon IXUS 400 digital camera.

### **Shake Video 2**

Indoor video with diffuse mixed artificial and natural lighting. Camera shakes intermittently. This video was recorded on an progressive scan Logitech Webcam.

### **Home Office (i-Lids pre-release)**

Outdoor video with diffuse natural lighting. Pedestrian repeatedly walks near a security fence.

### **CAVIAR (Fights & Runs Away)**

Indoor video with mixed direct sunlight and artificial lighting; several moving and stationary pedestrians, as well as motion clutter. Two pedestrians interact in a 'fight' scene.

Name	Dim.	Noise	Type	Source
Shake	320×240 px	High	Prog. Scan	Camcorder
Video 1	×243 frames	Outdoor	Colour	



Name	Dim.	Noise	Type	Source
Shake	160×120 px	Low	Prog. Scan	Webcam
Video 2	×278 frames	Indoors	Greyscale	



## A.2 List of video sources

- **Webcam:** Logitech QuickCam Zoom Webcam, CMOS Optical Sensor  
320×240px or 160×120 AVI.
- **Digital Camera:** Canon IXUS 400, Focal Length 5.4mm–10.8mm,  
320×240px at 15fps AVI.
- **Camcorder:** Canon MV700, Mini DV, 18× Optical Zoom, Focal  
Length 2.8mm–50.4mm

Name	Dim.	Noise	Type	Source
Home	360×288 px	Low	Prog. Scan	Unknown
Office	×8773 frames	Outdoors	Greyscale	



Name	Dim.	Noise	Type	Source
CAVIAR	384×288 px	High	Prog. Scan	Unknown
	×550 frames	Indoors	Colour	



## Appendix B

### Object labelling

Finding and labelling the objects after the binary detection mask has been produced is still an expensive step. Positive pixels must be examined for their connectivity to neighbouring positive pixels. The mask must be scanned multiple times and the neighbourhood of each pixel examined. A 'label space' copy of the mask is often required to store the results. Each connectivity check requires  $9N$  pixel operations, where  $N$  is the number of pixels in the mask. There are a number of different variations on the labelling algorithm. Here I used a version described by Davies (2005) [49] with slight modifications to keep the number of passes to two and avoid the need for a new label space storage area in memory.

The prior threshold process outputs a label space with motion pixels marked as '0' and non-motion pixels labeled as '-1'. However, because of the way this is stored in memory, there are 126 free positive levels remaining. In the first pass, the neighbourhood of each pixel is examined for the highest

value label value. Pixels labeled as  $-1$  are ignored and their neighbourhood is not examined, reducing computation in sparse masks.

An integer variable called 'counter' is maintained, which starts at  $+1$  and is incremented whenever a new object is encountered. Each motion pixel encountered is labeled as the lowest local label value or as the 'counter' value, if all neighbouring pixels are labeled as  $0$  or  $-1$ . This step requires  $9N$  pixel operations.

If all objects were rectangles, then this would be enough. However, irregular shapes with spurs will cause the system to label different parts of the same object with different label numbers, and this is unavoidable. So after the current pixel has been labeled, any label conflicts in the local neighbourhood are recorded in a 'collision index'. This step requires  $8N$  pixel operations, as the current pixel need not be read again.

When this pass is completed, the collision index is processed; in the final pass the highest object label replaces the others. This step requires  $N$  pixel operations, as the neighbourhood need not be accessed again. Figure B.1 provides pseudocode for this method.

Pixel operations are the slowest type of operation in the program due to the multiple memory and disk accesses. This object labeling method requires a total of  $18N$  operations. However,  $N$  may still be a large number when dealing with video, and it is clear that reducing  $N$  will give great benefits. This is achieved using spatio-temporal scaling described below (Section 3.3.2).

---

**Threshold** to '0' (motion) or '-1' (no motion)  
*counter* = +1

**First Pass**

if *pixel label*  $\geq 0$   
  Access pixel neighbourhood  
  *pixel label* = min(*counter*, neighbourhood)  
  Update 'collision index' using neighbourhood  
  if *counter* was used, increment

**Sort collision index**

**Second Pass**

  Replace each pixel label with number determined by collision index  
  Copy pixel into sprite

---

Figure B.1: Pseudocode for the object labeling method. This approach reduces pixel operations to  $18N$ .



## Appendix C

### Glossary of acronyms

- 1D+2 – 1D process followed by a 2D process
- 2D+1 – 2D process followed by a 1D process
- AM – appearance model
- BA – behaviour analysis
- CCTV – closed circuit television
- EKF – extended Kalman filter
- EM – expectation maximisation
- FP – false positive
- FN – false negative
- GMM – Gaussian mixture model
- HVS – human visual system
- LGN – Lateral Geniculate Nucleus

- LP – location prediction
- MD – motion detection
- MT – middle temporal
- pdf – probability density function
- TB – true blobs
- TBP – transient background problem
- TMF – temporal median filter
- SBP – stationary background problem
- s-t – spatio-temporal
- UKF – unscented Kalman filter

## Appendix D

### Author's Publications

- M. Sugrue and E.R. Davies. Towards Pedestrian Tracking in CCTV video for Crime Detection, *RE:HVS Basic Technology Project Open Day*, Imperial College, No. 30, 2004. [Poster]
- M. Sugrue and E.R. Davies. Tracking in CCTV Video Using Human Visual System Inspired Algorithm, *The IET Visual Information Engineering 2005*, Glasgow, 4-6 April 2005.
- M. Sugrue and E.R. Davies. Motion distillation for pedestrian surveillance, *The Sixth IEEE International Workshop on Visual Surveillance*, Graz, May 13, 2006.
- M. Sugrue. Motion Distillation, *BMVA One Day Tech. Meeting, Detection vs. Tracking*, July 5th 2006. [Talk]
- M. Sugrue and E. R. Davies. Motion Distillation, *Irish Machine Vision and Image Processing Conference 2006 (IMVIP'06)*, 30 Aug-1 Sept

2006, Dublin City University. [*\*Winner of Best Poster prize*]

- M. Sugrue and E. R. Davies. Motion detection and tracking by mimicking neurological dorsal/ventral pathways, Chapter in *Reverse Engineering the human vision system: next generation artificial vision systems*, Editors: Anil Bharath and Maria Petrou, with publishers, due 2007.
- M. Sugrue and E. R. Davies. Motion signals provide rapid discernment of pedestrians and pedestrian behaviour. *Electronics Letters*, 43(23):1267–1269, November 2007.
- M. Sugrue and E. R. Davies. Contrast independent motion detection using ‘inverse pair’ spatiotemporal edge detectors. *Electronics Letters*, 43(24):1346–1348, November 2007.

## Bibliography

- [1] P. S. Addison. *The Illustrated Wavelet Transform Handbook*. Institute of Physics, London, July 2002.
- [2] J. K. Aggarwal and Q. Cai. Human motion analysis: A review. *Computer Vision Image Understanding*, 73(3):428–440, 1999.
- [3] J. K. Aggarwal, Q. Cai, W. Liao, and B. Sabata. Nonrigid motion analysis: Articulated and elastic motion. *Computer Vision Image Understanding*, 70(2):142–156, May 1998.
- [4] J. K. Aggarwal, L. S. Davis, and W. N. Martin. Correspondence processes in dynamic scene analysis. *Proc. IEEE*, 69(6):562–572, 1981.
- [5] J. K. Aggarwal and N. Nandhakumar. On the computation of motion from sequences of images – a review. *Proc. IEEE*, 76(8):917–935, August 1988.
- [6] O. Alatas, P. Yan, and M. Shah. Spatiotemporal Regularity Flow (SPREF): Its Estimation and Applications. *IEEE Trans. Circuits and Systems for Video Technology*, (in press).

- [7] T. Amiaz, E. Lubetzky, and N. Kiryati. Coarse to over-fine optical flow estimation. *Pattern Recognition*, 40:2496–2503, 2007.
- [8] C. Anderson, P. Burt, and G. van der Wal. Change detection and tracking using pyramid transformation techniques. In *Proc. SPIE - Intelligent Robots and Computer Vision*, volume 579, pages 72–78, 1985.
- [9] M. Andersson and H. Knutsson. Transformation of local spatio-temporal structure tensor fields. In *IEEE Int. Conf. on Acoustics Speech Signal Processing (ICASSP'03)*, volume 3, pages 285–288, April 2003.
- [10] E. L. Andrade, S. Blunsden, and R. B. Fisher. Characterisation of optical flow anomalies in pedestrian traffic. In *Imaging for Crime Detection and Prevention*, pages 73–79, 2005.
- [11] L. Angrisani, M. Dapos Apuzzo, and R. Schiano Lo Moriello. The unscented transform: a powerful tool for measurement uncertainty evaluation. In *Proc. IEEE Int. Workshop on Advanced Methods for Uncertainty Estimation in Measurement*, pages 27–32, 2005.
- [12] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Processing*, 50:174–188, 2002.
- [13] A. Azarbayejani and A.P. Pentland. Recursive estimation of motion, structure, and focal length. *IEEE Trans. Pattern Analysis Machine Intelligence*, 17(6):562–575, 1995.

- [14] W. Bair, E. Zohary, and W. T. Newsome. Correlated firing in macaque visual area MT: Time scales and relationship to behavior. *J. Neuroscience*, 21(5):1676–1697, 2001.
- [15] S. Baker, T. Sim, and T. Kanade. When is the shape of a scene unique given its light-field: A fundamental theorem of 3D vision? *IEEE Trans. Pattern Analysis Machine Intelligence*, 25(1):100–109, 2003.
- [16] H. Bårman, L. Haglund, H. Knutsson, and G. H. Granlund. Estimation of velocity, acceleration and disparity in time sequences. In *IEEE Workshop on Visual Motion*, pages 44–51, Princeton, NJ, USA, October 1991. IEEE Computer Society Press.
- [17] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *Int. J. Computer Vision*, 12(1):43–77, 1994.
- [18] B. Bascle, P. Bouthemy, R. Deriche, and F. Meyer. Tracking complex primitives in an image sequence. In *Pattern Recognition Conf. A: Computer Vision & Image Processing*, volume 1, pages 426–431, 1994.
- [19] A. S. Bashi. A practitioner's short guide to Kalman filtering. Technical Report SAGES, University of New Orleans, 1998.
- [20] E. Baumgartner, W. Baumgartner, B. Borstner, J. Shawe-Taylor, and E. Valentine. *Handbook Phenomenology and Cognitive Science*. Tempus Programme of the European Union, 1996.



- [21] P. N. Belhumeur and D. J. Kriegman. What is the set of images of an object under all possible lighting conditions? In *Proc. IEEE Conf. on Computer Vision Pattern Recognition (CVPR)*, page 270, 1996.
- [22] P. N. Belhumeur and D. J. Kriegman. What is the set of images of an object under all possible illumination conditions? *Int. J. Computer Vision*, 28(3):1–16, 1998.
- [23] P. N. Belhumeur, D. J. Kriegman, and A. L. Yuille. The bas-relief ambiguity. *Int. J. Computer Vision*, 35(1):33–44, 1999.
- [24] M. Ben-Ezra, S. Peleg, and M. Werman. Robust, real-time motion analysis. In *Proc. Image Understanding Workshop (DARPA-IUW)*, pages 207–210, November 1998.
- [25] C. Benedek, L. Havasi, T. Szirányi, and Szlávik. Motion-based flexible camera registration. In *Proc. IEEE Conf. on Adv. Video and Signal Based Surveillance*, pages 439–444, 2005.
- [26] C. P. Bernard. Discrete wavelet analysis: A new framework for fast optic flow computation. In *Proc. European Conf. on Computer Vision (ECCV)*, volume 2, pages 354 – 368, 1998.
- [27] A. Bevilacqua. A novel background initialization method in visual surveillance. In *Proc. IEEE Int. Conf. on Pattern Recognition (ICPR)*, pages 614–617, 2002.

- [28] A. Bevilacqua. Effective object segmentation in a traffic monitoring application. In *Proc. IEEE Int. Conf. on Pattern Recognition (ICPR)*, pages 125–130, 2003.
- [29] A. Bevilacqua, L. Di Stefano, and A. Lanza. An effective multi-stage background generation algorithm. In *Proc. IEEE Conf. on Advanced Video and Signal Based Surveillance*, pages 388–393, 15–16 Sept 2005.
- [30] A. G. Bharatkumar, K. E. Daigle, M. G. Pandey, Q. Cai, and J. K. Aggarwal. Lower limb kinematics of human walking with the medial axis transformation. In *IEEE Workshop on Non-Rigid Motion*, pages 70–76, 1994.
- [31] J. Bigün and G. H. Granlund. Optical flow based on the inertia matrix in the frequency domain. In *Proc. SSAB Symposium on picture Processing, Lund, Sweden*, 1988.
- [32] M. J. Black, D. J. Fleet, and Y. Yacoob. Robustly estimating changes in image appearance. *Computer Vision Image Understanding*, 78:8–31, 2000.
- [33] A. Blake, R. Curwen, and A. Zisserman. A framework for spatio-temporal control in the tracking of visual contours. *Int. J. Computer Vision*, 11(2):127–145, 1993.
- [34] A. F. Bobick and J. W. Davis. Real-time recognition of activity using temporal templates. Technical Report 386, MIT Media Lab, Perceptual Computing Section, 1998.

- [35] K. H. Britten and H. W. Heuer. Spatial summation in the receptive fields of MT neurons. *J. Neuroscience*, 19(12):5074–5084, 1999.
- [36] S. C. Brofferio. An object-background image model for predictive video coding. *IEEE Trans. Communications*, 37:1391–1394, 1989.
- [37] T. J. Broida, S. Chandrashekhar, and R. Chellappa. Recursive 3-D motion estimation from a monocular image sequence. *IEEE Trans. Aerospace and Electronic Systems*, 26(4):639–656, 1990.
- [38] T. Brox, A. Bruhn, N. Papenbergh, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *Eighth European Conf. on Computer Vision (ECCV04)*, 2004.
- [39] A. Bruhn, J. Weickert, and C. Schnörr. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *Int. J. Computer Vision*, 61(3):211–231, 2005.
- [40] D. Buzan, S. Sclaroff, and G. Kollios. Extraction and clustering of motion trajectories in video. In *Proc. IEEE Int. Conf. on Pattern Recognition (ICPR)*, volume 2, pages 521–534, 2004.
- [41] A. Cavallaro and F. Ziliani. Image analysis for advanced video surveillance. In G. L. Foresti, P. Mähönen, and C. S. Regazzoni, editors, *Multimedia video-based surveillance systems: requirements, issues and solutions*, pages 57–67. Kluwer Academic Pub., 2000.

- [42] H. F. Chen, P. N. Belhumeur, and D. W. Jacobs. In search of illumination invariants. In *Proc. IEEE Conf. on Computer Vision Pattern Recognition (CVPR)*, page 1254, 2000.
- [43] K. Choo and D. J. Fleet. People tracking using hybrid Monte Carlo filtering. In *Proc. Int. Conf. on Computer Vision (ICCV)*, volume 2, pages 321–329, 2001.
- [44] J. C. Clarke, S. Carlsson, and A. Zisserman. Detecting and tracking linear features efficiently. In *Proc. British Machine Vision Association Conf. (BMVC)*, pages 415–424, 1996.
- [45] R. T. Collins. Mean-shift blob tracking through scale space. In *Proc. IEEE Conf. on Computer Vision Pattern Recognition (CVPR)*, volume 2, pages 234–240, 2003.
- [46] R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt, and L. Wixson. A system for video surveillance and monitoring. Technical Report CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University, 2000.
- [47] R. Cucchiara, M. Piccardi, and A. Prati. Detecting moving objects, ghosts, and shadows in video streams. *IEEE Trans. Pattern Analysis Machine Intelligence*, 25(10):1337–1342, October 2003.
- [48] C. Curio, J. Edelbrunner, T. Kalinke, C. Tzomakas, and W. von Seelen. Walking pedestrian recognition. *IEEE Trans. Intelligent Transportation Systems*, 1(3):292–297, 2000.

- [49] E. R. Davies. *Machine Vision: Theory Algorithms Practicalities*, 3<sup>rd</sup> edition. Morgan Kaufmann, 2005.
- [50] J. F. G. de Freitas, M. Niranjan, A. H. Gee, and A. Doucet. Sequential Monte Carlo methods to train neural network models. *Neural Computation*, 12(4):955–993, 2000.
- [51] H. Dee and D. Hogg. Detecting inexplicable behaviour. In *Proc. British Machine Vision Association Conf. (BMVC)*, volume 2, pages 477–486, Sept 2004.
- [52] J. Días, E. Ros, F. Pelayo, E. M. Ortigosa, and S. Mota. FPGA-Based real-time optical-flow system. *IEEE Trans. Circuits and Systems for Video Tech.*, 16(2):274–279, Feb 2006.
- [53] I. Dinstein. A new technique for visual motion alarm. *Pattern Recognition Letters*, 8:347–351, 1989.
- [54] W. H. Dittrich and S. E. G. Lea. Avian visual cognition: motion discrimination and recognition. Technical report, Dept. Psychology, Tufts University, 2001.
- [55] M. R. Dobie and P. H. Lewis. Object tracking in multimedia systems. In *Int. Conf. on Image Processing and its Applications*, pages 41–44, 1992.
- [56] S. L. Dockstader and A. M. Tekalp. A kinematic model for human motion and gait analysis. In *Proc. Workshop on Statistical Methods in Video (ECCV)*, pages 49–54, June 2002.

- [57] E. M. Drakakis. A review of retinomorphic devices. Technical report, Dept Bioengineering, Imperial College London, 2003.
- [58] M. S. Drew, Z. Li, and Z. Tauber. Illumination color covariant locale-based visual object retrieval. *Pattern Recognition*, 35:1687–1704, 2002.
- [59] M. S. Drew, J. Wei, and Z. Li. On illumination invariance in color object recognition. *Pattern Recognition*, 31(8):1077–1087, 1998.
- [60] M. S. Drew, J. Wei, and Z. Li. Illumination-invariant image retrieval and video segmentation. *Pattern Recognition*, 32(8):1369–1388, 1999.
- [61] A. Elgammal, R. Duraiswami, D. Harwood, and L.S. Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proc. IEEE*, 90(7):1151–1163, July 2002.
- [62] F. G. A. Faas and L. J. van Vliet. 3D-orientation space; filters and sampling. In *Proc. 13<sup>th</sup> Scandinavian Conf. on in Image Analysis (SCIA'03)*, pages 36–42, 2003.
- [63] G. Fan, V. Venkataraman, L. Tang, and J. Havlicek. A comparative study of boosted and adaptive particle filters for affine-invariant target detection and tracking. In *Proc. Computer Vision Pattern Recognition Workshop (CVPRW'06)*, pages 138–145, 2006.
- [64] M. J. Farah. *The Cognitive Neuroscience of Vision*. Blackwell, 2000.

- [65] G. Farneback. Fast and accurate motion estimation using orientation tensors and parametric motion models. In *Proc. IEEE Int. Conf. on Pattern Recognition (ICPR)*, pages 135–139, September 2000.
- [66] G. Farneback. Very high accuracy velocity estimation using orientation tensors, parametric motion, and simultaneous segmentation of the motion field. In *Proc. Int. Conf. on Computer Vision (ICCV)*, volume 1, pages 171–177. IEEE, IEEE Computer Society Press, July 2001.
- [67] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Comm. ACM*, 24(6):381–395, 1981.
- [68] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Int. Conf. on Machine Learning*, pages 148–156, 1996.
- [69] K. J. Friston and C. Büchel. Attentional modulation of effective connectivity from V2 to V5/MT in humans. *Proc. National Academy of Sciences, USA*, 97(13):7591–7596, 2000.
- [70] L. M. Fuentes and S. A. Velastin. Foreground segmentation using luminance contrast. In *Int. Conf. on Speech, Signal and Image Processing*, 2001.
- [71] L. M. Fuentes and S. A. Velastin. People tracking in surveillance applications. In *2nd Performance Evaluation on Tracking and Surveillance*, 2001.



- [72] N. Funk. A study of the Kalman filter applied to visual tracking. Technical Report CMPUT 652, University of Alberta, 2003.
- [73] P. Gabriel, J.-B. Hayet, J. Piater, and J. Verly. Object tracking using color interest points. In *Proceedings. IEEE Conf. on Advanced Video and Signal Based Surveillance*, 2005.
- [74] J. J. Gibson. *The Perception of the Visual World*. Riverside Press, Cambridge, 1950.
- [75] M. A. Giese and T. Poggio. Neural mechanisms for the recognition of biological movements. *Nature Reviews Neuroscience*, 4:179–192, March 2003.
- [76] N. Gilbert, A. Anderson, J. Backhouse, D. Birch, B. Collins, W. Dutton, J. Edwards, H. Smith, I. Forbes, W. Hall, A. Hopper, C. Jones FREng, M. Thomas, and N. McCarthy. *Dilemmas of Privacy and Surveillance: Challenges of Technological Change*. Number 1-903496-32-2. The Royal Academy of Engineering, March 2007.
- [77] B. Gloyer, H. K. Aghajan, K. Y. Siu, and T. Kailath. Video-based freeway-monitoring system using recursive vehicle tracking. *Proc. SPIE Image and Video Processing III*, 2412:173–180, March 1995.
- [78] G. H. Granlund and H. Knutsson. Orientation and velocity. In Gosta H. Granlund, editor, *Signal Processing for Computer Vision*, pages 219–258. Kluwer Academic Publishers, 1995.

- [79] S. Grossberg, E. Mingolla, and L. Viswanathan. Neural dynamics of motion integration and segmentation within and across apertures. *Vision Research*, 41:2521–2553, 2001.
- [80] I. Haritaoglu, D. Harwood, and L. Davis. W<sup>4</sup>: A real time system for detecting and tracking people in 2.5 D. In *Proc. European Conf. on Computer Vision (ECCV)*, volume 14, 1998.
- [81] C. Harris. Tracking with rigid models. In A. Blake and A. Yuille, editors, *Active Vision*, pages 59–73. The MIT Press, 1992.
- [82] D. J. Heeger and E. P. Simoncelli. Model of visual motion sensing. In L. Harris and M. Jenkin, editors, *Spatial Vision in Humans and Robots*, pages 367–392. Cambridge University Press, 1993.
- [83] J. Heikkilä and Olli Silvén. A real-time system for monitoring of cyclists and pedestrians. *Image Vision Computing*, 22(7):563–570, July 2004.
- [84] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, 1981.
- [85] T. Horprasert, D. Harwood, and L. S. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *Proc. ICCV FRAME-RATE Workshop*, pages 1–19, 1999.
- [86] Y. Z. Hsu, H. H. Nagel, and G. Rekers. New likelihood test methods for change detection in image sequences. *Computer Vision Graphics Image Processing*, 26:73–106, 1984.

- [87] C. Hue, J. Le Cadre, and P. Perez. Tracking multiple objects with particle filtering. Technical report, Inst. Nat. Recherche Informatique Automatique (INISA), Oct. 2000.
- [88] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *Proc. European Conf. on Computer Vision (ECCV)*, pages 343–356, 1996.
- [89] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *Int. J. Computer Vision*, 29(1):5–28, 1998.
- [90] M. Isard and A. Blake. Icondensation: Unifying low-level and high-level tracking in a stochastic framework. *Lecture Notes in Computer Science*, 1406:893–908, 1998.
- [91] M. Isard and J. MacCormick. BraMBLe: A Bayesian multiple-blob tracker. *Proc. Int. Conf. on Computer Vision (ICCV)*, 2:34–41, 2001.
- [92] R. Jain. Extraction of motion information from peripheral processes. *IEEE Trans. Pattern Analysis Machine Intelligence*, -(3):489–504, 1981.
- [93] R. Jain, W. N. Martin, and J. K. Aggarwal. Segmentation through the detection of changes due to motion. *Computer Graphics Image Processing*, 11:13–34, 1979.
- [94] R. C. Jain and H. H. Nagel. On the analysis of accumulative difference pictures from image sequences of real world scenes. *IEEE Trans. Pattern Analysis Machine Intelligence*, 1(2):206–213, 1979.

- [95] P. M. Jorge, A. J. Abrantes, and J. S. Marques. On-line object tracking with Bayesian networks. In *5<sup>th</sup> Int. Workshop on Image Analysis for Multimedia Interactive Systems (WIAMIS)*. Lisbon, Portugal, April 22-23 2004.
- [96] P. M. Jorge, A. J. Abrantes, and J. S. Marques. On-line tracking groups of pedestrians with Bayesian networks. In *Proc. PETS Workshop*, 2004.
- [97] S. J. Julier and J. K. Uhlmann. A general method for approximating nonlinear transformations of probability distributions. Technical report, Dept. Engineering Science, University of Oxford, 1996.
- [98] D. S. Kalivas and A. A. Sawchuk. Motion compensated enhancement of noisy image sequences. In *Int. Conf. on Acoustics Speech Signal Processing*, volume 4, pages 2121–2124, 1990.
- [99] K. P. Karmann and A. von Brandt. Moving object recognition using an adaptive background memory. In V. Cappellini, editor, *Time-Varying Image Processing and Moving Object Recognition 2*, pages 289–296. Elsevier, 1990.
- [100] H. Knutsson, L. Haglund, H. Bårman, and G.H. Granlund. A framework for anisotropic adaptive filtering and analysis of image sequences and volumes. In *IEEE Trans. Acoustics Speech Signal Processing*, volume 3, pages 469–472. IEEE, March 1992.
- [101] T. Kobayashi and N. Otsu. A three-way auto-correlation based approach to human identification by gait. In *Visual Surveillance*, 2006.

- [102] A. Kojima, T. Tamura, and K. Fukunaga. Natural language description of human activities from video images based on concept hierarchy of actions. *Int. J. Computer Vision*, 50(2):171–184, 2002.
- [103] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russell. Towards robust automatic traffic scene analysis in real-time. In *Proc. IEEE Int. Conf. on Pattern Recognition (ICPR)*, pages 126–131, Oct 1994. Jerusalem, Israel.
- [104] C.-S. Lee. Human identification using silhouette gait data. In *First Int. Conf. on Machine Learning (iCML'03)*, pages 21–25, 2003.
- [105] H. J. Lee and C. C. Hsieh. Extraction and matching of multiple moving objects via frame differencing. *J. Information Science and Engineering*, 6:409–424, Sept 1988.
- [106] J. A. Leese, C. S. Novak, and V. R. Taylor. The determination of cloud pattern motion from geosynchronous satellite image data. *Pattern Recognition*, 2:279–292, 1970.
- [107] B. Lei and L.-Q. Xu. From pixels to objects and trajectories: a generic real-time outdoor video surveillance system. In *IEE Int. Symp. Imaging for Crime Detection and Prevention (ICDP'05)*, pages 117–122, 2005.
- [108] Z. Li. Optimal sensory encoding. In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks: Second Edition*, pages 815–819. MIT Press, 2002.

- [109] B. P. L. Lo and S. A. Velastin. Automatic congestion detection system for underground platforms. In *Proc. Int. Symp. Intelligent Multimedia, Video & Speech Processing (ISIMP)*, pages 158–161, May 2001.
- [110] W. Long and Y. H. Yang. Stationary background generation: An alternative to the difference of two images. *Pattern Recognition*, 23(12):1351–1359, 1990.
- [111] J. Lou, T. Tan, and W. Hu. Visual vehicle tracking algorithm. *Electronics Letters*, 38(17):1024–1025, August 2002.
- [112] J. Lou, H. Yang, W. Hu, and T. Tan. An illumination invariant change detection algorithm. In *5<sup>th</sup> Asian Conf. on Computer Vision (ACCV)*, 2002.
- [113] T. D. Lucas and T. Kanade. An interactive image registration technique with an application to stereo vision. *Proc. DARPA Image Understanding Workshop*, 1:121–130, 1984.
- [114] D. Makris. Visual learning in surveillance systems. Transfer report, Dept. Electrical, Electronic and Information Engineering, City University, London, 2001.
- [115] D. Makris and T. Ellis. Path detection in video surveillance. *Image Vision Computing*, 20(12):895–903, October 2002.
- [116] J. Malik, J. Weber, T. Luong, and D. Koller. Smart cars and smart roads. In *Proc. British Machine Vision Association Conf. (BMVC)*, pages 367–382. Birmingham, UK, 2005.

- [117] M. Markou and S. Singh. Novelty detection: a review – part 1: statistical approaches. *Signal Processing*, 83:2481–2497, 2003.
- [118] M. Markou and S. Singh. Novelty detection: a review – part 2: neural network based approaches. *Signal Processing*, 83:2499–2521, 2003.
- [119] W. N. Martin and J. K. Aggarwal. Survey: Dynamic scene analysis. *Computer Graphics Image Processing*, 7:356–374, 1978.
- [120] N. J. B. McFarlane and C. P. Schofield. Segmentation and tracking of piglets in images. *Machine Vision Applications*, 8:187–193, 1995.
- [121] A. M. McIvor. Background subtraction techniques. Technical report, Reveal Ltd, 2000.
- [122] D. Minnen, I. Essa, and T. Starner. Expectation grammars: leveraging high-level expectations for activity recognition. In *Proc. IEEE Conf. on Computer Vision Pattern Recognition (CVPR)*, volume 2, pages 626–632, June 2003.
- [123] T. B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *Computer Vision Image Understanding*, 81:231–268, 2001.
- [124] H. Momiji. Review of mathematical models of retinal functions motivated by physiology. Technical report, Imperial College London, 2003.
- [125] H. Momiji, A. A. Bharath, M. W. Hankins, and C. Kennard. Modelling retinal functions: Fovea. Technical report, Imperial College London, 2003.



- [126] A. Monnet, A. Mittal, N. Paragios, and V. Ramesh. Background modeling and subtraction of dynamic scenes. In *Proc. Int. Conf. on Computer Vision (ICCV)*, volume 2, page 1305, 2003.
- [127] M. J. Nadenau, S. Winkler, D. Alleysson, and M. Kunt. Human vision models for perceptually optimized image processing—a review. *Submitted to Proc. IEEE*, --, Sept 2000.
- [128] A. Naftel and S. Khalid. Video sequence indexing through recovery of object-based motion trajectories. In *Proc. Irish Machine Vision and Image Processing Conf. (IMVIP)*, pages 232–240, 2004.
- [129] H. H. Nagel. Image sequences – ten (octal) years – from phenomenology towards a theoretical foundation. *Int. J. Pattern Recog. Artif. Intell.*, 2:459–486, 1988.
- [130] H. Nait-Charif and S. J. McKenna. Tracking the activity of participants in a meeting. *Machine Vision Applications*, 17(2):1432–1769, 2006.
- [131] P. Neri, M. C. Morrone, and D. C. Burr. Seeing biological motion. *Nature*, 395:894–896, 29 October 1998.
- [132] W. Niu, J. Long, D. Han, and Y.-F. Wang. Real-time multi-person tracking in video surveillance. In *Proc. Pacific Rim Multimedia Conf.*, volume 2, pages 1144–1148, 2003.
- [133] W. Niu, J. Long, D. Han, and Y.-F. Wang. Human activity detection and recognition for video surveillance. In *Multimedia and Expo*, volume 1, pages 719–722, June 2004.

- [134] M. S. Nixon, T. N. Tan, and R. Chellappa. *Human Identification Based on Gait*. Springer, 2006.
- [135] S. A. Niyogi and E. H. Adelson. Analyzing and recognizing walking figures in xyt. Technical report, M.I.T. Media Lab Vision and Modeling, 1994.
- [136] K. Nummiaro, E. B. Koller-Meier, and L. Van Gool. A color-based particle filter. In *1<sup>st</sup> Int. Workshop on Generative-Model-Based Vision GMBV'02, in conjunction with ECCV'02*, pages 53–60, 2002.
- [137] N. Ohta. A statistical approach to background subtraction for surveillance systems. In *Proc. Int. Conf. on Computer Vision (ICCV)*, volume 2, pages 481–486, June 2001.
- [138] K. Okuma, A. Taleghani, N. de Freitas, J. Little, and D. Lowe. A boosted particle filter: Multitarget detection and tracking. In *Proc. European Conf. on Computer Vision (ECCV)*, pages 28–39, 2004.
- [139] J. Owens, A. Hunter, and E. Fletcher. Novelty detection in video surveillance using hierarchical neural networks. In *Int.l Conf. Artificial Neural Networks (ICANN)*, pages 1249–1254, 2002.
- [140] S.-L. Peng. Temporal slice analysis of image sequences. In *Proc. IEEE Conf. on Computer Vision Pattern Recognition (CVPR)*, pages 283–288, 1991.

- [141] S.-L. Peng and G. Medioni. Interpretation of image sequences by spatio-temporal analysis. In *Proc. Workshop on Visual Motion*, pages 344–351, 1989.
- [142] S.-L. Peng and G.G. Medioni. Spatio-temporal analysis of an image sequence with occlusion. In *Proc. Image Understanding Workshop (DARPA-IUW)*, pages 433–442, 1988. (Referred to by McFarlane and Schofield 1995).
- [143] M. Pic, L. Berthouze, and T. Kurita. Adaptive background estimation: Computing a pixel-wise learning rate from local confidence and global correlation values. *IEICE Trans. Information and Systems*, E87-D(1):50–57, 2004.
- [144] M. Piccardi. Background subtraction techniques: a review. In *Int. Conf. on Systems, Man and Cybernetics*, volume 4, pages 3099–3104, Oct 2004.
- [145] R. Pless. Spatio-temporal background models for outdoor surveillance. *Applied Signal Processing*, 2005:2281–2291, 2005.
- [146] P. Wayne Power and J. A. Schoonees. Understanding background mixture models for foreground segmentation. In *Proc. Image and Vision Computing New Zealand*, pages 267–271, 2002.
- [147] A. Prati, R. Cucchiara, I. Mikic, and M. M. Trivedi. Analysis and detection of shadows in video streams: A comparative evaluation. *Proc. IEEE Conf. on Computer Vision Pattern Recognition (CVPR)*, 02:571, 2001.

- [148] A. Prati, I. Mikic, M. M. Trivedi, and R. Cucchiara. Detecting moving shadows: Algorithms and evaluation. *IEEE Trans. Pattern Analysis Machine Intelligence*, 25(7):918–923, 2003.
- [149] N. J. Priebe and S. G. Lisberger. Estimating target speed from the population response in visual area MT. *J. Neuroscience*, 24(8):1907–1916, 2004.
- [150] J. W. Roach and J. K. Aggarwal. Computer tracking of objects moving in space. *IEEE Trans. Pattern Analysis Machine Intelligence*, PAMI-2(2):127–135, 1979.
- [151] K. Roberts and M. Nashman. Real-time model-based tracking combining spatial and temporal features. *J. Intell. Robotic Syst.*, 5:25–38, 1992.
- [152] P. L. Rosin. Thresholding for change detection. Technical Report ISTR-97-01, Brunel University, 1997.
- [153] Y. Rui and Y. Chen. Better proposal distributions: object tracking using unscented particle filter. In *Proc. IEEE Conf. on Computer Vision Pattern Recognition (CVPR)*, pages 786–793, 2001.
- [154] K. Sato and J.K. Aggarwal. Temporal spatio-velocity transform and its applications to tracking and interaction. *Computer Vision Image Understanding*, 96:100–128, 2004.

- [155] R. Sekuler, S. N. J. Watamaniuk, and R. Blake. Perception of visual motion. In H. Pasher, editor, *Stevens Handbook of Experimental Psychology*, pages 1–56. J. Wiley, 2001.
- [156] A. Selinger and L. Wixson. Classifying moving objects as rigid or non-rigid. In *Proc. Image Understanding Workshop (DARPA-IUW)*, 1998.
- [157] F. Sengpiel. Motion perception is learned, not innate. *Nature Neuroscience*, 9(5):591–592, 2006.
- [158] M. N. Shadlen and W. T. Newsome. Motion perception: seeing and deciding. *Proc. National Academy of Sciences, USA*, 93(93):628–633, 1996.
- [159] E. Shechtman and E. Irani. Space-time behavior based correlation. In *Proc. IEEE Conf. on Computer Vision Pattern Recognition (CVPR)*, volume 1, pages 405–412, 2005.
- [160] E. P. Simoncelli and D. J. Heeger. A model of neuronal responses in visual area MT. *Vision Research*, 38(5):743–761, 1998.
- [161] K. Skifstad and R. Jain. Illumination independent change detection for real world image sequences. *Computer Vision Graphics Image Processing*, 46:387–399, 1989.
- [162] E. E. Snyder, S. A. Rajala, and G. Hirzinger. Image modeling, the continuity assumption and tracking. In *Proc. IEEE Int. Conf. on Pattern Recognition (ICPR)*, pages 1111–1114, 1981.

- [163] B.-S. Sohn, C. Bajaj, and V. Siddavanahalli. Volumetric video compression for interactive playback. *Computer Vision Image Understanding*, 96:435–452, 2004.
- [164] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Proc. IEEE Conf. on Computer Vision Pattern Recognition (CVPR)*, volume 2, pages 246–252, 1999.
- [165] C. Stauffer and W. E. L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Trans. Pattern Analysis Machine Intelligence*, 22(8):747–757, 2000.
- [166] B. Stenger, P. R. S. Mendonça, and R. Cipolla. Model-based hand tracker using an unscented Kalman filter. In *Proc. British Machine Vision Association Conf. (BMVC)*, volume 1, pages 63–72, 2001.
- [167] M. Sugrue and E. R. Davies. Tracking in CCTV video using human visual system inspired algorithm. In *Proc. IEE International Conference on Visual Information Engineering (VIE 2005)*, University of Glasgow, Glasgow, pages 417–423, 4–6 April 2005.
- [168] T. Syeda-Mahmood, D. Ponceleon, and J. Yang. Validating cardiac echo diagnosis through video similarity. In *Proc. of ACM Multimedia*, 2005.
- [169] I. M. Thornton, R. A. Rensink, and M. Shiffrar. Active versus passive processing of biological motion. *Perception*, 31:837–853, 2002.

- [170] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: Principles and practice of background maintenance. In *Proc. Int. Conf. on Computer Vision (ICCV)*, volume 1, pages 255–261, 1999.
- [171] T. Troscianko, A. Holmes, J. Stillman, M. Mirmehdi, D. Wright, and A. Wilson. What happens next? the predictability of natural behaviour viewed through CCTV camera. *Perception*, 33(1):87–101, 2004.
- [172] J. K. Tsotsos, D. J. Fleet, and A. D. Jepson. Towards a theory of motion understanding in man and machine. In W. N. Martin and J. K. Aggarwal, editors, *Motion Understanding: Robot and Human Vision*, pages 353–417. Kluwer, 1988.
- [173] T. Tsukiyama and Y. Shirai. Detection of the movements of persons from a sparse sequence of TV images. *Pattern Recognition*, 18:207–213, 1985.
- [174] S. Ullman. The interpretation of structure from motion. *Proc. R. Soc.*, B203:405–426, 1979.
- [175] S. Vaccari. Sistemi di videosorveglianza. Technical Report 2148 061247, Università Degli Studi Di Bologna, Facoltà Di Ingegneria, 2003.
- [176] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan. The unscented particle filter. *Adv. Neural Information Processing Systems*, 13:584–590, 2001.



- [177] I. R. Vega and S. Sarkar. Statistical motion model based on the change of feature relationships: Human gait-based recognition. *IEEE Trans. Pattern Analysis Machine Intelligence*, 25(10):1323–1328, 2003.
- [178] E. A. Wan and R. van der Merwe. The unscented Kalman filter. In S. Haykin, editor, *Kalman Filtering and Neural Networks*, pages 221–280. Wiley, 2001.
- [179] M. Wan and J.-Y. Herve. Adaptive, region-based, layered background model for target tracking. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 1, pages 803–807, 2006.
- [180] Y. Wang, J. F. Doherty, and R. Van Dyck. Moving object tracking in video. In *29<sup>th</sup> Applied Imagery Pattern Recognition Workshop (AIPR'00)*, page 95, 2000.
- [181] G. Welch and G. Bishop. An introduction to the Kalman filter. Technical Report TR 95-041, University of North Carolina, 2003.
- [182] A. H. Wertheim. Motion perception during self-motion: The direct versus inferential controversy revisited. *Behavioural and Brain Sciences*, 17(2):293–355, 1994.
- [183] Y. Wexler, E. Shechtman, and M. Irani. Space-time video completion. In *Proc. IEEE Conf. on Computer Vision Pattern Recognition (CVPR)*, volume 1, pages 120–127, July 2004.
- [184] R. Williams. *The Animator's Survival Kit*. Faber and Faber, 2001.

- [185] J. Wills, S. Agarwal, and S. Belongie. A feature-based approach for dense segmentation and estimation of large disparity motion. *Int. J. Computer Vision*, 68(2):125–143, June 2006.
- [186] C. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Trans. Pattern Analysis Machine Intelligence*, 19(7):780–785, 1997.
- [187] L.-Q. Xu. Robust detection and tracking of multiple objects in cluttered scenes. In *Spatiotemporal Image Processing*, pages 1–4. One Day BMVA symposium at the Royal Statistical Society, 24 March 2004.
- [188] J. Yang and A. Waibel. Tracking human faces in real-time. Technical Report CMU-CS-95-210, School of Computer Science, Carnegie Mellon University, Nov. 1995.
- [189] Y. H. Yang and M.D. Levine. The background primal sketch: an approach for tracking moving objects. *Machine Vision Appl*, 5:17–34, 1992.
- [190] A. Yilmaz and M. Shah. Matching actions in presence of camera motion. *Computer Vision and Image Understanding*, 104:221–231, 2006.
- [191] J.-H. Yoo and M. S. Nixon. On laboratory gait analysis via computer vision. In *AISB '03 Symposium on Biologically-Inspired Machine Vision, Theory and Application*, pages 109–113, 2003.

- [192] R. A. Young, R. M. Lesperance, and W. W. Meyer. The Gaussian derivative model for spatial vision. I. Retinal mechanisms. *Spatial Vision*, 14:261–319, 2001.
- [193] R. A. Young, R. M. Lesperance, and W. W. Meyer. The Gaussian derivative model for spatial vision. II. Cortical data. *Spatial Vision*, 14:321–389, 2001.
- [194] W. Yu, G. Sommer, and K. Daniilidis. 3D-orientation signatures with conic kernel filtering for multiple motion analysis. In *Proc. IEEE Conf. on Computer Vision Pattern Recognition (CVPR)*, volume 1, pages 299–311, 2001.
- [195] Q. Zhou and J. K. Aggarwal. Tracking and classifying moving objects from video. In *Proc. Performance Evaluation of Tracking and Surveillance (CVPR PETS)*, pages 46–54, December 2001.
- [196] S. Zhou, R. Chellappa, and B. Moghaddam. Visual tracking and recognition using appearance-adaptive models in particle filters. *IEEE Trans. Image Processing*, 13:1491–1506, 2004.