

Explaining visible behaviour

by

Hannah-Mary Dee

**Submitted in accordance with the requirements
for the degree of Doctor of Philosophy.**



**The University of Leeds
School of Computing**

September 2005

The candidate confirms that the work submitted is her own and that the appropriate credit has been given where reference has been made to the work of others. This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

Abstract

This thesis presents a novel approach to the problem of behaviour modelling within computer vision. This technique is not based upon statistical measures of typicality, but upon building an understanding of the way people navigate towards a goal. Representing movement through the scene in terms of the known goals and obstacles and interpreting people's behaviour as representative of underlying intentions enables behaviour to be *explained* in terms of these previously defined goals.

A family of related algorithms for performing this goal-directed analysis of behaviour are presented and evaluated, alongside a number of metrics for measuring how well the computed explanation matches the observed behaviour. These measurements can be interpreted as measurements of goal-directedness or intentionality.

The system is evaluated using a novel methodology which involves comparing the algorithmic output with the performance of humans engaged in a visual surveillance task. An application of this technique is demonstrated within the visual surveillance domain, providing classification of behaviour patterns as *explicable* or *inexplicable*.

The advantages of such an approach are multiple: it handles the presence of movable goals (for example, parked cars) with ease, and trajectories which have never before been presented to the system can be classified as explicable. The output of the system (for example "Agent n is heading towards goal m " with an associated score indicating how good this explanation is) are easily interpreted. The systems described in this thesis could also in principle be extended to handle richer varieties of scene, moving obstacles, and more complicated systems of goals.

Acknowledgements

I would like to thank my supervisor, David Hogg, for support and guidance over the course of this PhD. I'd also like to thank my colleagues in the Leeds Vision group, particularly Chris Needham and Derek Magee. The vision group has provided me with countless stimulating conversations, invaluable technical advice, several useful pieces of c++ and an object tracker.

I have a huge debt of gratitude to the many volunteers who watched videos of behaviour in car-parks and foyers in order to provide the ground truth used in Chapters 7 and 8. They have been paid in chocolate, but that was insufficient recompense given the extreme dullness of the task.

Thanks are due to Neil Johnson of Irisys and Ruth Conroy-Dalton of UCL who have kindly given permission for me to reproduce some diagrams within Chapter 2.

I'd also like to thank all of the lodgers who have contributed to my mortgage at some point during the last four years (Tom Wiltshire, Mark Addy, Wills Towle, Mike Sandell, Thomas Chalk, Roger Boyle, Paolo Santos, Ben Gwynne and Haley, Fausto Spoto). I'd like to thank my parents for their assistance, too.

Special thanks are due to my partner, Roger Boyle.

Declarations

Some parts of the work presented in this thesis have been published in the following articles:

- Dee, H. M. and Hogg, D. C.**, “Is it interesting? Comparing human and machine judgements on the PETS dataset”, *Proceedings of the Performance Evaluation of Tracking and Surveillance (PETS) Workshop, European Conference on Computer Vision*, Prague, Czech Republic (2004)
- Dee, H. M. and Hogg, D. C.**, “Detecting inexplicable behaviour”, *Proceedings of the British Machine Vision Conference*, Kingston, UK (2004)
- Dee, H. M. and Hogg, D. C.**, “On the feasibility of using a cognitive model to filter surveillance data”, *Proceedings of the IEEE International Conference on Advanced Video and Signal-Based Surveillance*, Como, Italy (2005)

Contents

1	Introduction	1
1.1	The problem domain	1
1.1.1	Visual surveillance	1
1.2	Modelling behaviour in intentional terms	2
1.2.1	The Intentional Stance	2
1.2.2	Applying intentionality	3
1.3	Thesis overview	3
2	Background and previous work	5
2.1	Surveillance in the real world	5
2.1.1	Deciding which scenes to watch	6
2.1.2	Evaluating the effectiveness of CCTV	8
2.1.3	Concluding remarks upon the practical and social aspects of real-world CCTV installations	9
2.2	Computer Vision	9
2.2.1	Modelling scene geography	10
2.2.2	Behaviour modelling within computer vision	14
2.2.2.1	Behaviour modelling with Hidden Markov Models	14
2.2.2.2	Behaviour modelling with Bayesian networks	17
2.2.2.3	Other statistical and machine learning approaches to behaviour modelling	19
2.2.2.4	Ad-hoc approaches to behaviour modelling for surveillance	22
2.2.3	Concluding remarks on computer vision	23
2.3	Cognitive and philosophical considerations	23
2.4	Navigational strategies	25
2.4.1	Human path planning and spatial cognition	25
2.4.1.1	Cognitive maps	26

2.4.1.2	Path planning and distance perception	26
2.4.1.3	Simulations	29
2.4.2	Path planning in robotics and other quests for the ideal path . . .	30
2.5	Concluding remarks	33
3	Tracking and scene modelling	35
3.1	Introduction	35
3.1.1	Experimental data	36
3.2	Tracking the moving objects	37
3.3	Obstacles	39
3.4	Exits	41
3.4.1	Finding and representing the exits with a mixture of Gaussians . .	44
3.4.2	Some thoughts upon exits and goal-directedness	47
3.5	Object permanence	48
4	Building an agent-centered representation of the scene	50
4.1	Sub-goals	51
4.1.1	Determination of candidate sub-goals within a bitmapped representation	53
4.1.2	Determination of candidate sub-goals within a polygonal representation	54
4.2	Concluding remarks	55
5	Navigational strategies and path comparison	58
5.1	Navigational strategies: Shortest path vs. Simplest path	59
5.1.1	Path generation	60
5.2	Finding the closest path to the trajectory	63
5.2.1	Image to ground plane transformation	63
5.2.2	Hausdorff measures	63
5.3	Measuring how closely the agent is following the path	67
5.4	Concluding remarks	76
6	Another way of measuring goal-directedness	78
6.1	Goal classification	79
6.2	Using goal classification to explain behaviour	80
6.3	Concluding remarks	83

7	The measurement of interesting behaviour	85
7.1	The evaluation of event detection	86
7.2	Psychological evaluation	87
7.3	Correlation statistics	89
7.4	Between-human correlations	91
7.4.1	Ranking of the car-park dataset	91
7.4.2	Ranking of the PETS2004 dataset	92
7.4.3	Consideration of high-variance cases	93
7.4.4	Concluding remarks upon the human ranks	96
7.5	Comparing human rankings with computer-generated scores: Shortest path and simplest path	97
7.6	Comparing human rankings with computer-generated scores: The online algorithm	98
7.6.1	Bitmapped or polygonal representation?	99
7.6.2	Lowest-cost goal or closest goal to trajectory end?	100
7.6.3	The effect of limiting depth of search	103
7.7	An evaluation of the exit model	106
7.8	Consideration of high-variance cases	107
7.9	Concluding remarks	108
8	An example surveillance application	110
8.1	Choosing a threshold	111
8.2	Concluding remarks	114
9	Conclusions	115
9.1	Summary	115
9.2	Discussion	116
9.3	Future work and possible extensions	117
9.3.1	Modelling the scene: learning and extending	117
9.3.2	Working within a larger scene area	121
9.3.3	Other possible extensions	122
A	Correlation results with individual humans	124
A.1	The car-park dataset	124
A.2	The PETS2004 dataset	127
	Bibliography	131

List of Figures

2.1	A CCTV control room	7
2.2	A Hidden Markov Model (HMM)	15
2.3	A Coupled Hidden Markov Model	16
2.4	Pedestrian scene and behaviour vector (from Johnson 1998)	20
2.5	Ranking of strategies used in path planning (from Golledge 1995)	27
2.6	Asymmetry in path planning	28
3.1	The car-park scene	36
3.2	The PETS2004 scene	37
3.3	Obstacle models for the car-park dataset	40
3.4	Obstacle models for the PETS2004 dataset	41
3.5	Trajectory start and end points	43
3.6	Hand labelled occlusion classification	43
3.7	Cootes and Taylor’s altered EM algorithm to fit a mixture of m Gaussians to n samples x_i	44
3.8	Mixture models trained upon raw points and selected points	45
3.9	Mixture model trained upon selected data points	46
3.10	Mixture model of exits illustrated alongside obstacles	47
3.11	Hand-crafted exit model and scene: car-park dataset	48
3.12	Hand-crafted exit model and scene: PETS2004 dataset	48
4.1	How sub-goals change over time	52
4.2	An illustration of the sub-goal algorithm in action	54
4.3	The determination of candidate sub-goals with a polygonal representation	55
4.4	Some example agent-centered maps	56
5.1	Simplest path algorithm	60
5.2	Shortest path algorithm	61
5.3	All potential shortest and simplest paths for example agents	62

5.4	An illustration of a problematic trajectory	65
5.5	The selection of matched points in Hausdorff and Monotonic Hausdorff calculations	66
5.6	Illustrations of all potential simplest and shortest paths with closest path highlighted	66
5.7	Examples of goal-directed behaviour where the monotonic Hausdorff distance is high	67
5.8	Agent 44, a trajectory identified as having 4 segments	68
5.9	Agent 22, a trajectory identified as having 2 segments	69
5.10	Agent 36, a trajectory identified as having 3 segments	69
5.11	Trajectory turning points located using angular disparity alone	71
5.12	The effect of varying the weighting factor λ	72
5.13	Sample explanations: Same path, low scoring (PETS2004)	75
5.14	Sample explanations: Same path, high scoring (car-park)	75
5.15	Sample explanations: Different path (PETS2004)	76
5.16	Sample explanations: Different path, problematic (car-park)	77
5.17	Sample explanations: inexplicable (car-park)	77
6.1	An example trajectory: frame numbers inside circles	81
6.2	State transition diagram indicating the cost of each transition	83
7.1	Examples where closest goal to finish and lowest cost goal are very near	101
7.2	Closest or lowest-cost? Overshooting a corner	101
7.3	Closest or lowest-cost? Multiple low-cost goals	102
7.4	Closest or lowest-cost? Problematic trajectories	102
7.5	The number of goals classified as being at each level of sub-goal analysis during uncapped search	104
7.6	Example output from depth capped search showing that in many cases, higher level sub-goals do not add much to the analysis.	105
7.7	Graph showing Spearman's Rho for evenly spaced exits, learned exits and hand-crafted exits	106
7.8	Graph showing Kendall's Tau for evenly spaced exits, learned exits and hand-crafted exits	107
7.9	Trajectories with a high level of disagreement between human and machine ranks	108
7.10	Correlation with the mean human result: overview of the car-park dataset	109
7.11	Correlation with the mean human result: overview of the PETS2004 dataset	109

8.1	ROC curves for various values of T_H . Thresholds should be selected which maximise the true positive rate.	112
8.2	The effect of thresholding by trajectory T_C	113
8.3	The effect of thresholding by frame	114
9.1	Tracks and scene from a pedestrian area: the location of the obstacles could be inferred from the tracks alone	118
9.2	An example of an agent where path difference is due to an artifact of the obstacle model	119
9.3	Locations of sub-goals determined by path partitioning, shown next to the car-park scene.	120

List of Tables

5.1	Cost function summary statistics (Equation 5.6)	74
5.2	Angular disparity summary statistics (Equation 5.4)	74
5.3	Angular disparity ignoring small angles summary statistics (Equation 5.7)	74
6.1	Patterns of goal activity over time, corresponding to Figure 6.1	82
7.1	Between-human Spearman’s correlation matrix, car-park dataset	92
7.2	Between-human Kendall’s correlation matrix, car-park dataset	92
7.3	Overview of the PETS2004 dataset	94
7.4	Between-human Spearman’s correlation matrix, PETS2004 dataset	95
7.5	Between-human Kendall’s correlation matrix, PETS2004 dataset	95
7.6	Correlations between use of shortest and simplest path metrics and the human rankers, car-park dataset	98
7.7	Correlations between use of shortest and simplest path metrics and the human rankers, PETS2004 dataset	98
7.8	Correlation statistics for the car park dataset comparing polygonal and bitmapped implementations	99
7.9	Correlation statistics comparing closest and lowest cost goal	103
7.10	Human mean correlations with depth limited search: both datasets	103
A.1	Individual correlations: Carpark dataset, shortest vs. simplest path, R_s	125
A.2	Individual correlations: Carpark dataset, shortest vs. simplest path, T_K	125
A.3	Correlation statistics for the car park dataset comparing polygonal and bitmapped implementations	125
A.4	Correlation statistics for the car-park dataset comparing closest and lowest cost goal	126
A.5	Correlation statistics for the car-park dataset comparing depth limited search: R_s	126

A.6	Correlation statistics for the car-park dataset comparing depth limited search: T_K	126
A.7	A comparison of regularly-spaced, learned and hand-crafted exit models within the carpark dataset: Correlations with the human mean	127
A.8	Correlations between shortest and simplest path metrics and the human rankers, R_s , PETS2004 dataset	128
A.9	Correlations between shortest and simplest path metrics and the human rankers, R_s , PETS2004 dataset	128
A.10	Correlation statistics for the PETS2004 dataset comparing polygonal and bitmapped implementations	129
A.11	Correlation statistics for the PETS2004 dataset comparing closest and lowest cost goal	129
A.12	Correlation statistics for the PETS2004 dataset comparing depth limited search: R_s	130
A.13	Correlation statistics for the PETS2004 dataset comparing depth limited search: T_K	130

Chapter 1

Introduction

1.1 The problem domain

This thesis investigates the modelling of human behaviour from video sequences. The motivations behind performing such behaviour modelling within computer vision are manifold. Modelling the dynamics of a system and predicting where that system is going to be in the next time-step is a technique at the heart of most tracking applications. Modelling human and animal behaviour over longer intervals of time has applications in areas as diverse as livestock monitoring, virtual reality, and surveillance. By developing accurate models of the way in which people move through a scene, the way in which they interact with their environment and so on, their current behaviour can be explained and classified, and their future behaviour predicted.

1.1.1 Visual surveillance

One particular domain in which such techniques have shown promise is that of visual surveillance. The number of surveillance cameras in the United Kingdom is difficult to estimate, but has been put as high as four million [84]. Needless to say, not all of these are watched all the time (with one for every 15 people in the country, the surveillance industry would have to be vast). Surveillance cameras are generally used for *reactive* policing, that is, the gathering of evidence after a crime has occurred. The task of surveillance is also a fundamentally boring job: on the vast majority of these cameras, nothing of interest

happens at all. Video surveillance is therefore a good candidate for task automation.

The automated monitoring of video cameras for specific events (motion in areas where there should not be any, for example) is a well understood problem. Modelling the sorts of behaviour patterns one typically finds in pedestrian scenes and then detecting outliers to the model is a more sophisticated way of approaching the problem. It is this second approach that will be investigated in depth in this thesis.

1.2 Modelling behaviour in intentional terms

The work described here is specifically concerned with the construction and evaluation of models of *intentional* behaviour. Intentional behaviour is behaviour directed towards a goal, and in modelling this type of behaviour this thesis assumes that it is useful to model the intentions behind the behaviour. When engaging in a surveillance task, and watching the behaviour of others, particular questions are asked: *What are they up to?* or *Where are they going?*... Indeed, what is sought is an *explanation* for the agent's behaviour. These explanations are formulated in terms of the goals of the agent – they are generally intentional explanations – and it is that which motivates the work in this thesis.

1.2.1 The Intentional Stance

This work has been inspired in part by the work of the philosopher Daniel Dennett. Dennett has long been an advocate of what he calls “The Intentional Stance” - see, for example, [35, 36]. He divides the world into three varieties of system - physical, designed, and intentional - and three corresponding ways of thinking about systems. If we adopt the *Physical Stance* towards an object we take into account its physical characteristics whilst trying to explain its behaviour. As an example, consider a human drinking water from a glass. It is possible to describe such an event in purely physical terms - the chemical changes in the person's brain lead to chemical reactions in muscles, which move the arm. . . Adopting the *Design Stance* involves thinking about the object as having been designed to perform a task. Taking this perspective on the previous example, we can think of the arm as a system “designed” to move and lift objects, and the human as a system designed to need water, and so on. In a sense, adopting this stance involves modelling the system from an engineering perspective. Finally, adopting the *Intentional Stance* involves treating the object as an intentional agent and reasoning about its past and future behaviour on the grounds of its beliefs and desires. Returning to our drinker, the intentional stance allows us to talk of thirst, and motivation, and actions which will slake that

thirst. Each stance enables explanation of the object's behaviour on different terms, and each stance provides a different answer to the question "why?".

In behaviour modelling within computer vision, the currently dominant models involve analysing motion in a statistical manner and discerning patterns of activity over time. The humans (or animals, or vehicles...) under investigation are essentially treated as objects to be observed, measured and predicted based upon their visible patterns of motion alone. As the models are statistical, they cannot say anything more than that the objects typically move in a particular fashion. Such an approach can be characterised as adopting Dennett's *Physical Stance* towards the *objects* within the scene.

1.2.2 Applying intentionality

This thesis will investigate ways in which the intentions of agents can be inferred from their visible behaviour. A practical application of an intentional model of behaviour will be developed in the surveillance domain. This application will be based around making a simple model of those goals which are typical for a scene, a model of how people navigate towards a particular goal, and determining how consistent a given agent's behaviour is with motion towards (one of) these possible goals. In doing this, this work is the first within computer vision to propose stepping back from the visual information and attempting to draw conclusions about human visible behaviour from the realms of intentions and psychology rather than from the realm of statistics.

1.3 Thesis overview

In this introduction, a general introduction to the problem under consideration and an overview of the proposed approach have been presented. Chapter 2 presents a more detailed analysis of the problem and a review of related work from within computer vision, and from other related disciplines. The rest of this thesis is organised as follows:

Chapter 3 describes data collection and scene modelling. The tracking process is outlined. Approaches to modelling scene geography are described, including hand-crafting models and learning the location of scene elements (exits) using an approach based upon Gaussian mixtures.

Chapter 4 describes the construction of an agent-centered map of the scene from the information gathered in Chapter 3. The concept of a *sub-goal* is introduced.

Chapter 5 investigates the way in which people navigate through a scene. Two alternative navigational hypotheses are stated (*Shortest path* and *Simplest path*). The agent-centered representation from Chapter 4 is used as the basis for generating all possible paths through the scene from the agent's current position, and then various distance metrics are described for comparing the trajectory of the agent with one of the ideal paths. The distance metrics involve distance in space (*Hausdorff* distance, and a modification called *monotonic Hausdorff* distance) and other metrics based upon angular disparity and relative proportions of path segments. These metrics, it is argued, enable measurement of the *intentionality* or *goal-directedness* of the agents.

Chapter 6 presents a different way of assessing the goal-directness or otherwise of a trajectory. The algorithm presented in this chapter (called the *online* algorithm) uses a finite state model to determine how good each of the known goals is as an explanation for the agent's trajectory.

Chapter 7 presents a novel approach to the evaluation of surveillance systems. This approach involves comparing the output of the intentionality-based algorithms described in earlier chapters with the performance of humans undertaking a similar task: deciding how interesting the behaviour of each agent is. Correlations are provided comparing human performance with that of the shortest and simplest path metrics from Chapter 5 and the online metric from Chapter 6.

Chapter 8 describes a specific surveillance application, applying the results of the online algorithm described in Chapter 6 to the problem of filtering surveillance data.

Finally, conclusions and suggestions for future work are presented in Chapter 9.

Chapter 2

Background and previous work

This thesis proposes using a cognitive computational model of human intentional behaviour to inform a computer vision system with application to surveillance and hence this chapter must provide an adequate grounding in all of these areas.

The problem of visual surveillance will be tackled in Section 2.1, providing historical background, motivation and some practical considerations associated with real-world large scale visual surveillance installations. Section 2.2 provides an overview of the applications of Computer Vision technology to surveillance. This work falls into two broad categories - the modelling of scene geography (Section 2.2.1) and the modelling of behaviour (Section 2.2.2). The approach this thesis takes is to try and *explain* the behaviour of agents within a surveillance scenario in terms of known *goals*, an approach which is motivated in Section 2.3, covering intentionality and the nature of explanation from a cognitive-philosophical perspective. Finally, as the model developed within this thesis is a model of intentional, goal-directed behaviour, some consideration of navigational strategies is appropriate. Section 2.4 outlines approaches to the problem of path-planning and navigation from within the psychological and the robotics literature.

2.1 Surveillance in the real world

Whilst there has been a lot of work in the computer vision literature on automated visual surveillance, much of it ignores the practises of real-world closed circuit television

(CCTV) installations and operatives. This section attempts to go some way towards addressing this problem by investigating the ways in which CCTV installations actually work, and the ways in which technology could be used to improve their working.

The precise number of CCTV cameras in the UK is unknown: the human rights group *Liberty* put it at 4 million in 2005 [84] and state that 78% of the Home Office crime prevention budget has gone towards CCTV since 1994. Whether four million is an accurate estimate is difficult to ascertain, but as McCahill and Norris say:

... in the first decade of the new millennium, when the average Briton leaves their home what will be remarkable is if their presence is not seen, their behaviour not monitored and their movements not recorded by the omnipresence of the cameras, CCTV operators and video recorders [95] (p.15).

Anti-CCTV campaigners bemoan the fact that we are constantly watched, but a survey of the literature suggests that their concerns are at least in part unfounded. There may be a massive number of cameras, but these are not continually monitored. In local authority CCTV installations, with some hundreds of cameras, only a small number are ever watched. Tower Hamlets, for example, has 237 street CCTV cameras linked to a control room in which up to 5 operatives and 2 police officers monitor a bank of screens. Whilst in theory all cameras are monitored, only 10 to 12 are monitored in real-time with the rest only watched following an incident - monitored only in recorded time¹. In Manchester, around 80 city centre CCTV cameras are monitored by up to 5 operatives via a bank of 48 screens² with further banks of screens in the same control room devoted to NCP car-parks, and additional cameras covering arterial routes into the city. Two banks of screens from Manchester CCTV control room are shown in Figure 2.1. Liverpool has around 250 cameras, and a similar number of operatives³. In the London borough of Wandsworth, there are around 250 cameras which are monitored part time (8am until Midnight) by two operatives and one police officer [96]. Practically, it is acknowledged that each operative can only really monitor one screen at a time.

2.1.1 Deciding which scenes to watch

The question of which cameras to watch is a difficult one to answer. Existing systems involve the operators themselves selecting which cameras to monitor. This leaves the system open to abuse and discrimination in a way that has attracted the ire of human

¹Figures from Ms H Mallinder, Tower Hamlets Antisocial Behaviour Control Unit, personal communication, 2005.

²Figures from the director of Manchester CCTV, 2005.

³Figures from Mr L Walters, Liverpool CityWatch, personal communication, 2005.

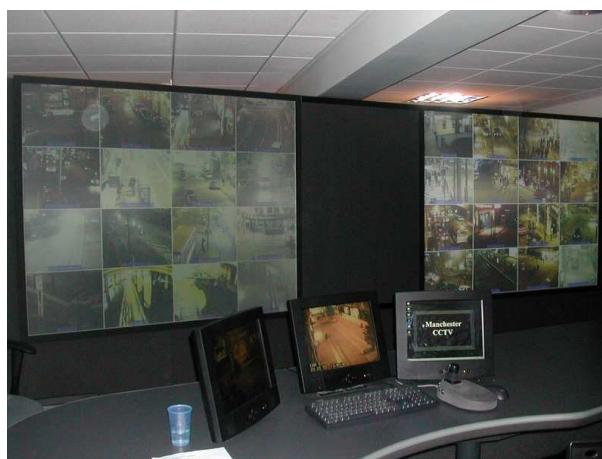


Figure 2.1: A CCTV control room

rights and anti-surveillance groups. Studies show [96, 106, 135] that CCTV operatives deciding which cameras to monitor are guided less by the behaviour of the people in the scene and more by their appearance. This is probably inevitable, given the snap decisions which have to be made, with a small number of operatives monitoring several hundred cameras. If the operative only has a few seconds in which to make a judgement, all she or he can really call upon are static cues such as appearance. A further problem with CCTV operators is the obvious one of boredom: in the vast majority of surveillance situations, nothing happens [135].

Norris and Armstrong have carried out an in-depth study of custom, rules and practice in surveillance installations including long-term monitoring of both a city centre and a small town CCTV system. In [107], drawing on the work of Harvey Sacks [118], they codify the 7 *working rules* of surveillance installations. Sacks investigated the way in which police officers infer the criminality or otherwise of people from their appearance, behaviour and location. Norris and Armstrong have adapted this framework to the situation of surveillance. Both police officers “on the ground” and surveillance operatives have a similar decision to make: whether or not to target an individual for further investigation based upon their appearance and other cues.

The first three of these *working rules* are direct descendants of Sacks’s work with police officers, such as “*Certain people are immediately worth of surveillance because they are known by operators to have engaged in criminal or troublesome behaviour in the past*” [107] (p.118). The remaining four are more surveillance specific, and have grown out of Norris and Armstrong’s field studies into the way surveillance operatives choose which people to target. Rules four to six concern spatio-temporal patterns of

behaviour, and the targeting of people considered to be *out of place* or *out of time*, such as the targeting of homeless people in the city centre (ibid, p.141). Rule 7 is the only rule to refer directly to the surveillance cameras, and states that “*Operators learn to see those who treat the presence of the cameras as other than normal as other than normal themselves*” (ibid, p.119). Smith, in [135], observed CCTV operatives in a typical “Little Brother” installation at an un-named UK college and noted that camera selection was often driven by boredom. Playing “hide and seek” with security officers on the ground, reading newspapers, and frequent tea or coffee breaks all helped to alleviate the boredom. One operative admitted to targeting a specific camera on his own car all evening.

Given that not all cameras can be monitored, and the unsolved problem of selecting which camera to monitor, CCTV installations are largely used in reactive policing. Even this application is not without technical problems – stored video footage is notoriously difficult to search. It is often *multiplexed* either temporally (by interleaving frames from multiple cameras) or spatially (storing the output from multiple cameras in an array on the screen at a lower spatial resolution) and usually only 3 frames per second are stored. One example mentioned in [95] is that of the London nail bomber, who was arrested 13 days after the first bomb had gone off (and who set two more bombs in the intervening period). Finding evidence from CCTV involved searching 1097 videotapes with some 26,000 hours of material, much of it multiplexed. It is estimated that some 4,000 person-hours of video analysis was involved, before footage of the bomber at the site of the first of the three bombs was recovered.

2.1.2 Evaluating the effectiveness of CCTV

One question which needs to be addressed in any consideration of practical CCTV operation is whether or not the cameras are effective in reducing crime. The installation of CCTV in the UK has happened at a remarkable rate – in 1994 there were an estimated 16 town centre schemes with 400 cameras, and by the end of 2002 there were approximately 500 such systems with 40,000 cameras [4]. Surprisingly, this expansion has happened without much systematic evaluation of whether or not the systems actually work. In [95] and [4] surveys of evaluative studies are presented and the only conclusion that can be drawn from these is that nobody really knows whether it works. Some evaluations suggest CCTV works, and some do not.

The reasons for such varied outcomes of evaluation reflect the difficulty of evaluating a system in a complicated real world environment. Tilley, in [148] determines 9 possible confounding variables many of which are echoed in other studies of CCTV eval-

uation [4, 40, 134]. These include problems associated with changes other than the introduction of CCTV, such as background fluctuations in crime rate and changes to the area under surveillance. They also include the commonly cited and contradictory problems of *diffusion*, where areas near to CCTV installations also experience a drop in crime rate, and *displacement*, where crime is simply displaced to neighbouring areas without surveillance. CCTV seems to have different effects on different types of crime - and these effects do not seem to be consistent across different CCTV installations.

2.1.3 Concluding remarks upon the practical and social aspects of real-world CCTV installations

CCTV systems are pervasive, have documented problems with monitoring, targeting and retrieval, and have not been clearly proven to reduce crime. Yet a recent report [108] estimates that in the years 1994-2004 between 4 and 5 billion pounds have been spent on the installation and maintenance of CCTV systems. This does not include the cost of monitoring or retrieving video. A few high profile cases of CCTV success have entered the public consciousness, starting with those grainy images of Jamie Bulger in 1993, and again most recently with the London bombers of the 7th and 21st July 2005. But these successes are the exception rather than the norm (and may well be due to information from other sources of intelligence). Such publicity serves to reinforce the public perception of CCTV as a force for good, and to reassure the public that these cameras are actually useful, although the evidence as it stands is far from conclusive.

2.2 Computer Vision

There is a large body of work within computer vision which deals with analysis of the types of video scene captured by CCTV cameras – pedestrian areas, car-parks, roads, shopping malls and the like. A good deal of attention has been paid to the problem of tracking moving agents and various related problems such as occlusion analysis. Tracking generally involves some form of background subtraction to identify foreground pixels, and then the application of some model of motion such as a Kalman filter [79] or particle filter (sometimes called CONDENSATION) [68] to perform the actual *tracking* - the identification of foreground pixels over time as belonging to a particular moving object. Appearance based approaches to tracking using contours [6–8] and pattern matching [128, 150] have shown success. The tracking of multiple objects (including people) using Bayesian [69] approaches has also been successful.

Adaptive background models can over time incorporate parts of the foreground, and so the detection of abandoned objects becomes problematic, and a number of papers are devoted to the detection of suspicious packages (e.g., [50]) or incorporate the ability to detect if a person is carrying an object (e.g., [57]). A related body of work concerns itself with the identification of individuals over time from CCTV footage, either from their clothing (e.g. [11]) or via face recognition (see [111, 161] for an overview).

The work which will be described in this section is that which concentrates on deriving models of geographical features within a scene, the analysis of behaviour (at a higher level than just tracking), and the interplay between behaviour and geography. Those systems which model behaviour at a finer grained level, such as gait analysis, will not be considered here.

2.2.1 Modelling scene geography

Modelling certain features of a scene can improve a tracking application in a number of ways. Knowledge of entrance and exit points can assist in tracker initialisation and knowledge of the paths agents typically take through a scene can be fed back into a tracker to help disambiguate difficult cases. For higher level applications, performing behaviour modelling or cognitive analysis of a scene, a rich scene model can assist greatly. Entrances and exits form goals, places where people are often inactive can be flagged as such, and defining parts of a scene as paths (perhaps even directional paths) can help with atypical behaviour detection. Occlusion reasoning and scene modelling are related in that they can both help to disambiguate meaning in these situations – indeed they can be handled together (such as in Stauffer [138]).

A persistent problem in the tracking and scene modelling domain is that of assigning meaning to the start and end points of trajectories. The end point of a trajectory could correspond to an agent leaving the scene, or to an agent passing behind an occlusion, or to the tracker simply “losing” that agent. The modelling of these as entrances and exits has been performed by a number of researchers. There are three main decisions each of these researchers has to make. Firstly, *how to model the spatial extent and location of each entrance or exit* (model type); secondly, *how to determine the number of exits in the scene* (model order); and finally *how to find the size and location of each exit* (model parameters).

In [138] Stauffer couples the problem of determining entrances and exits – he uses the terms tracking *sources* and *sinks* – with the problem of fixing broken tracking sequences. This work is a development of ideas presented by Russell and colleagues in [64, 112], who

were concerned with maintaining the identity of objects over multiple non-overlapping cameras in a vehicle tracking scenario. In this earlier work, the problem was to determine correspondences between tracked objects across different scenes with very few entry and exit points (the scenario was a freeway in the United States). Stauffer has used a similar technique in less constrained single camera scenes. In such a scene, using a conservative⁴ tracker, tracking output can consist of numerous “tracklets”, or partial tracks. The end of a tracklet may correspond to an object leaving the scene, or it may correspond to the tracker losing that object. Stauffer’s insight is to couple the problem of stitching together these broken tracklets (the object correspondence problem dealt with in [64, 112]) with the estimation of scene entrances and exits. If a tracklet ends near an exit, it is more likely to have ended because the tracked object left the scene than because the tracker has failed. The entrances and exits are modelled as two state hidden state models (one model for each entrance-exit pair) with Gaussian output probabilities, and the model parameters are iteratively estimated using standard Expectation Maximisation (EM) estimation. Model order is determined by using a variant on minimum description length (MDL). By stitching together the most likely pair of tracklets at each iteration (a *hard* assignment: once two tracklets are paired they are not reconsidered as part of the exit model) and updating the track stitching correspondences alongside the exit estimation, both problems can be solved simultaneously.

McKenna and Nait-Charif [98, 99, 104] have performed scene modelling in a more constrained environment, that of a single room inside a home. The system they develop is for fall detection in a supportive home environment, and as such they wish to be able to detect falls, but also to summarise the video for privacy reasons. They use Gaussian mixture models (GMMs) to represent entrance zones and inactivity zones, trained using EM estimation. Model order is determined using maximum penalised likelihood (MAP) estimation, which they claim results in Gaussian components that correspond to meaningful semantic regions. In this application, all entrance zones are doors to the room and hence bi-directional. As these doors are at the edge of the scene the entrance zones can be modelled by fitting a GMM to trajectory start and end points in 1 dimension. Inactivity zones are also learned, by fitting a 2 dimensional GMM to points in the scene where the agent’s velocity falls below a certain threshold. The application they describe uses this scene model to summarise activity (“*Enter through the hall door, sit on the sofa and then exit through the rear door*” [104] becomes HSR) and to detect unusual inactivity, such as a fall, by detecting inactivity outside of the learned inactivity zones.

⁴A “conservative” tracker is one which only identifies an object or agent as present if there is a high probability of this being the case: very few false positives are returned, but the chance of temporarily losing an object is high.

Makris and Ellis [90–94] have developed a scene modelling technique learned from the tracks of moving agents. The central feature of this technique is the creation of “*routes*”, “*junctions*” and “*paths*”. The approach starts with the detection of routes, which are built up over time from a number of trajectories. Each route is represented as a spline and a set of vectors normal to the spline direction which define the extent to which trajectories deviate from the route spline. Routes are learned by grouping geometrically adjacent trajectories. Each new trajectory is compared to existing routes, and the closest route is updated with data points from the new trajectory unless the distance is over some threshold, in which case, the trajectory is used to start a new route. Paths and junctions emerge from a second level of processing in which route sections that are similar are merged, and a junction placed at each end. In [92] Makris and Ellis address the learning of entry and exit zones. The authors compare K-means and GMM approaches to exit modelling, and conclude that Gaussian mixture models trained using the EM algorithm provide a more accurate estimate of exit location and extent. Model order is determined by overestimating the number of Gaussians, and then deleting those which are associated with a low density of observations.

A research area closely related to that of scene modelling is that of occlusion handling or occlusion detection. Occlusions are related to obstacles, and often researchers attempt to model both at the same time. The distinction this thesis will draw between the two is that occlusions are defined with respect to the camera: they occur when something falls between the lens of the camera and the object of interest, and can be due to static scene features (hedges, walls) or moving objects (like a van coming between a person and the camera), and they may or may not affect the behaviour of people moving around within the scene. Obstacles, in contrast, exist in specific ground plane locations. They may or may not occlude the camera – however they do affect the paths of the people moving within the scene. In [113] vehicular occlusions are handled by maintaining a ground plane representation and an estimate of vehicle size. Senior, in [128] handles static foreground occlusions by maintaining three models: pixels are classified as either foreground occluding pixels, background pixels or moving object pixels. In [127] a method for learning a model of scene occlusions from the tracks of moving agents using minimum description length is described, which creates successively more detailed depth models by dividing the scene into “layers”. In [55] Greenhill and others develop this approach. Using a simple image to ground plane computation based upon the observation that people farther away are both smaller and higher up in the image plane, a depth map is developed. All pixels belonging to a moving person are assigned a distance from the camera determined by the location in the image plane of the top of that person’s head. These depths are

then regularised – smoothed spatially whilst preserving depth discontinuities – and the resultant occlusion images deal well with difficult scenes such as a tube station with its stepped rows of ticket machines. The regularisation is performed using a Hopfield neural network.

In [56], Grimson et al describe the Forest of Sensors project at MIT. This uses the tracker outlined in [140] over a distributed array of sensors, which between them cover a large area of the MIT campus. They hypothesise that simply through tracking motion, a range of different computations about the nature and typicality of activity on a site can then be made, and also that certain aspects of scene geography can be mapped out. Their multi-camera system automatically calibrates to a world-coordinate system and produces ground plane coordinates, and it is this which enables the mapping of occlusions. Using an estimate of height to compute the distance from the lens to a pedestrian within the scene, it is obvious that the portion of the field of view between the camera and the person is unoccluded. Likewise, when a person goes behind an object, it can be assumed that there is an obstacle or occlusion at that point.

Xu and Ellis [42, 158] have also carried out research into occlusion analysis, however they deal with tracking through occlusions and rely upon a hand crafted model of actual occlusion location. Their classification of occlusions into “*long term*”, “*short term*” and “*border*” occlusions is a useful one for many tracking applications. Long term occlusions are those such as doors, or buildings which abut the edge of the scene. These are occlusions from which agents are not expected to emerge. Short term occlusions are those such as trees – agents may disappear behind these occlusions, but they are expected to come out on the other side. Border occlusions occur at the edge of the camera’s field of view.

In [117], Rowe proposes a system based not upon computer vision but upon multiple pressure sensors, in which a particular conception of *suspicious* behaviour is measured. Suspicious behaviour is defined as that which involves deception or concealment: behaviour with multiple inconsistent goals. The outwardly detectable signs of such behaviour include attempts to hide from other agents, and changes in direction or acceleration. The scene model Rowe proposes involves first quantizing the scene by dividing it into a number of squares, and then scoring areas of the scene as obstacles. For non-obstacle portions of scene, predicted occupancy rates are calculated based upon the ideal paths through the scene. Also calculated for each square is a score representing its visibility, depending upon the optimal paths and the location of the obstacles. The system is only demonstrated in simulation.

2.2.2 Behaviour modelling within computer vision

Much work in computer vision, particularly in the area of visual surveillance, centers around the detection of events of a particular type. As computer vision systems tend to involve building some form of model of reality then comparing the interpreted visual input to this model, there are two main ways of detecting events. You can either build a model of what you are *not interested in*, and define behaviour which does not fit this model as some form of event, or you can build a model of the specific behaviour you are trying to detect and directly detect the events. The former approach is more common in surveillance, as the aim is to be able to detect a large class of events without any *a priori* understanding of the shape these events might take. The second form of event detection can be thought of as a special form of *classification*. A third approach (which is more common in work within constrained environments) is to use machine learning techniques to automatically derive some number of categories of event.

A large “toolkit” of techniques exists for behaviour classification and summarisation. Once the behaviour in question has been tracked and modelled and transformed into some numerical representation, a whole armoury of statistical methods can be used in the classification of these representations – the partitioning of the resultant behaviour space. Examination of the members of each class or partition then enables the authors to semantically label the behaviours: people walking to the left, for example, or cars reversing up a slip road. Hidden Markov Models and Bayesian Networks are the most popular approaches used in the literature. Indeed, a recent review [21] describes the field almost entirely in terms of these techniques. However, other statistical methods are also brought into play. The following sections outline the applications of these various techniques to the problem of modelling the behaviour of people.

2.2.2.1 Behaviour modelling with Hidden Markov Models

The temporal relationships between events are often modelled using Hidden Markov Models, or HMMs. In an HMM, an underlying process is modelled based upon observation of its effects. In many situations within computer vision, it is impossible to observe the underlying process, but it is possible to infer from observation that there is something causing the observation (such as when we infer the existence of a moving object from the changing colours of pixels). The temporal aspect of HMMs comes from the ability to extrapolate from an observation at a particular time step and to predict using the underlying process what it (and its observable effects) will be at the next time step. Figure 2.2 shows a graphical representation of the factorisation of joint density that is a Hidden Markov

Model.

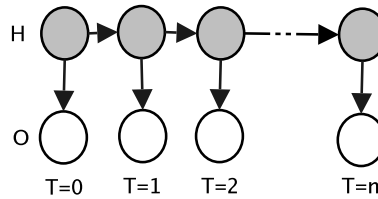


Figure 2.2: A Hidden Markov Model (HMM)

First order HMMs make two main assumptions: firstly, that all information required to model the next stage of the process is present in the current stage, and that the behaviour being modelled is the result of a single underlying process. These assumptions are both violated by most real-world scenarios in vision, and a whole family of Hidden Markov Model variants have grown up, exploiting the predictive power of the HMM but extending its ability to deal with complex, real-world data.

One such model is the CHMM – Coupled Hidden Markov Model – of Brand et al., introduced in [18]. In this model two (or more) HMMs are coupled, with the state of each at time t affecting the state at time $t + 1$. A diagram illustrating this HMM architecture is shown in Figure 2.3. They demonstrate the improved performance of this on a dataset featuring T'ai Chi manoeuvres in which each hand is modelled as a separate but coupled process. Oliver et al. go on to demonstrate this model's usefulness in modelling pedestrian activity for surveillance, analysing actions which occur between two pedestrians [109, 110]. The CHMM is particularly suited to this sort of analysis as there are two pedestrian behaviour patterns which may or may not be linked – and the links may be weak or strong. That is, the behaviour of each pedestrian at time t may or may not be affected by the behaviour of the other pedestrian at time $t - 1$. They train their model on synthetic data, and compare its performance to straightforward HMMs on both synthetic data and a mixture of synthetic and real data. The CHMM architecture is shown to be very good at modelling specific patterns of interaction between two pedestrians, such as *change-direction*, *meet*, *chat*, *continue together*: indeed, the CHMM architecture obtains a 100% success rate at recognising the behaviour patterns upon which it was trained.

Gong and Xiang, in [53], describe event detection and recognition in an airport scenario. The events they are detecting emerge from the data, without manual labelling or any form of top-down input into the event model. These are obtained from Pixel Change History (PCH) alongside an adaptive GMM for background modelling. This detects pixel-level changes which are more than just motion. The resultant 7D feature vector is then clustered in feature space using a GMM, with order selected using Minimum Description

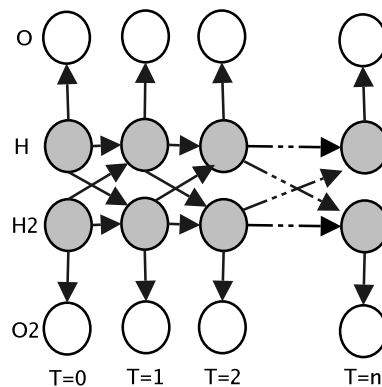


Figure 2.3: A Coupled Hidden Markov Model

Length. Each cluster is then labelled as a different class of event. Upon investigation the events are things like moving trucks, moving cargo lifts and so on. It is important to stress that none of these categories are specified in advance, but rather emerge from the data. The temporal relationships between these detected events are modelled using a new variant Hidden Markov Model which they call a Dynamic Multi-Linked Hidden Markov Model (or DML-HMM). The DML-HMM is similar to Brand et al.'s CHMM, but in the DML-HMM not all hidden states are connected. By learning which hidden state variables are interconnected the DML-HMM has a structure which better reflects the relationships between events.

Brand and Kettner [17] use entropy minimisation to determine the structure of an HMM for the detection of events in video: typically, HMM transition topology is either hand-crafted, learned by clustering, or discovered through some form of heuristic search. Instead, by minimising the entropy over the model, the data relative to the model, and the cost of encoding aspects of the data not captured by the model, they learn a structure which reflects the structure in the video clips they have been trained on. They demonstrate that these models can be used to detect unusual behaviour (by noting times at which the HMM assigns a very low likelihood to the data).

A number of other variants on the Hidden Markov theme have been proposed and demonstrated within the behaviour modelling domain. Variable Length Markov Models (VLMMs) have been used [47,48,121] for the modelling of behaviours in a number of settings. Cyclic HMMs [88] can be used for the modelling of recurrent or cyclic behaviour. Parallel HMMs (PaHMMs) have been shown to be successful in the recognition of related behaviours, such as the movement of both hands when using sign language [151].

In addition to this family of related HMM models, researchers have coupled HMM systems to other techniques, statistical and otherwise – such as VLMMs and vector quan-

tization [48]. Ivanov and Bobick in [70] separate the problem of recognising primitives from the problem of recognising structure. They use HMMs for detecting primitive events (such as *car-enter*, *car-stop* and *person-exit*) and use a stochastic context free grammar to recognise higher level events such as *drop-off*. The grammar is hand-crafted for each scenario.

Another similar approach has been developed by Wu and others [157], in that they have developed a representation of the scene and then use statistical learning techniques to spot out-of-the-ordinary behaviour patterns. In [157] paired HMMs are used to represent the behaviours and support vector machines are subsequently used to partition the behaviour space. They have a high success rate in spotting unusual behaviours, but the unusual behaviours they detect consist of people driving in a zig zag or circular pattern in a car-park so are quite far from normal behaviour.

2.2.2.2 Behaviour modelling with Bayesian networks

A second family of graphical models has been extensively used in the modelling and prediction of uncertain events – such as the visible behaviour of humans. This class of models includes Bayesian Belief Networks (BBNs), which represent system states at a particular time instant (or at all time instants, in the case of static systems) and Dynamic Bayesian Networks (DBNs) which incorporate temporal information within their structure. Bayesian Networks are directed acyclic graphs, in which the nodes represent particular states or variables, and the arcs connecting the nodes represent causal relationships between those variables. If a node has a known value, it is said to be an evidence node. A node can represent any kind of variable, be it an observed measurement, a parameter, a latent variable, or a hypothesis. The strength of the causal influences are encoded by associating with each arc a conditional probability. BBNs represent factorisation of a joint distribution over all variables. These probabilities can be learned from experience by *training* which uses iterative schemes to find a maximum likelihood for the parameters, implemented as localised message passing operations.

Remagnino, Tan and Baker in [114, 115] develop a Bayesian network based model for the classification and annotation of multi-agent actions. This system uses Bayesian networks on two levels. Firstly, the agent level in which each moving object within the scene is associated with its own multi-layered Bayesian network called a *behaviour agent*. These behaviour agents have input nodes associated with characteristics such as speed, acceleration and heading. These input nodes feed up to hidden nodes (dealing with the dynamics of the object or agent) which in turn feed up into the final behaviour nodes which provide the most probable interpretation of the agent's behaviour. The second level

upon which they operate involves a Bayesian network called a *situation agent*. These higher level Bayesian networks are called into play when the Euclidean separation between two *behaviour agents* falls below a specified threshold, and encodes information about the interaction between the two behaviour agents (such as *the pedestrian is passing by car three*). In [114] the issue of interactions involving more than one agent is raised, and the authors suggest that a third level of Bayesian network would be required to handle such complicated interactions (a *scene agent*).

Another approach to event detection is exemplified by Intille and Bobick in [67]. This is the use of multi-layered Bayesian Networks to model various aspects of a particular sub-set of structured multi-agent behaviour. The behaviour they are modelling is that of American Football set “plays”, which are structured, highly choreographed actions. The approach adopted is to use expert information - from an American Football coach - to encode the actions of each player during a specific play, and to build up a multi-layered model of what is actually going on in the scene based upon the visibly determinable goals of the individual agents involved and the temporal and spatial relations between those agents. Finally the relationships between these atomic representations are used to determine the type of multi-agent action being performed. Bayesian networks are used at two stages in this process - to integrate the uncertain data from the visual trajectory information, and to perform the multiagent behaviour analysis.

The work described in [67] is of particular interest to the current thesis for another reason: it is one of the few works in the computer vision literature which acknowledges that the agents in a scene are goal-directed individuals. Intille and Bobick do not perform much high level reasoning about these goals, preferring to model low-level attributes. This is perhaps understandable given the nature of their domain: modelling for each agent their understanding of the game is a much greater task than to model their immediate goals (in terms of things like *catch-pass* or *block-defender*). Integration into a higher level representation is done in a top-down way: these goals act as evidence towards one play or another. This approach works very well in a domain where the structure is known in advance, such as they find within the highly constrained world of American Football in which very little within-play replanning occurs. That is, once the players on the field are engaged in a particular “play”, they finish it and do not suddenly change their goals. Crucially, the authors state that they do not detect “None of the Above”, so are unable to determine patterns which do not fit one of their plays.

Hongeng and others in [60] (expanded upon in [100]) describe a system based upon Bayesian networks which recognise and categorise single agent (single “thread”) events. They have a finite state machine which operates on the output nodes of the Bayesian

networks and recognises temporally extended multi thread events. The events are hand-coded into the Bayesian network (i.e., “Converse”, an event which occurs when an agent approaches another reference agent, slows down and stops has nodes for “reference person”, “getting closer” and “slowing down”).

Buxton and Gong, in [22] describe a Bayesian network based system for monitoring activity in certain types of surveillance situation: specifically, traffic motion at a junction. Their architecture features a preattentive system operating on low-level behaviours, such as velocity and orientation, and a central attentional system which evaluates higher level behaviour patterns such as “overtaking”. This was further developed by Buxton and Howarth [20, 62, 63] who enhanced the attentional use of Bayesian interaction agents to provide conceptual descriptions of behaviour. The attentional component of the system consists of a “tasknet” for a higher level behaviour, and once a tasknet is activated it begins gathering evidence for that particular task (e.g., *gross-change-in-motion* is evidence for the *give-way* tasknet, and once the lower level networks have reported this, the *give-way* tasknet will search input for other related components).

2.2.2.3 Other statistical and machine learning approaches to behaviour modelling

Johnson and Hogg [74,75,77,78] have developed a method for behaviour modelling which enables prediction of future behaviour, a form of trajectory classification and the detection of unusual or atypical behaviour patterns. This is achieved through a multi-layered approach in which firstly trajectories are sub-sampled to produce flow vectors representing position and instantaneous velocity, and then subjected to a version of Vector Quantization (Altruistic Vector Quantization, or AVQ) producing a codebook of representative prototype vectors: this provides the “state space”. These prototypes are then used to train an artificial neural network (ANN) which contains a layer of leaky neurons. The leaky neurons are vital to this approach as it is these which are responsible for encoding the temporal nature of trajectories: each leaky neuron takes just one input and produces just one output, but the output depends upon the neuron’s history (as each maintains a trace of prior inputs). A second neural network with 100 output nodes is attached to the output of the leaky neurons and performs AVQ on an agent’s whole trajectory. This produces a set of trajectory prototypes (which form a *behaviour space*) and new trajectories can be compared to the existing prototypes for classification and event detection. Images depicting Johnson’s scene and a sample behaviour vector are reproduced in Figure 2.4.

Sumpter and Bulpitt [146] present a related technique using Neural Networks to quantize over trajectories for behaviour modelling and prediction. The network they describe consists of two competitive learning networks, linked by a layer of leaky neurons. In this

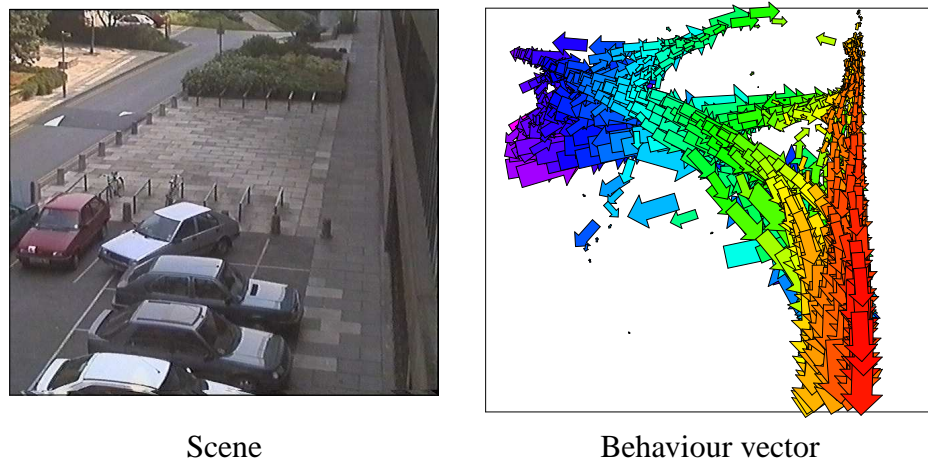


Figure 2.4: Pedestrian scene and sample behaviour vector, reproduced from [74] with permission.

way, their approach differs from Johnson and Hogg’s, who require an extra learning stage for modelling the whole trajectory. Both of these approaches are capable of prediction and extrapolation, as partial trajectories or configurations can be matched to the closest trajectory prototype in behaviour space.

Grimson et al, in [56], analyse the behaviour of the agents within the scene using various clustering approaches. They propose two families of approach, one involving Wallace’s Numerical Hierarchical Cluster (NIHC) [152] and one using a GMM based approach combined with K-means. The NIHC approach assigns data points randomly to clusters in a binary tree, and then iteratively reduces the entropy of this tree. Finally, a Minimum Description Length cut is made which finds a level of clusters that best describe the dataset. Given these clusters, particular patterns of behaviours emerge such as “people in a queue” (a cluster of small slow moving objects with low directionality of motion). The second approach they describe involves overfitting a large number of Gaussians, each representing a small portion of the 6 dimensional state space (x , y , dx , dy , size, aspect-ratio). This is then clustered using K-means, and the resultant graph is partitioned using a Hopfield network. The first graph cut divides the behaviours into leftbound and rightbound traffic, and the subsequent child nodes represent faster and slower vehicles, pedestrians and the like.

In [137] another classification system is described (also part of MIT’s “Forest of Sensors” system described earlier). This system has similarities to that of Johnson and Hogg [75, 77, 78] in that they use Vector Quantization to produce a number of prototypes. These prototypes form a codebook which is then used in the place of the original

dataset – each data point is considered not as itself but as its most representative prototype. From these a co-occurrence matrix is produced and then this new space is split iteratively into two sets, producing a hierarchical tree structure of behaviour patterns. The lower branches of this tree represent semantic categories, such as “pedestrians on a lawn”, or “activity near a loading dock”.

A related approach is that of Gong and colleagues [54, 65, 131, 132, 154] which also attempts to automatically summarise or categorise activities within video. Whilst these systems sometimes incorporate Bayesian Networks or Hidden Markov Models, they are characterised by a combination of many different statistical methods and do not really fit in any one category of approach. Another way in which these approaches stand out from other work in the surveillance domain is that they do not explicitly model or track events at the object level at all. Instead, they compute pixel-wise change which provides a crude measure of motion within the scene, and build a layer of filters on top of this (typically wavelet based filters). In [131, 132] the output of these filters is analysed using GMMs to detect events, and then these are clustered (using K-means) to detect higher level events. In [54] the filters are used to create a histogram which provides a continuous scene descriptor, and then subjected to PCA, retaining just the top three eigenvectors to reduce noise. Both approaches can be used to identify events in video sequences which correspond to specific activities – in [131, 132] the system detects events within a shop scenario such as “picking up a can”, and in [54] events such as a car reversing are detected. Hung and Gong in [65] present a technique based upon correlating salient motion. Saliency is defined as a measure of the entropy of the data over a spatio-temporal neighbourhood. By correlating salient events, interactions can be determined. The least frequent interactions are found to be unusual or interesting events (cars going the wrong way at an intersection, for example).

Hongeng, in [59] takes this a step further in learning and predicting simple time-dependant patterns of behaviour over time – using a Markov Network he demonstrates the possibility of learning global event configurations from local ones. The system he is learning is the simple one of table setting: the system learns that the knife goes to the right of the fork etc.

Jan et al in [71] working explicitly in the surveillance domain have used Artificial Neural Networks to detect suspicious behaviour by training a network to perform a non-linear partitioning their behaviour space. In this work, the behaviour of actors is transformed into a 49 dimensional feature vector by taking the velocity of the actors head at 5 frames/second for 10 seconds. They state that suspicious behaviour is associated with “jerky” head movements, and the partitioning they arrive at distinguishes between the

actors behaving in this jerky fashion and those behaving normally.

2.2.2.4 Ad-hoc approaches to behaviour modelling for surveillance

In [103] a method for detecting atypical behaviours in the interaction of objects is presented. This uses Baumberg and Hogg's [6–8] tracker for non-rigid objects (i.e., people) and cars were tracked by hand. Inspection of the collected trajectories confirmed the authors' *a priori* judgements about typicality – people normally start near cars then move away at increasing speed, or people start away from cars then move directly towards a specific vehicle, decreasing in speed only at the end of the trajectory. Thus they define atypical behaviour as moving slowly near a number of cars, or approaching a number of different cars in turn. The algorithm proposed in this work detects such atypical events by finding the points of closest approach to each vehicle, converting the trajectory into these *landmark points* and then analysing the resultant representation.

In [12] a database for surveillance applications is introduced. This integrates output from the multi-camera surveillance system developed by Makris, Xu and Ellis [91, 93, 158] and allows for easy searching and retrieval by a structured data representation. The structure takes the form of four levels of data abstraction – the Image Framelet layer, the Object Motion layer, the Semantic Description layer and the Metadata layer. The Image Framelet layer stores camera-specific representations of moving objects: foreground pixels after background subtraction provide a pictorial representation of the salient features of moving objects. The Object Motion layer contains tracking output unified over multiple cameras: information such as bounding box and velocity. This lower-level tracking related information comes from the systems developed by Xu and Ellis, as described in [158]. The Semantic Description layer stores information about routes and scene entry and exit points using the scene model of routes, entrances and exits of Makris and Ellis [93]. The Metadata layer is generated from the information in the lower layers such as point of entrance, and time spent in each route node. The database can extract tracks based upon any of the layers - people occupying particular places, or following particular routes.

In [89] a system is presented which uses a multi-layered Finite State Machine (FSM) approach to the detection of unusual activities in video. Their architecture allows either for a learned model of logical events (learned from the absolute positions of individual objects over time) or for the explicit programming of expected states, such as “person with bag” and “owner and bag on ground”. The ELEVIEW surveillance system of Shao et al [130] uses a similar state transition diagram to classify behaviour in elevators as *normal*, *suspicious*, *overstaying* or *stain* (the result of graffiti).

2.2.3 Concluding remarks on computer vision

This section has described the different approaches computer vision scientists have used in the area of automated visual surveillance, concentrating upon the modelling of the scene and the modelling of behaviour. Approaches have ranged from the entirely statistical to the entirely hand-crafted. Within behaviour modelling, a layered approach has often been adopted with statistical (e.g., object tracking) methods used to obtain information about the movement of the people within the scene, and then other methods (statistical or otherwise) operating upon the output of these low-level systems to perform behaviour analysis.

2.3 Cognitive and philosophical considerations

The approach this thesis proposes represents a significant departure from the currently dominant statistical school in computer vision. The insight upon which this is based is that when humans attempt to perform a surveillance task, (unless required to make a snap decision) what *we* try to do is to ascertain the goals of the agents moving around within the scene. The question is, whether incorporating such notions as *intentionality* into a vision system can enable it to perform well in a surveillance task, and whether the subsequent system can tell us anything useful about the behaviour of the agents within the scene, either in terms of prediction, classification or explanation.

In doing this, is it necessary to assume that there are such things as goals, and beliefs, somehow inside the heads of the agents within the scene? Within the philosophy of mind, this position is far from uncontroversial. To borrow a contentious idea from Philosophy and use it without acknowledging its uncertain status would be a mistake. Thus the purpose of this section is to outline the major lines of disagreement within philosophy upon this topic, and to argue that such debates can be sidestepped by adopting a pragmatic and instrumentalist account of beliefs, desires and goals.

Philosophers of mind call the idea that our behaviour is mediated and governed by beliefs and goals *Folk Psychology*. The strongest formulations of folk psychology would claim that when we are engaged in walking across a car-park we have some goal in mind, and some beliefs about how we should go about reaching that goal, and some combination of these beliefs and desires provides the causal basis for our behaviour. This is sometimes formulated in a sort of predicate calculus – if you desire x (to get to your car), and believe that doing y will bring about x (walking around the hedge) then as a rational agent you can be expected to do y . Folk psychology provides a means of predicting the behaviour of

ourselves and others, and provides us with an explanatory framework for understanding action. Philosophical positions on folk psychology range from strong formulations or realism, such as that of Fodor [44] and others [61], through instrumentalist theories such as Dennett's [35–37], to eliminativism [26, 27, 142].

One of the central arguments against folk psychology involves casting folk psychology as a theory of human behaviour. This is considered in some detail by one of the most vociferous opponents of folk psychology, Paul Churchland, in [26]. If it can be argued that folk psychology doesn't have the status of theory, then there is no way of proving or disproving it (with the implication *so why are we bothering to discuss it anyway?*). If it is a theory, then we can evaluate it and come to a decision as to its veracity. Churchland believes that our everyday folk psychological terms are similar in status to our everyday folk conceptions of physics. Folk Physics has been shown to be severely lacking in detail and utility (indeed, [97] has shown that our folk conceptions of physics are so outdated most people expect things to behave in an Aristotelian, or maybe Medieval, fashion, rather than in a Newtonian one). Eliminativists argue that our folk psychological concepts will be overtaken by a more accurate account of the mental when we develop a mature science of the mind [142], in exactly the same way that Aristotelian physics has been superseded.

Even Paul Churchland agrees that whatever else we do with the concepts of folk psychology, we successfully use them to predict and explain the behaviour of others. The ontological status of these *propositional attitudes* (beliefs, desires, goals) might be in question, but their utility is clear. Dennett likens the terms of folk psychology to physical abstracta such as *centres of gravity* [36, 37]: In the same way that it is useful to model a body as a point mass, even though we know that this is not actually the case, it is useful to model the behaviour of ourselves and others as though things like *belief* exist.

Dennett's intentional stance provides a framework for classifying our explanations as well as providing an instrumentalist account of the propositional attitudes. According to Dennett, our explanations fall into three categories depending upon the "stance" we take towards the phenomenon we are explaining. The ideas put forward by Dennett are closely related to the distinction found within the philosophy of mind, and psychology, between differing *levels of explanation*. It is possible to explain human cognition (and behaviour) at the level of neurology, or at the level of scientific psychology, or even at the intermediate level of cognitive neuroscience. Importantly for Dennett's account, the object in question does not need to *possess* internal states corresponding to the propositional attitudes in question, it merely has to be explicable in terms of such states. Famously, the Intentional Stance can be used to "explain" the activity of a thermostat (it *wants* to keep the room at 23 °C, it *believes* that changing a particular boiler setting will achieve this . . .), and at the

other end of the spectrum certain neurologists can be considered as applying the Physical Stance to human behaviour.

This thesis therefore shall adopt an instrumental account of folk psychology: the beliefs and goals of the agents within the car-park or other pedestrian scene may not actually exist, but even if they do not they have explanatory power and are useful shorthands for the underlying behavioural motivators, be they at the neurological level or otherwise. Given these assumptions it would seem eminently sensible to take these goals into account, and adopt the *Intentional Stance* towards the *agents* within the scene.

2.4 Navigational strategies

As this thesis attempts to model human intentional behaviour in a scene containing obstacles, it needs to take into consideration the way in which people actually navigate around a scene, taking into account the obstacles and their goals. A major criticism of Dennett's intentional stance is the assumption of rationality (see, for example, [142]). A consideration of the psychological literature will reveal whether our day-to-day path planning activity is rational (do we actually take the shortest or least-cost path?) and enable subsequent models to reflect more accurately the way people really plan their route.

The section provides some background to the question of navigational strategies, and as such includes literature from a range of disciplines. It starts with psychological approaches to path planning and concludes with a consideration of path planning and scene learning within robotics, a field driven more by bright ideas than by psychological plausibility, but interesting nonetheless.

2.4.1 Human path planning and spatial cognition

Investigations into human spatial cognition have looked into the way in which we represent spatial information to ourselves – our cognitive maps – and the way in which we plan paths (presumably using such maps) through an environment to our chosen destination or goal. Studies into what could be called the micro-planning (the way in which an agent plots an avoidance path around a specific obstacle) are rare, and the majority of work has been on navigating through larger scale scenes either in real or virtual environments. Related to the path planning literature is work upon perceptual distance. This is not the same as real-world distance, as our internal representations or calculations of distance in space are influenced by other factors, such as perceived effort, journey time or the number of features along the way.

2.4.1.1 Cognitive maps

Cognitive mapping has been studied extensively from the perspective of animal psychology, human psychology and robotics. The ability to form a representation of our environment, and the objects within that environment, and to then situate oneself within that representation seems to be a fundamental ability. Even insects have been shown to build fairly detailed representations of their environs (see [49] for an overview). Such representations incorporate landmarks (in the form of trees, buildings and so on), spatial information, and “global” features such as the position of the sun⁵. One tension in the cognitive mapping literature is between cognitive maps as collections of landmarks with spatial information stored as a secondary consideration, and with cognitive maps as spaces which happen to contain landmarks [160]. This distinction is further muddled by the habit of roboticists and others working within computational models of cognitive mapping to call everything that is perceived as an object “a landmark” (see Section 2.4.2).

Yeap and Jefferies, in [160], discuss *early* cognitive mapping (the way in which we build up a representation of a new area with a view to exploration) and as such they are interested in our representations of space rather than the way in which we move through that space towards any particular goal. They prefer a space-based approach, with obstacles and landmarks forming boundary points to, or being situated within, that space. This approach is contrasted with object based approaches containing limited distance information. There is some evidence that the space-based approach is more psychologically plausible, as researchers have recently [153] determined that space and distance information are relied upon more than other elements of a cognitive map such as inter-landmark angles in humans (unlike rats [9]).

2.4.1.2 Path planning and distance perception

The perception of distance is a complicated matter, and is of direct relevance to the way in which people navigate through a scene. Naïvely, people could be expected to take the shortest route from A to B but this is not always the case. Whilst route length is related to journey time (and straight line distance) [122], it is not the only component people take into account when choosing a route. Researchers commonly distinguish between *vista distance*, which is distance perceivable directly such as the distance from one side of a plaza to another; *pictorial distance*, which is distance perceived via a map or other pictorial representation; and *environmental distance*, which is distance we perceive by

⁵This is of particular interest to ethological researchers because it is not only global but also changes over time, as some rather confused honeybees flown from New York to California by Renner [116] confirmed.

interacting with and navigating through our environment. It is this last form of distance which is of interest here.

Golledge, in [51], carried out a two part study into route selection in a university campus. Part one of the study used map-based measures, and part two used the real campus. Some of the findings were replicated across both parts of the study, and it is these which are most interesting. Asymmetries in planning (i.e., coming back via a different route) were common in the map based study⁶, but were also fairly common in one of the real-world scenarios, in which 75% of subjects returned via a different route. In [52] Golledge ranks the different strategies used in path planning as shown in Figure 2.5.

1. Shortest distance
2. Least time
3. Fewest turns
4. Most scenic/aesthetic
5. First noticed
6. Longest leg first
7. Many curves
8. Many turns
9. Different from previous (novelty)
10. Shortest leg first

Figure 2.5: Ranking of strategies used in path planning, from [52]

Golledge’s study highlights the fact that human path planning is not as simple as just finding the shortest or quickest path from A to B (indeed, reported in the map-based part of Golledge’s study is the effect of “trip chaining”: if a trip is planned from A to B to C, the chosen route from A to B might be different to that which would be chosen were B the final destination). Golledge’s study has been followed by a number of investigations into the perception of environmental distance [10, 72, 73, 102].

The fact that Golledge found “fewest turns” or *simplest path* to be one of the most attractive metrics for path planning implies that either our distance perception is skewed

⁶Although the startling levels of asymmetry found with map based planning might be due to perspective effects.

by path complication, or there is some other reason for preferring simplest paths (for example, some form of cost is associated with changes in direction). The first of these options motivates the study of environmental distance.

Distances are perceived as being shorter or longer dependant upon whether they proceed away from or towards primary route nodes or reference points (Sadalla et al [119]). This is borne out by anecdotal accounts of people taking different routes from and to a particular place. Conroy-Dalton [29] calls this the British Library theory, after the place in which she observed the behaviour (see Figure 2.6 (a)). An informal survey within the School of Computing at Leeds showed that in the 15 people polled, asymmetry in planning is strikingly evident in the way people plan their route to and from Leeds City Station: Figure 2.6 (b) shows the results. This could be due, as Conroy-Dalton suggests, to a preference for the straightest path between A and B, or it could be due to wanting to make the first stretch of a journey the most significant (in terms of distance travelled towards final goal).

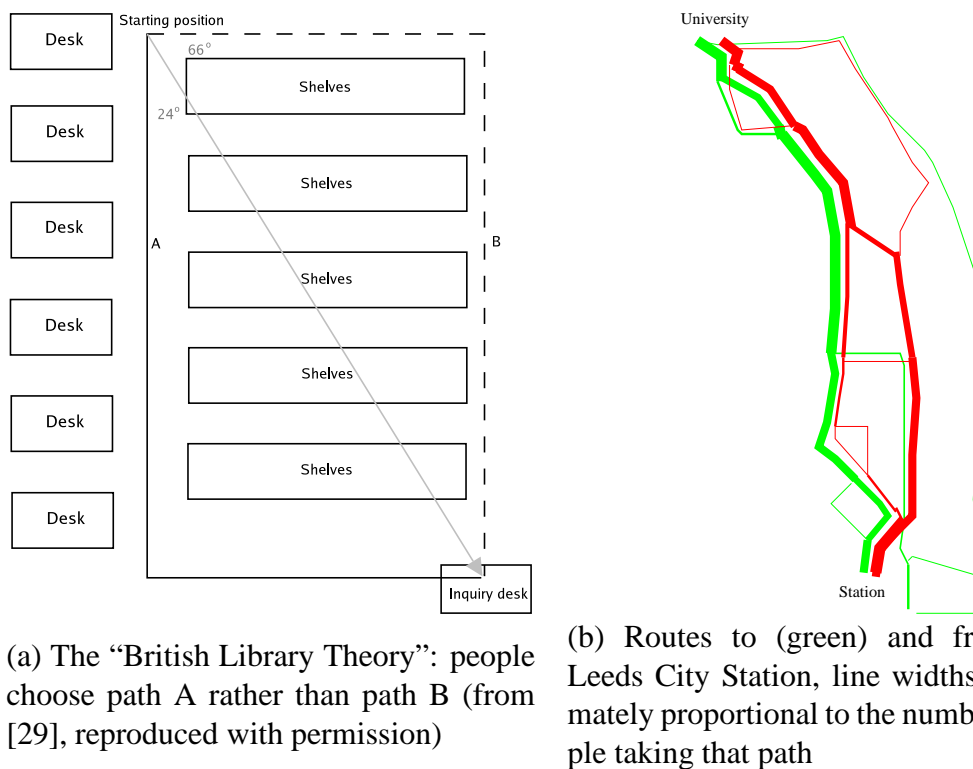


Figure 2.6: Asymmetry in path planning

It has been shown that the more landmarks or features there are along a route, the longer that route is perceived to be. Montello [102] calls this *feature accumulation*. It has also been shown that segmented routes are perceived as longer – a path containing seven right angled turns is perceived as being longer than a same length path containing

only two such turns [120]. Allen, in [1, 2] has shown that distance perception is affected by whether the distance being judged spans a boundary between two areas (or route *segments*). Indeed, Allen suggests that the subdivision of a route into segments seems to be a fundamental feature of our macro-spatial perception.

More recent studies in virtual environments only partly bear out such findings – in [73] an experiment is described in which subjects are shown routes of identical length but differing numbers of turns. When shown two routes, subjects perceived the one with more turns as being longer. When only shown one or the other of the routes, no difference in perceived distance was shown. Jansen Osman and Berendt have carried out a series of virtual environment studies [10, 72] in an attempt to disambiguate the effects of feature accumulation and path segmentation to determine whether junctions and turning points make a path appear longer through segmentation, or whether the perceptual lengthening effect is just due to feature accumulation.

Other possible reasons for people occasionally preferring simplest path over shortest is the limiting capacity of short-term memories, and the complications of giving directions. Indeed, choosing or determining simplest paths automatically for navigational software is proving an increasingly popular pursuit, given that shortest paths are often complicated in terms of directions [41]. Montello identifies a number of other possible complicated or confounding variables such as perceived effort, or attractiveness of route (preferences for parks over rubbish dumps have unsurprisingly been demonstrated).

2.4.1.3 Simulations

Within the field of transport studies, investigations into the interactions between pedestrians and their environment (and any obstacles within that environment) are common. These investigations take many forms: interviews, analysis of video footage, field observation and agent-based computer modelling all play a role. These simulations often center around the modelling of crowd behaviour (e.g., [143]) using cellular automata (e.g., [3]).

In [83, 155] an approach which combines video analysis of actual pedestrians and interviews in order to produce an agent-based computer model of behaviour is described. The types of question the model intends to answer are those of the form *What would happen if we were to place lamp-posts here, here, and here?*, and thus the first question to answer is *How do pedestrians behave when faced with an obstacle like a lamp-post?*. The agent-based system (called PEDFLOW) models the world as a grid occupied by obstacles (represented as occupied cells in the grid) and pedestrians as autonomous software agents (the size of one grid square) with limited knowledge of this world. Agents can move one square in the grid forwards, sideways or diagonally in each time step and their behaviour

is determined by rules and parameters. Rules are collections of logical statements such as *vacant lane on left AND blockage ahead THEN move to left*. Parameters affect the way individual pedestrian agents implement the rules, such as preferred distance from other pedestrians. Each agent has a goal, and no memory, so perceives the situation afresh each time step.

2.4.2 Path planning in robotics and other quests for the ideal path

Within the fields of robotics, scheduling, and navigational applications the aim is not to find the most psychologically plausible path through some representation but to find the most efficient: the shortest, or the quickest, or the path that mimimizes some other cost function. As the previous section has shown, this is not always the way humans approach the task of path planning. This section will start with a consideration of navigational aids, then move on to path planning and the representation of space within robotics.

Whilst many applications of path planning for satellite navigation or other navigational aids (such as Internet “driving directions” servers) can be thought of as finding the optimal path through space, in reality they are concerned with finding the optimal path through a network – a network of roads. The determination of the shortest path in a network is a well-understood problem [39]. However, this is not always the easiest path for human navigators to follow and whilst much research concentrates on features of the route description, without really considering features of the route itself (e.g. [34, 125, 144]), some recent work has concerned itself with qualities of the path.

Approaches to finding the optimum path where the terrain is more complicated (by including penalties for turning, or by modelling different surfaces, or by including some other cost term) have recently begun to use graph based methods. Early approaches, described in [5] have involved three steps: first the generation of a rasterized friction surface representing the variability of the terrain, then the labelling of cells in the rasterized representation with cumulative cost, and finally tracing backwards from the goal position to the current position. Stefanakis and Kavouras [141] have shown that a more efficient way of determining paths with other cost functions is to first represent the problem as a graph with weighted edges, and then to use graph based techniques for finding the lowest cost path through the graph.

Winter, in [156] presents an algorithm which favours paths that proceed in a straight line, by creating a representation of the network (a dual graph) which can incorporate information other than just distance in space. The two stage algorithm involves first constructing the dual graph and then applying Dijkstra’s algorithm to find the most cost ef-

fective route through it. Tested on a street network of some 2500 edges (representing part of Vienna), this software can be used for finding paths which favour simple routes, or straighter routes, or indeed any other cost function that can be implemented within a graph structure. Duckham and Kulik, in [41] have presented an algorithm serving a similar purpose: in their work they associate a cost or weighting with each pair of connected edges and do not take into account distance information at all. Results are presented for a number of paths through the road network of Bloomington, Indiana, and despite omitting all distance information from their calculations the simplest paths were on average only 16% longer than the corresponding shortest path. They state that the simplest paths their algorithm determines are also “cognitively plausible”.

As mentioned earlier, explicitly goal-directed behaviour in robot navigation is rare: localisation (in which a robot learns a place and then has to return to it) is covered widely, and scenarios in which a robot learns a model of its environment with a view to exploration (including obstacle avoidance) are much more common. Pictorial representations of location are used by Yagi et al [159] and also by Zheng [162], in which some form of panoramic view is captured by the robot and then compared using picture matching metrics to determine whether the robot is in the right place.

Borenstein and Koren [15] first proposed the use of occupancy grids for mobile robot navigation. In their system, virtual force fields are constructed in which obstacles exert a repulsive force and goals exert an attractive force. Summing over the “forces” acting upon the robot provides it with its new heading. The idea of occupancy grids was developed in their 1990 paper [16], in which the perceived presence or absence of obstacles causes the values of an occupancy grid to be incremented or decremented. This provides the mobile robot with a virtual landscape map in which following the valleys provides obstacle avoidance behaviour.

Within that subset of robotics dominated by vision scientists, the robot location problem has become known as SLAM, for Simultaneous Localisation And Mapping. SLAM systems couple the problem of correcting for the inevitable drift in the robot’s odometric sensors (the problem of localisation) to the problem of building a representation of the scene. There is a sense in which the pictorial systems mentioned earlier [159, 162] perform SLAM-type functions, in that the robot learns a representation of its current position. However, the pictorial representations are not really maps in any useful sense of the word.

Much work in SLAM uses non-visual sensors, such as Chong and Kleeman’s [25] work with SONAR, and Thrum and Fox’s work with laser range finders [147]. Davidson and Murray, in [32] describes an approach which uses active vision with a stereo head

to build a map of the robot's environment. There are two key features to their approach: Firstly, the initial selection of features or "landmarks" is fairly arbitrary. If the robot's set of features for a certain location falls below a threshold it looks for more, and if an expected feature is not detected a certain proportion of the time that feature is deleted. In this way the feature set is updated and over time comes to contain stable features. Secondly, the selection of which features to look for (and verify the location of) at any particular time is driven by uncertainty: the most uncertain feature is chosen for the head to fixate upon. This feature has the highest informational content. All SLAM work uses some variant on the occupancy grid approach to mapping the environment: landmarks are identified, and then placed in some global coordinate system (along with the location of the robot itself).

Non-metrical representations of space are used in Meng and Kak's NEURO-NAV [101] where relations between areas are instead represented topologically (*corridor1* is connected to *junction3*, for example). By combining two neural network based models called "Hallway follower" and "Landmark detector", the robot can navigate around the environment represented by its topological map. Hallway following and obstacle avoidance is handled in a completely different way by RoBEE [123], which takes optical flow measurements from divergent stereo inputs (one on each side of its "head"), and by maintaining the same perceived rate of travel at each side stays equidistant from the walls of a corridor.

Dealing specifically with route planning, the approach taken in the robotics literature is generally to leave as large a "clearance" between the navigating robot and any obstacles as possible (see, for example, [24]). Faltings and Pu [43] deal with a dynamic world in which the robot can move obstacles out of its way, and utilise an internal map which they claim to be akin to mental imagery. Fraichard and others [45,46,124] are more concerned with smooth paths for car-like robots, and consider both the shortest path problem and the problem of independently moving obstacles, thus emphasising the temporal element of their representations.

A more fine-grained obstacle avoidance behaviour is described by Brock and Khatib in [19] in which they outline their "Elastic Strips" framework. Any planned path for a robot can be thought of as defining a volume of space through which the robot would pass were they to follow that path. By modelling this volume as an *elastic* tunnel, which expands to fill the empty space but contracts where obstacles are present, they determine the area of space which the robot could move through whilst safely avoiding contact with obstacles.

2.5 Concluding remarks

This chapter has provided a background to four main areas. The practical aspects of CCTV and surveillance installations have shown the motivation for working in this area. Work within computer vision for automated visual surveillance has been described, showing that a good deal of research has been carried out into the statistical modelling of human behaviour and the modelling of geography, but little or no work has tried to approach the problem from the perspective of the psychology of the agents within the scene. Philosophical concerns with propositional attitude psychology have been outlined and shown to be avoidable in the current situation. Finally, an overview of the way in which people actually navigate through a scene and the way in which roboticists have modelled navigation and spatial representation has been presented. The aim of this thesis is to bring these different threads together and show that incorporating ideas from human navigation and a consideration of the possible goals of the people under surveillance can be useful in the business of behaviour modelling.

Existing work in computer vision for surveillance has either concentrated on the mechanics of tracking, or upon statistical techniques which rely upon large training sets of data for each particular scene. Scene based techniques have a number of drawbacks when it comes to real-world surveillance scenarios:

Rare paths Systems such as those described by Johnson [74] or Makris [93] take a large training dataset and derive from this a statistical model of the paths or routes people typically take through a scene. In constrained scenes with defined paths (and given a large enough dataset) these systems can then perform typicality detection by comparing a new trajectory to the model and calculating some distance measure. But in many real-world scenes, particularly those with large open spaces such as car-parks, people moving around the scene do not stick to paths, and a small number of people will take “unusual” short cuts. These short cuts can be perfectly reasonable for the scene – and perfectly goal-directed – but unless there are sufficient examples of each short cut in the training set these rare paths will be identified as problematic. They are, of course, atypical. But they are not the sort of thing an ideal surveillance system should single out as they are completely explicable.

Dynamic scenes Scenes in which the goals and/or the obstacles move around provide another problem for systems tied too closely to geography. This is a particular problem in car-parks, although other pedestrian scenes have similar issues (imagine a train station concourse, with people clustered around luggage waiting for their

platform to be announced: the flow of people around the scene is predictable to and explicable by a human observer, but as the waiting people come and go the patterns of movement change).

Moving cameras In real-world surveillance installations, the cameras are often PTZ (pan-tilt-zoom) cameras that enable operatives to direct their attention to different areas of the scene. With such cameras, it would be necessary not only to perform registration to work out where exactly the camera was pointing, but sufficient training data would need to be collected from the camera pointing in every possible direction. This data collection problem would be extremely hard. A less serious problem for geographically rooted systems is that of slight drift in camera orientation with “static” cameras. Anecdotally, cleaning staff have been known to nudge static cameras a few millimetres in one direction or another, and this can cause problems with systems incapable of minor recalibration.

Other surveillance related systems have relied upon hand-crafted models of the types of behaviour that are to be detected, such as approaching a number of cars in turn [103] or rapid head movements [71], or by modelling the interactions between agents (for example [20] using a Bayesian context). These systems pay little or no attention to the structure of the scene and instead concentrate on the agents within the scene and their patterns of behaviour and interaction.

This thesis aims to tread a path between these two broad families of approach, neither relying solely upon geographical information or upon (sometimes hand-crafted) models of patterns of behaviour. Using a model of human navigation inspired by work in psychology and philosophy, coupled to a simple model of the scene and positional tracker output, it is hoped that the problems of earlier systems can be avoided. Rare yet still goal-directed paths will be treated appropriately; changing goal locations are easy to incorporate; and whilst the systems described in this thesis are not implemented in a multi-camera PTZ context, it could be adapted to work with any tracking system capable of providing agent locations in some coordinate system (with an appropriate scene model).

Chapter 3

Tracking and scene modelling

3.1 Introduction

To devise an intentional account of the behaviour of agents within a scene, certain things about the geography of the scene have to be modelled. It is necessary to determine which areas are potential goals, and which areas can be thought of as obstacles. These areas usually correspond to objects (hedges, walls, buildings etc.) in the case of obstacles, and doors, roads, or other ways out of the scene in the case of geographical goals. Also required is knowledge of the location and direction of motion of the moving elements of the scene - people and cars¹. These are the intentional agents under investigation.

This chapter details the tracking and modelling of the elements within the scene in order to support such an intentional investigation. Firstly, it will consider the choice of scene. There is then a consideration of tracking: how to find out where the agents moving around the scene actually are. It then goes on to discuss the geography of the scene: the question of how to detect and model those geographical features which will affect the behaviour of the agents (exits and obstacles). Finally, there is a brief consideration of “object permanence”: how to determine whether an agent who has stopped has left the scene or is still present.

¹Moving cars are treated as agents in this work.

3.1.1 Experimental data

The initial scene selected for the development of ideas using intentionality in behaviour modelling is a car-park at the University of Leeds. The scene has several important characteristics. It is well-lit, visible from above, and covers a large area of ground (approximately 300m from nearground to farground). It is free from major sources of occlusion - although trees and buildings are part of the scene, they do not obstruct the view of the car-park in any significant manner. These features allow the capturing of video footage of a large scene area using a single static camera. Over time, the scene also includes a number of large moving objects (cars). As these cars park (or drive away) they provide a changing layout of goals.



Figure 3.1: The Leeds car-park (hereafter “the car-park scene”)

An hour’s footage of this car-park was captured early in the day. The footage features the car-park between 9am and 10am and has a good number of pedestrians and moving vehicles. A standard commercial digital video camera was used for the filming, and the footage was then sub-sampled spatially and converted into Quicktime for storage purposes. The resultant video is 352 by 288 pixels and sampled at 15 frames per second. This scene is pictured in Figure 3.1.

The second scene used is from the ECCV-PETS2004 (European Conference on Computer Vision - Performance Evaluation of Tracking and Surveillance workshop) dataset². The PETS series of workshops make available public datasets for the comparison of tracking and surveillance technologies. The PETS datasets are provided with “*ground truth*” information, about the actions, locations and behaviours of the actors contained within.

This particular workshop (ECCV-PETS2004, hereafter PETS2004) provided a number of short videos in MPEG-2 format, each featuring a particular type of behaviour -

²This dataset comes from from the EC Funded CAVIAR project/IST 2001 37540

some with interactions between actors (e.g., meet and talk) and some with individual behaviours (e.g. simply walking across the scene). The activity was filmed in the foyer at INRIA Rhone-Alpes, in France. The scene contains a small number of static obstacles, and all behaviour is performed by actors. These videos were provided alongside information about the position of each agent, thus providing a second test scene for the current project without the need for tracking. Due to the short duration of these clips and the small number of actors present (23 individual behaviour patterns in all), the exit modelling techniques discussed in this chapter were not applicable to this dataset. Hence the PETS2004 dataset is discussed at length in the following chapters as a useful second scene for testing high level behaviour-related algorithms, but is only briefly described here as lower level techniques for scene analysis and object tracking were not required or appropriate.



Figure 3.2: The PETS2004 scene

3.2 Tracking the moving objects

Moving objects were initially located using Magee's object tracker [86]. This makes use of a variant on the Stauffer and Grimson GMM-based adaptive colour models [140] for both foreground and background, and also incorporates a shape model for vehicle tracking. This tracker is efficient in situations where the objects are relatively large, and move in predictable fashions. In the current application, we wish to track both people and cars across a scene which covers a large area. The tracker does not cope as well with this variety of scene and target - a pedestrian at the top of the image (in the far distance) is only two or three pixels tall, which provides insufficient information about colour for the foreground mixture models to stabilise. In addition to this, there are a number of moving objects within the scene which are neither pedestrians nor cars (trees, hedges, pigeons etc.).

Ideally, each blob reported by the tracker would correspond to one and only one object within the scene. This is not the case, and five different blob-object mappings are found in the raw tracker output.

1. The blob which is attached to one object sometimes becomes associated with another (for example, when a lorry passes in front of a pedestrian). This results in a one-to-many blob-object mapping over time.
2. Large objects are sometimes tracked with more than one blob, causing a many-to-one blob-object mapping in space.
3. Objects get “lost” by one blob, and then picked up by another. This results in a many-to-one blob-object mapping over time.
4. There is a large amount of tracker error - caused by objects stopping and their associated blob continuing to exist (this is made more likely by the high resolution required to track pedestrians as well as cars); by noise and by camera shake. Thus, there are some blobs without associated objects.
5. Some objects are missed completely, for a variety of reasons. This means there are some objects without associated blobs.

The tracking phase of this project has therefore been two-stage. The Magee object tracker was used to gain initial estimates of object position, and the resultant output file has been hand-edited to ensure that in all cases each blob represents one and only one object. This hand-tracking procedure involved investigating each trajectory and correcting the reported position of the object centroid in cases where error had arisen. All trajectories were investigated, resulting in a situation where some objects have been entirely tracked by hand, some partially by hand and most completely tracked automatically. Approximately 20% of object trajectories were altered in some way.

There are some unfortunate side effects of this two-stage procedure. Firstly, due to the labour intensive nature of the hand-tracking process, only the object centroid position was recorded. Thus hand-tracked objects lack the height and width estimates provided by the Magee tracker. This also forces the choice of object centroid (as opposed to base-point or top) as the point chosen to represent the location of the object. Secondly, there are differences of accuracy and smoothness in the trajectories of hand tracked and mechanically tracked objects. Nevertheless, the output of this process is trajectory (x, y, t) information on each moving object, with each moving object within the scene corresponding to one and only one object in the output of the tracking process.

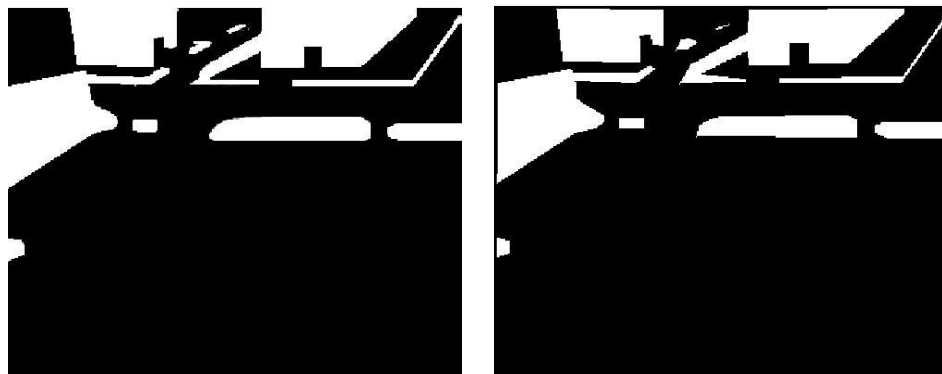
These data points are then Kalman smoothed [80] for two purposes - firstly to reduce noise (in the hand-tracked points in particular). Secondly, as this thesis is investigating intentional behaviour, some indication of what *direction* the objects are heading in is vital. The Kalman filter's velocity vector is a reliable indication of each object's current direction. The Kalman smoothing thus updates the x and y position of the object, and also stores a velocity vector at each time point. Using the velocity component of the Kalman filter as an indication of the objects current direction has a desirable side effect: the Kalman filter contains an uncertainty estimate, and in the absence of input (for example, when an agent stops moving), the directional uncertainty grows causing the direction component to vary a great deal. As there are no points within the scenes under consideration at which agents are expected to stop (benches, cash machines etc.), this noise serves to provide a means of penalising trajectories where the agent is stationary for any length of time.

3.3 Obstacles

The obstacle models have been hand-crafted, taking into account the location of obstacles within the scene and also the movement of the people within the scene. Those areas of the scene which agents cannot cross are marked as such. This model is created in the image plane rather than the ground plane for simplicity's sake. The decision to work in the image plane is not without complications: when an object trajectory crosses an obstacle this can be for one of two reasons - it can be due to noise in the tracking process, or it can be due to the object passing in front of the obstacle. The second case is much more likely with tall objects such as vans, where the object centroid is quite likely to come between the face of an obstacle and the camera.

In the initial formulation, the obstacle model took the form of a bitmapped representation with areas of the scene marked out on a pixel by pixel basis, and is shown in Figure 3.3 (a). This led to certain problems with the granularity of the model, especially around the edges of obstacles. When calculating the area of scene visible from a certain location, small variations in position can cause large variations in visible area (due to the edges of the obstacles being quantized).

To circumvent this, an obstacle model based upon a polygonal representation of the scene has also been developed. The polygonal obstacle model is pictured in Figure 3.3 (b), and is derived by straight line approximation from the bitmapped representation using the algorithm detailed in [87]. This algorithm first finds maxima of curvature, and uses these points as a first estimate of straight line approximation. Points are then iteratively removed



(a) Hand Crafted

(b) Polygonal



(c) Car-park scene

Figure 3.3: Obstacle models for the car-park dataset shown alongside scene: features from the scene (e.g. the hedge) are clearly visible in the obstacle model

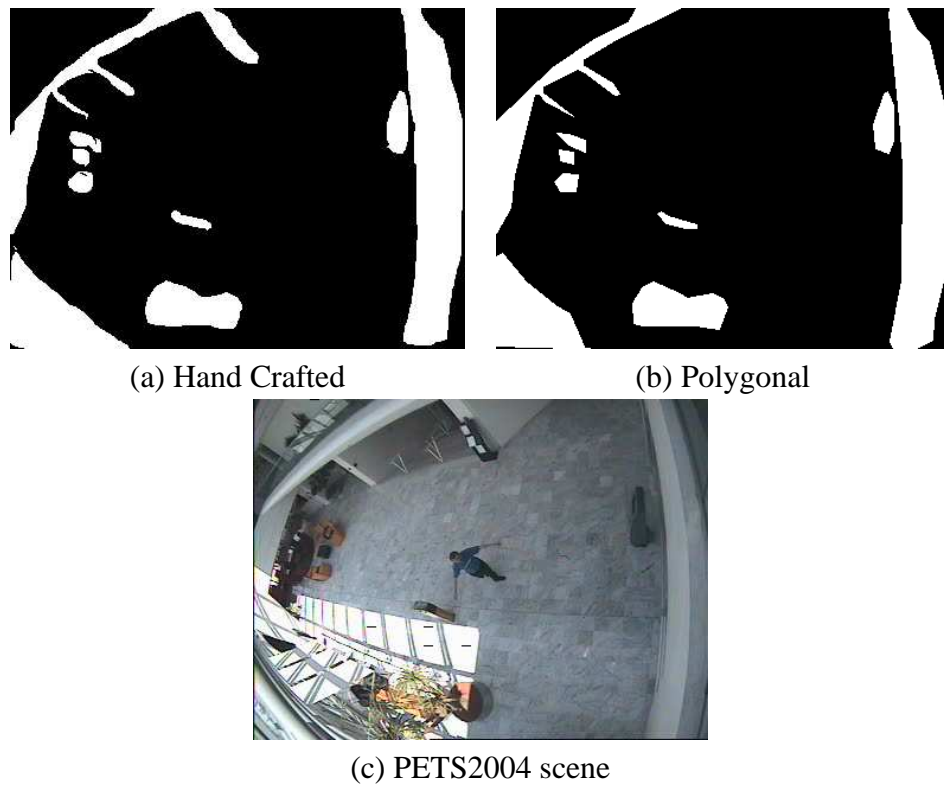


Figure 3.4: Obstacle models for the PETS2004 dataset shown alongside scene

and the remaining points' locations are adjusted to ensure a locally optimal fit³. Figure 3.4 shows the bitmapped and polygonal obstacle models for the PETS2004 dataset.

3.4 Exits

The assumption is made in this work that in pedestrian scenes, people have geographical goals. These goals could be a car, or a doorway, or a cash machine, or the entrance to a particular shop. The salient features of these goals are that they are geographically located and extended in space.

Considering the car-park scene detailed earlier, a person has one of two goals: either to *find a specific exit* or to *find a specific car*. The situation is simplified further in the PETS2004 foyer scene as there are no cars, and thus we need only consider exits. This gives two levels of goal which have to be modelled: cars, and exits. Exits can be located by examining the end points of trajectories as in [93,99,138]. Assuming that each entrance

³The initial straight line approximation thus created does not touch the edge of the scene, which can cause problems when calculating paths: impossible paths are postulated around the back or across the top of obstacles. The model has been adapted so that obstacles touch the edge of the scene where required.

can also be an exit enables the start points of trajectories to contribute in the same way, doubling the size of the data-set of exit points. A more detailed analysis may be required in scenes which feature, for example, one way streets, but within the current scenes all exits can be treated as bidirectional. Trivially, cars can be located by assuming that each time a car trajectory ends a car can be found (if not at an exit).

The set of trajectory finish points can therefore be considered to be a set containing the location of all the cars within the scene and the location of the exits from the scene. Trajectory start points similarly contain places where people have entered the scene (entrances, which in the current scene are also exits) and cars (when a car-parks, and its passenger(s) emerge, the pedestrian trajectories begin at the car). Borrowing terminology from Ellis & Xu [158] trajectories can be expected to end in a number of different types of situation, or “occlusion”. Border occlusions are where people leave the scene at the edge. Long term occlusions are where people leave the scene in the middle (by entering a building through a door, or by walking behind a building or wall which abuts the edge of the scene). Short term occlusions occur when people walk behind objects like trees or walls and then reappear.

For the sake of simplicity all three cases can be treated identically (and no attempt will be made to unify trajectories in situations with short term occlusions). This is not ideal, as in some instances agents move behind occlusions and then emerge again, and the system described here makes no attempt at unifying these trajectories. With a single fixed camera, it is not possible to see behind objects and determine the actual ground-plane position of all agents within the scene, so positional knowledge around occlusions is not available. The introduction of statistical occlusion reasoning such as that developed in [158] would provide an element of continuity in those cases where an agent’s trajectory has been split, however it would also add a layer of complication to the reasoning. Thus split trajectories are not re-unified around short-term occlusions. This simplification falls into the same category as that of working in the image plane detailed in Section 3.3: given a multi-camera system it would be possible to circumvent these problems, but it is beyond the scope of the current thesis.

The sets of start and end points to the trajectories from the car-park scene are depicted in Figure 3.5. Comparing this set of points with the scene, certain features are clear: a door, the edge of a building, the edge of a hedge. These are all occlusions of some type. It is also clear from Figure 3.5 that even after processing the dataset is not free of noise.

The image in Figure 3.6 shows the exit points classified by hand into three categories, corresponding to those identified by Ellis and Xu [158].

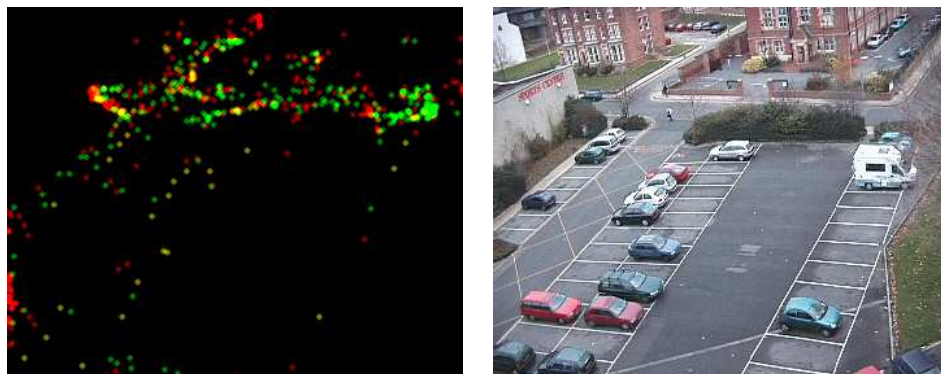


Figure 3.5: Trajectory start points (green) and end points (red), shown alongside the original scene for comparison. Points which appear yellow are those where green and red overlap.

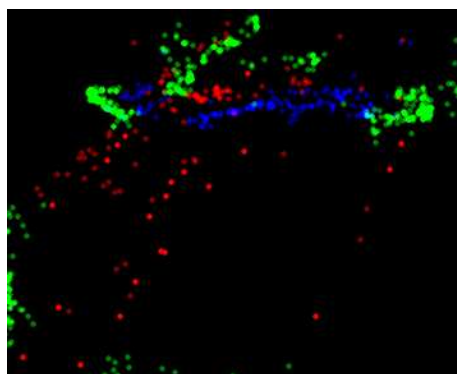


Figure 3.6: Hand labelled ground truth: Long term occlusions (green), short term occlusions (blue) and noise (red). Features of the scene, such as the hedge in the middle and the popular exits, are discernible from these data points.

3.4.1 Finding and representing the exits with a mixture of Gaussians

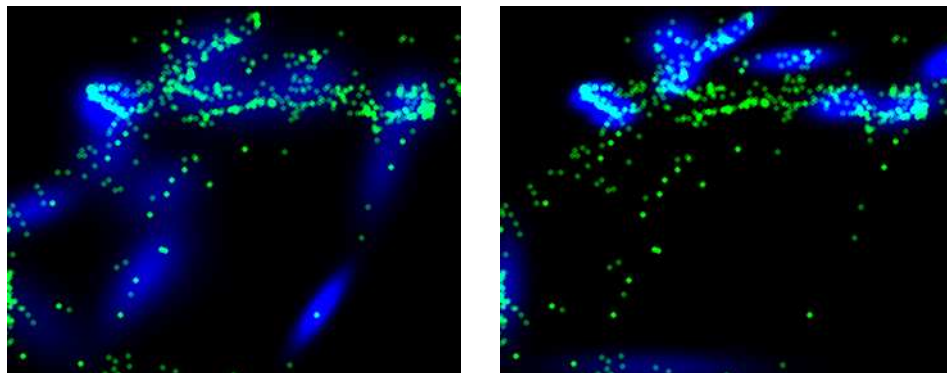
The exits (border occlusions, doors and so on) have spatial extension, and so are best modelled by a representation which also has spatial extension. As in [98, 138] a Gaussian mixture is used to model the exits in the scene. The models are trained using Cootes and Taylor's adaptive kernel version of the Expectation-Maximisation (E-M) algorithm [30] which combines two methods for creating a Gaussian mixture model. The first of these (the adaptive kernel method) uses one Gaussian for each data point, and allows the scale of the kernels (or Gaussians) to differ to accommodate differing densities of data points. As it uses one Gaussian per data point, the adaptive kernel method is computationally expensive. The second is the E-M algorithm, first introduced in [33], which enables the modelling of a large number of data points with a small number of Gaussians, by iteratively computing the contribution of each data sample to each Gaussian then recomputing the Gaussian parameters. The E-M algorithm is sensitive to initialisation and can sometimes result in Gaussians which represent just one data point (over-fitting). Cootes and Taylor's adaptive E-M algorithm (shown in Figure 3.7) alters the M step of the E-M algorithm to incorporate information from the adaptive kernel method, resulting in an algorithm in which singularities do not occur, and which is less sensitive to initialisation than the original E-M method. The difference between this and the original E-M method of Dempster et al. [33] is the addition of the term T_i representing the kernel covariance of a sample calculated using the adaptive kernel method (shown in Figure 3.7 in boldface).

<p>E-step: Compute the contribution of the i_{th} sample to the j_{th} Gaussian.</p> $p_{ij} = \frac{w_j N(x_i; \mu_j, \sigma_j^2)}{\sum_{k=1}^M w_k N(x_i; \mu_k, \sigma_k^2)}$ <p>M-step: Compute the parameters of the Gaussians.</p> $w_j = \frac{1}{n} \sum_{i=1}^n p_{ij}$ $\mu_j = \frac{\sum_{i=1}^n p_{ij} x_i}{\sum_{i=1}^n p_{ij}}$ $\sigma_j^2 = \frac{\sum_{i=1}^n p_{ij} [(x_i - \mu_j)(x_i - \mu_j)^T + \mathbf{T}_i]}{\sum_{i=1}^n p_{ij}}$

Figure 3.7: Cootes and Taylor's altered EM algorithm to fit a mixture of m Gaussians to n samples x_i

This technique provides location estimates which represent the distribution of trajectory start and finish points within the scene, and which also represent the spatial extent of those clusters of points which make up the basis for this location estimate. Such models are still sensitive to initialisation and experiments were conducted to determine the best initial values to use. The K-means algorithm was found to be more reliable than seeding the mixture with random points, or random points from the dataset, yet has the advantage of being computationally inexpensive.

Figure 3.8 (a) depicts mixture models trained on the complete set of trajectory start and end points from the car-park dataset. These models were initialised using values derived from an application of K-means. The mixture model thus trained has a few drawbacks – there is a Gaussian centered over a row of cars, for example. There are several ways these problems could be circumvented. Applying the mixture based approach outlined above to the long term occlusion points only, as expected, provides a much clearer picture of where the exits in the scene lie. Figure 3.8 (b) illustrates the output of this approach.

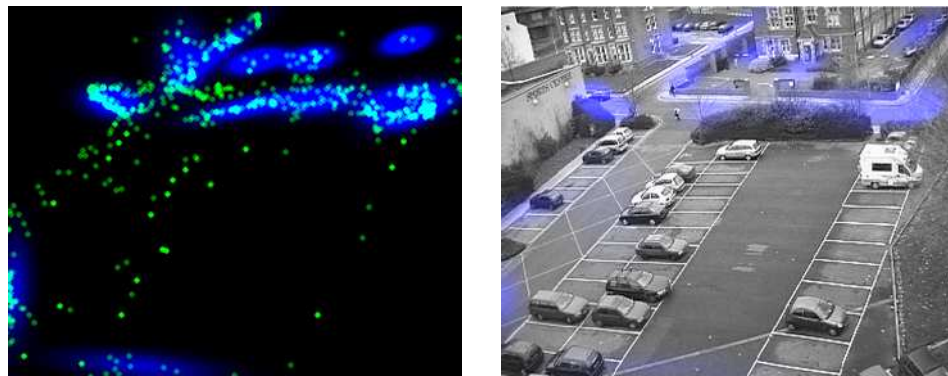


(a) Gaussian mixture model trained upon raw start and end points. (b) Mixture model trained upon long term occlusion points only

Figure 3.8: Mixture models trained upon raw points and selected points

Figure 3.9 shows both long and short term occlusion data points modelled as a GMM. This shows (as the images in Figure 3.8 suggest) that selecting just those points which represent real entrances or exits to the scene results in a reasonable exit model for the car-park scene.

It would be desirable to develop an automatic way of distinguishing between those members of the training set which are due to noise, and those members of the training set which are due to people actually leaving the scene. The noise points are due to temporally transient events, such as cars parking, or vans unloading. Some of these events could be expected to be more compact in the time domain: for example, a van stops (end point),



(a) Both long and short term occlusion points, but no “noise”. (b) Exit model superimposed on scene

Figure 3.9: Mixture model trained upon selected data points

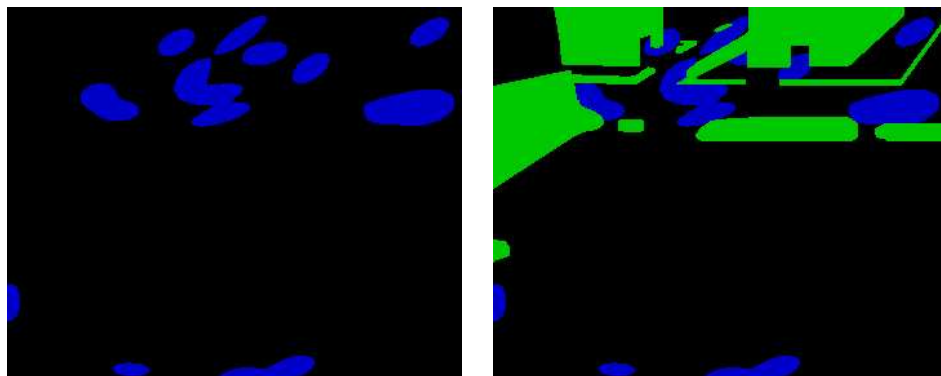
two people get out of it (two start points), unload something, get back in the van (two end points) then drive off (one start point). This sequence of events results in six noise-related data points which are close together in both space and time. Given this expected characteristic of noise data points, entropy related measures were considered as a means of distinguishing between noise (low entropy) and real exits (high entropy). However, coincidentally, one of the actual exits has fewer than six data points associated with it, and the “real exit” data points are only slightly more spread out in the time domain than those associated with the van sequence.

Noise points might also be expected to be spread out in the the spatial domain: for example, a large number of disparate data points correspond to the cars parked in the car-park. These might be expected to contribute to a mean (a Gaussian, or a K-means cluster) which has a large spatial variance. There is a different problem with using spatial variance as a criterion for distinguishing real exit points from noise, which is illustrated in the car-park scene by the large hedge running across the middle of the scene. Without performing some form of occlusion reasoning, the hedge has to be considered as one or more large exits, spread out over space - i.e., with a large spatial variance.

It has been shown by others e.g. [92,99,138] that given sufficient quantities of training data one can model exits reliably using a GMM trained upon trajectory start and end points. However, given just an hour’s footage the problem of distinguishing actual exits from noise is a problem, for the reasons highlighted in the preceding paragraphs. Over time, however, it is clear that the number of trajectory start and end points associated with exits would greatly outnumber those associated with transient events such as cars parking or vans unloading. Indeed, over the course of several weeks, the exits to the scene would

become obvious. It might even be the case that Gaussian components with small variance and low weight appear over the parking spaces.

What is required is an indication of location and spatial extent for geographical goals, and to obtain this in the absence of a large enough dataset of trajectories, output from the mixture based approach using hand classified occlusion data points has been used. For computationally expedient reasons each exit is modelled as a solid ellipse defined by those points one standard deviation or less from the mean of that Gaussian. The Gaussian based exit model is shown in Figure 3.10(a), and the exit and obstacle models combined are shown in Figure 3.10(b). Taking points one standard deviation from the mean of each Gaussian component as defining the exit means that some trajectory start and end points fall outside the “exit”.



(a) Gaussian exit model thresholded at 1 standard deviation from each component mean

(b) Exit and obstacle model combined

Figure 3.10: Mixture model of exits illustrated alongside obstacles

3.4.2 Some thoughts upon exits and goal-directedness

The aim of this thesis is to model goal-directed behaviour, and as such, the locations of the goals would at first thought appear to be vital. However, one hypothesis that will be investigated in Chapter 7 is that such a model may be unnecessary: if the conception of goal-directed behaviour is flexible enough, perhaps it will be possible to determine a measure of intentionality without actually knowing where the goals are. Clearly this approach loses some explanatory power, as it is no longer possible to say *Agent n is going towards goal m*, but nonetheless it might be possible to say *Agent n is moving through the scene in a goal-directed fashion*, and maybe even *Agent n is moving through the scene*

towards a goal located in the bottom left.

A “perfect” exit model has also been constructed for the car-park scene by hand. This consists of rectangles (rather than ellipses) and makes use of knowledge of the scene to specify where all of the doors, exits and occlusions actually are. The purpose of this model is to provide a benchmark against which the learned exit model can be compared. This hand-crafted exit model is shown in Figure 3.11. The exit model for the PETS2004 scene was hand-crafted in the same way, and is shown in Figure 3.12.



Figure 3.11: Hand-crafted exit model and scene: car-park dataset

3.5 Object permanence

One question which needs to be addressed and one for which the presence of a spatially extended model of the exits is vital is that of object permanence. In the car-park scenario, the end of an agent’s trajectory can signify one of three things: the agent has left the scene, the agent has passed behind an occlusion, or that a car has parked.



Figure 3.12: Hand-crafted exit model and scene: PETS2004 dataset

Parked cars are incorporated into the scene model as valid goals for the agent. The approach adopted in this thesis is a simple one: If a trajectory ends within one standard deviation of one of the mixture model components, it is assumed to belong to an agent who has left the scene. If not, it is assumed to belong to a car which has parked. The start of a trajectory has a similar array of possible meanings: either an agent has entered the scene, or appeared from behind an occlusion, or a car has started up and is about to drive away. As for the driving off of parked cars, it is assumed that if a trajectory starts within a certain distance from a place where a car is assumed to be, the goal corresponding to that car is removed and it is assumed to have “driven off”. Effectively this means that there is a circle around each parked car, and if a trajectory starts within that circle the car is assumed to have driven off.

This does not make the correct assumptions about object permanence in all cases, as there are some exits which have been missed by the exit model and some exits whose extent has been underestimated. However in the case of those “missed exits”, this sub-optimal approach has a lucky side effect: a (non-spatially extended) goal is placed at the point at which the trajectory ends. This leads over time to an accrual of “goals” at the missed exits. Within the current implementations, parked cars become incorporated into the goal model as possible targets, but not into the obstacle model. If the system were to be extended to incorporate cars-as-obstacles as well as cars-as-goals, the object permanence solution described here would obviously break down.

Chapter 4

Building an agent-centered representation of the scene

The previous chapter discussed the construction of models representing the goals and obstacles within the scene, and the tracking of the agents - the determination of the location of people and cars over time. This chapter deals with the interaction between these different types of things and presents the basis of a model of the way in which an agent navigates around obstacles towards one of the goals. To do this, the geography of the scene is characterised in such a way as to take into account the position and motion of the agent creating an agent centered representation, which can be thought of as a map of possible intentions.

Analysis starts with the determination of the area of scene directly visible to the agent. This is bounded by parts of the edges of the scene and parts of the edges of visible obstacles, with lines of sight bounding the visible area where an edge or vertex of an obstacle is found. The area thus defined is similar to the Absolute Space Representation (ASR) of Yeap and Jefferies [160], in that it represents the area of the scene that the agent (or robot, in their case) has visual knowledge about.

4.1 Sub-goals

In a scene without obstacles, it is possible to determine which goals are consistent with the movement of an agent by working out whether the agent is moving towards, or away from, each goal. With obstacles, the problem becomes more complicated. The agents' actual goal may be obscured, requiring the agent first to move away from the goal in question in order to circumnavigate some intervening obstacle. Or it may be that the agent cannot get to that goal from where they currently are: the route to the goal might involve leaving the scene. In order to account for this behaviour, this thesis proposes the use of virtual "*sub-goals*", which are defined as points in the scene where an agent might choose to change direction.

Such sub-goals are central to this approach. They allow people to go around corners – if there is not a direct path to a particular goal from the current location, that does not mean that goal is not a possible explanation for the behaviour in question: there may exist an interim position with a direct path *to* the goal and *from* the current position. Such a position becomes a sub-goal.

This analysis is based upon the hypothesis that in general, the path an agent takes is a series of straight line segments through free-space, terminating at tangential points on the obstacles, and connected by curved segments around the boundaries of obstacles (in environments where the boundaries of obstacles are curved). For a scene with only polygonal obstacles, all segments are straight and the turning points are tangential vertices of obstacles. This hypothesis will be considered in detail in Chapter 5.

The construction of sub-goals is based upon geographical information about the location of obstacles, the current location of the agent within the scene x and their direction of motion θ , and upon counterfactual reasoning. From the current position x , a segment of the scene is investigated. The aim is to discover places to which the agent could travel directly and that allow the agent access to places to which they cannot travel directly. The concept of a sub-goal and the way in which sub-goals change over time is illustrated in Figure 4.1.

From Chapter 3 we know the position of the agent x , their direction of motion θ and the location of the exits and obstacles. The algorithm for determining the location of sub-goals and subsequently labelling areas of scene from this information is different for bitmapped and for polygonal obstacle models. Indeed, as the algorithms determine what is visible based upon information from the obstacle model, the bitmapped model results in a bitmapped scene and the polygonal model results in a scene represented as a collection of polygons. The following sections consider each type of scene in turn.



When the agent (circled in red with direction of travel indicated by an arrow) enters the scene, it is unclear which exit he will use. The agent is headed away from all of the exits from the scene which are within his line of sight (those goals circled in blue at the bottom of the scene). Possible sub-goals are circled in green - in order to reach areas out of sight, he would need to pass by a sub-goal.

As the agent progresses through the scene, some goals become less likely explanations for the agents' behaviour, and some sub-goals disappear as he is no longer heading towards them.

This process continues, and more sub-goals disappear.

Finally, as the agent is near his final goal, there are just two sub-goals active and only a few possible exits these sub-goals might lead to.

○ Sub-goal ○ Agent ○ Goal

Figure 4.1: How sub-goals change over time

4.1.1 Determination of candidate sub-goals within a bitmapped representation

The determination of sub-goals in a bitmapped scene involves scanning the scene from x looking for regions which fit certain criteria. Initially, pixels are labelled as either being directly visible from x (labelled **V**), obstacle (labelled **O**), or not visible from x (labelled **N**). Given this classification, the next stage is to look for possible sub-goals in the direction of the agent’s travel, allowing one radian either way for deviation from the straight line path. These boundaries correspond roughly to our maximum angle of vision, and are motivated by the assumption that we look where we are going. Thus those pixels that are classified as **V** and which lie within an arc through x from $\theta - 1$ to $\theta + 1$ are investigated further, searching for pixel neighbourhoods containing all three labels of pixels. This is achieved by passing a square window (5 pixels by 5) over the image and determining how many different pixel labels are in that window. Regions containing all three types of pixel label are candidate sub-goals – that is, the agent at x might be headed towards x' (it is directly visible and within their angle of vision), x' is next to an obstacle, and were they at x' they would be able to see more of the scene (it neighbours upon areas that are not directly visible from x). A constraint is included to stop rows of sub-goals being constructed along the edge of obstacles. This is due to the saw-toothed nature of diagonal bitmapped edges, which “hide” pixels from view that would otherwise be marked as **V**.

When a candidate sub-goal has been found, scanning starts from x' in all directions, pixels are labelled and further sub-goals are searched for in a similar manner. Pixels directly visible from x' but not from x are labelled as **S1** (visible from 1 sub-goal) and pixels directly visible from any newly discovered sub-goals (but not more directly) as **S2**, enabling analysis of which actual goals are accessible one sub-goal, and which actual goals would require the agent to pass through two sub-goals. These stages are illustrated in Figure 4.2. In Figure 4.2 the agent is represented by a red circle with an arrow corresponding to their velocity vector. Sub-goals are shown as green circles, the obstacle model is shown in black, and areas which are not visible (either directly or via a sub-goal or two) in white. Pale blue areas are directly visible, but the agent is headed away from them, and pink areas are those directly visible to the agent and within their $\theta \pm 1$ field of view. Areas shaded light yellow represent areas visible via one sub-goal and dark yellow by two sub-goals. The bitmapped implementation described here stops the sub-goal analysis at two levels of sub-goal (“level 2” sub-goals) for computational reasons, although such an analysis could in principle be continued to an arbitrary depth.

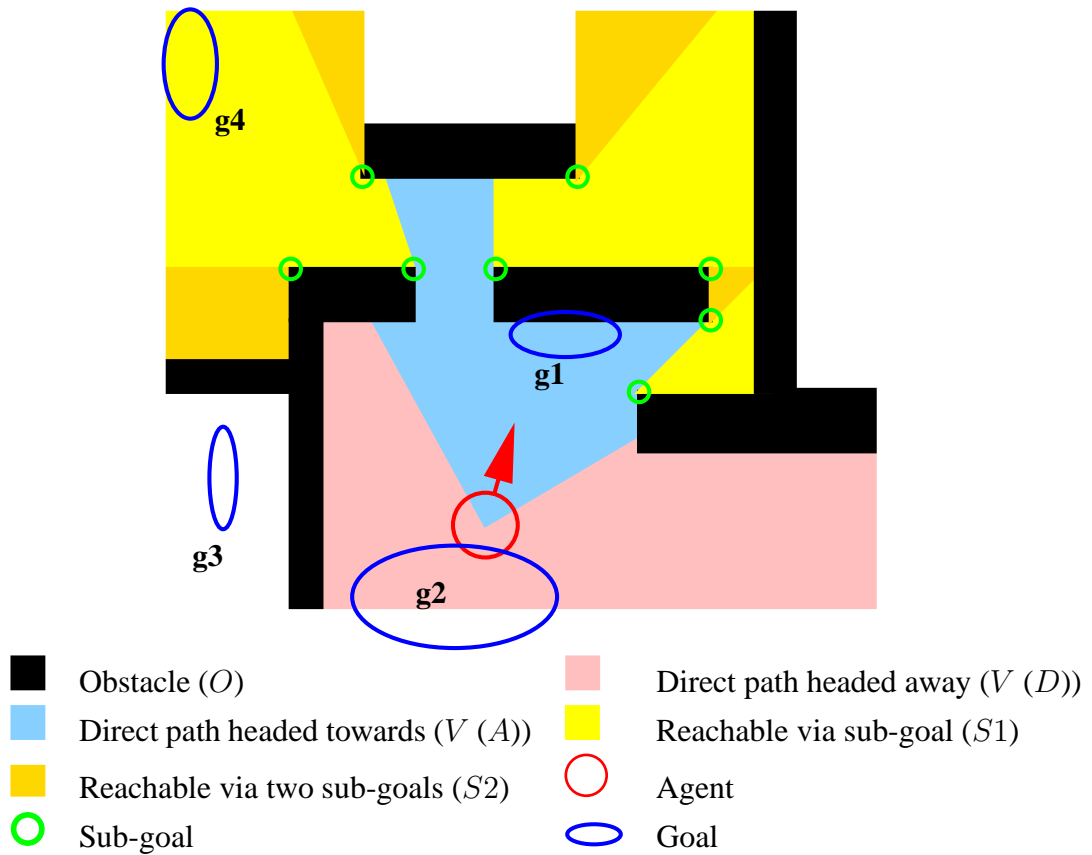


Figure 4.2: An illustration of the sub-goal algorithm in action

4.1.2 Determination of candidate sub-goals within a polygonal representation

With a polygonal obstacle map, the algorithm for determining sub-goals is considerably simpler. For each obstacle within the scene, consider each of its vertices v_i in turn taking a line from x through that vertex. If the neighbouring vertices (v_{i+1} and v_{i-1}) are both on the same side of the line through x and v , then v is a tangential vertex on that obstacle (as with the bitmapped representation, the line between x and v must fall within one radian of direction of the agents' velocity vector). In order for a tangential vertex to be a potential sub-goal, it must be visible from x : that is, the line from x to v must not pass through any other obstacles. Visible tangential vertices are, by definition, sub-goals. This is illustrated in Figure 4.3, in which visible vertex A is a sub-goal as both of its neighbouring vertices are on the same side of the line through the agent and the vertex. Vertex B is not, as the neighbouring points are on either side of the line, and vertex C is not as it is obscured by an obstacle.

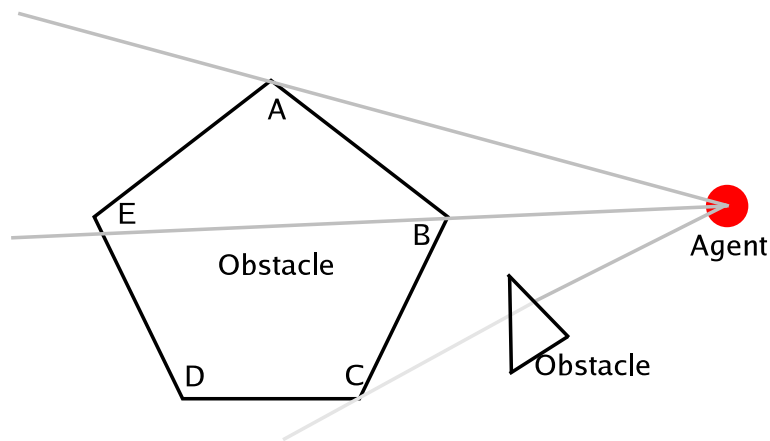


Figure 4.3: The determination of candidate sub-goals with a polygonal representation. Vertex A is a sub-goal, as the vertices to either side of it lie on the same side of the line from agent through A. Vertex B is not, as the vertices to either side on the obstacle lie on different sides of the line from agent through B. Vertex C is not, as it is obscured from view by the smaller obstacle.

Further sub-goals can be discovered in an analogous fashion simply by repeating the process with the location of the sub-goal in the place of x , and the polygon representing the area already visible treated as another, virtual obstacle, as paths to further sub-goals should not cross areas of the scene already visible. This procedure can be continued until the entire scene is classified.

4.2 Concluding remarks

The agent-centered representation just described provides a model of the scene with goals and sub-goals. Sub-goals are represented as points on the edge of obstacles, and are created in places where, if the agent were at that point, he or she would be able to see parts of the scene previously obscured by obstacles. It also provides a classification of each part of the scene (indeed, each pixel) as one of: obstacle, directly visible, accessible by a sub goal (or two, or three...), or not accessible at all. Some example agent-centered representations are shown in Figure 4.4. In this Figure, sub-goals are shown in yellow, with green lines linking each sub-goal to any further sub-goals it might lead to. The agent is shown as a red dot with a line indicating direction of travel. Areas of the scene which are directly visible are shown in white, and those accessible by one or more sub-goals are shown in successively darker shades of grey. Geographical goals are represented using a large blue dot, which is placed at the mean of the mixture model component. A dot is

used rather than the full spatial extent of the ellipse to prevent the illustrations becoming too cluttered. Cars are illustrated using small blue dots.

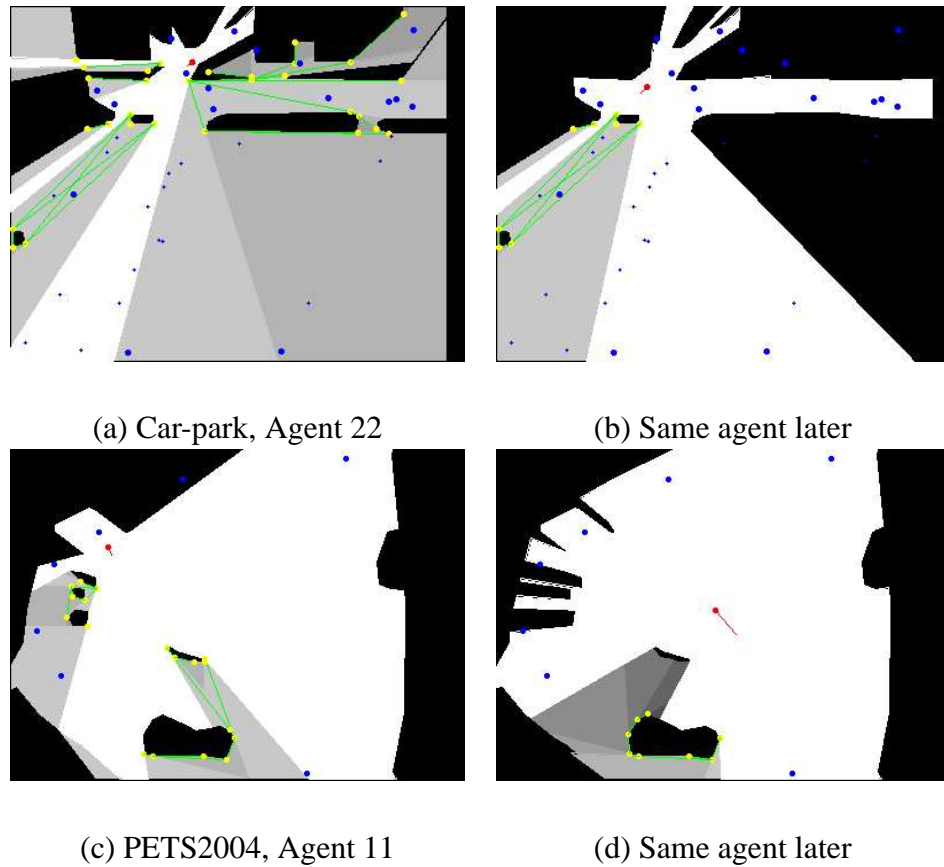


Figure 4.4: Some example agent-centered maps. The agent is shown as a red dot with a line indicating direction of travel, sub-goals are shown in yellow and lines between sub-goals in green. Goals are shown as blue dots. The area directly visible to the agent is shown in white, and areas visible by one or more sub-goals are shown in progressively darker shades of grey.

When constructing an agent centered representation of the scene, it is assumed that the agent either knows the area in question, or can see the whole area (that is, the obstacles are low enough for the people moving around the scene to see over). This assumption is implicit in the condition that sub-goals do not “open-up” areas of the scene which have been previously visible. Indeed in a completely unknown scene with high obstacles (like a maze, to take an extreme example), the approach described here would not be applicable at all. In such a situation the only thing it would be possible to say about the agent’s representation of scene is that the agent would know the area directly visible (shown in white in Figure 4.4), and they could probably work out the first level of sub-goals: those places where they could go to see further. They would not know what would be visible

from each of the sub-goals unless they were to walk up to each in turn. As this thesis is dealing with intentional behaviour, path planning, and rational agents, it assumes that the agents under investigation know enough about their environment to make rational choices about which route to follow.

Chapter 5

Navigational strategies and path comparison

The current chapter concerns itself with a general hypothesis, which involves looking at the behaviour of the people in the scenes under investigation and evaluating whether the intentional model implied by the previous chapter is a *good* model of this behaviour. For example, do we *really* navigate towards a goal in a piecewise linear fashion? There is a second type of evaluation, which can be called *Psychological Evaluation*, which is to be considered in Chapter 7. This second form of evaluation is intricately associated with an application of these ideas in a surveillance scenario and involves comparing the performance of algorithms based upon the intentional models described herein with the performance of humans in a surveillance task.

Taking a point near the start point of a trajectory as the origin it is possible to project all possible paths from the origin to each known goal as predicted by some navigational strategy. In this instance the sub-goal determination described in the last chapter is being used to predict possible future paths through the scene. The trajectory of the agent can then be compared to this tree of potential paths to the known goals within the scene, to determine whether their progress through the scene matches one of the predicted routes and if so, how closely they are following it.

In this chapter, two related models of human behaviour will be introduced and used to generate trees of ideal paths. The first, *simplest path*, predicts that the route people choose to take to a goal is one which consists of as few sub-goals as possible. The second

is *shortest path*, in which it is not the number of sub-goals which determine the choice of route but the overall length of the path to the final goal. In the process of investigating these goal-directed hypotheses, some possible measures of goal directedness will be introduced. Having described the navigational strategies to be considered, this chapter will go on to discuss various distance metrics for comparing trajectories to paths or routes, and concludes with some results.

5.1 Navigational strategies: Shortest path vs. Simplest path

There are two models of human navigation which are to be considered in this section. The first of these models will be called *simplest path*. The *simplest path* to a goal is the path which passes through the smallest number of sub-goals – if a goal is accessible by two sub-goals and also by three, *simplest path* predicts the agent will take the two sub-goal route even if it is longer than the three sub-goal route. This is achieved computationally at the area categorisation stage: if an area might fall into more than one categorisation, the “lowest level” characterisation is the one used. That is, if an area which is directly visible is also visible by a sub-goal or two, it is classified as *Directly Visible* rather than as *Accessible by sub-goal*. Likewise, if an area is accessible via one sub-goal it is not classified as *Accessible by 2 sub-goals* even if this is also appropriate. This ordering is a simple way of approximating the intuition that generally, people take the simplest path to their goal.

As was seen in Section 2.4.1, the way in which people actually plan a path through a scene is quite a complex matter. The most popular path planning strategy according to Golledge [51] is to find the shortest path between two points taking into account any obstacles. The second of the models to be considered here is just this – *shortest path*. The shortest path is made up of the shortest collection of straight line segments through free space, terminating at a goal and beginning at the trajectory start, changing direction at the tangential vertices of obstacles.

There are cases in which the shortest path is not one of the simplest paths, but in the majority of cases in the scenes discussed here, the two strategies predict agents will take the same path. As a result of this, any comparison of the two is only going to show small differences, however, it is still worthwhile as these small differences raise interesting questions about both the way we navigate and the quality of the models described here.

5.1.1 Path generation

The simplest path algorithm is laid out in Figure 5.1.

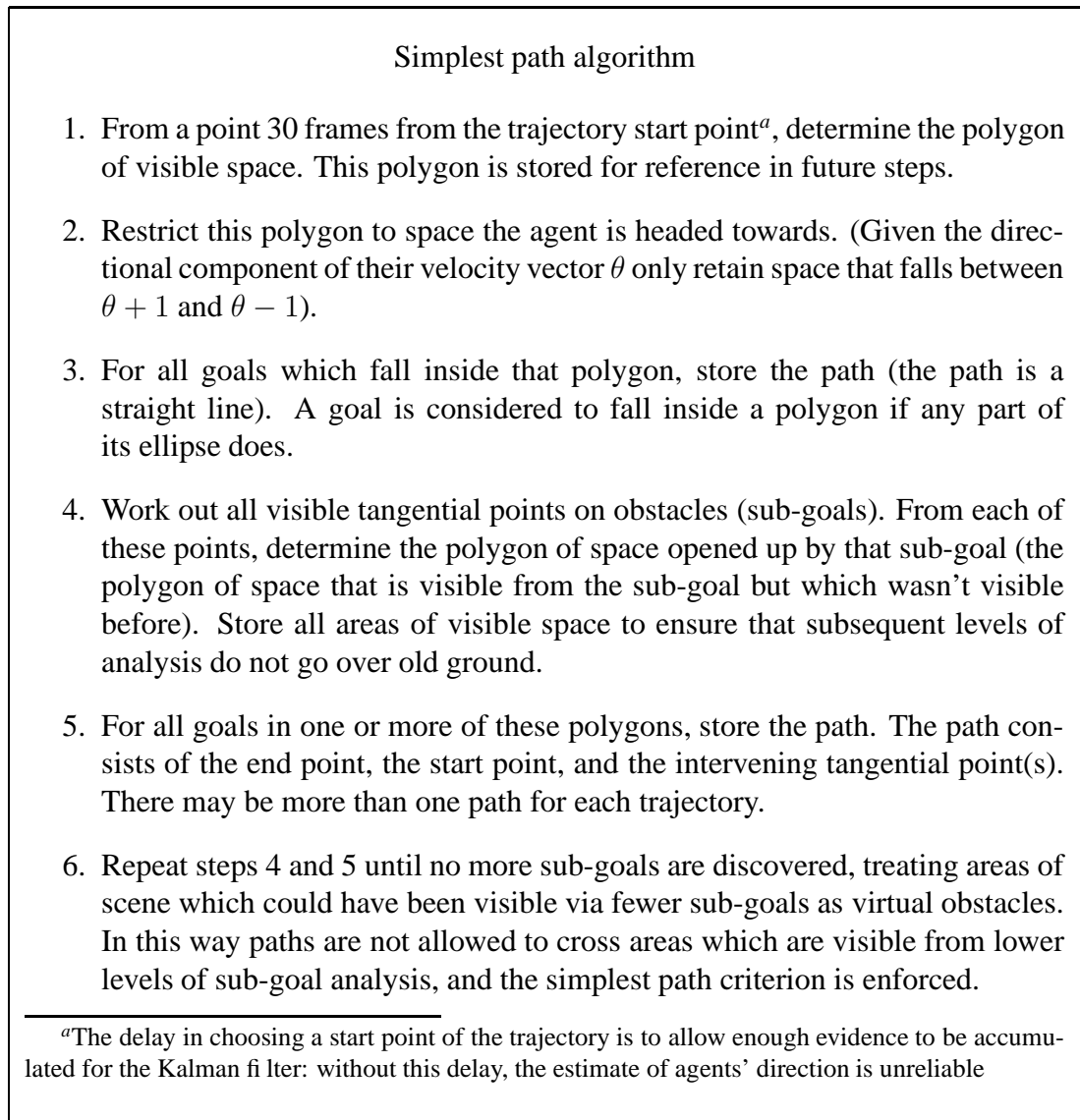


Figure 5.1: Simplest path algorithm

For shortest path, the algorithm is similar: indeed, steps one and two are identical. However, shortest paths are allowed to cross areas which *could* have been visible from a smaller number of sub-goals. Each path has to be internally consistent but not globally consistent, and this is achieved by searching recursively rather than by storing all areas of visible space at each level of sub-goal analysis. All paths to all goals are stored, and once the entire scene has been analysed the lengths of the various paths to each goal are compared to determine the shortest. The shortest path algorithm is described in detail in

Figure 5.2.

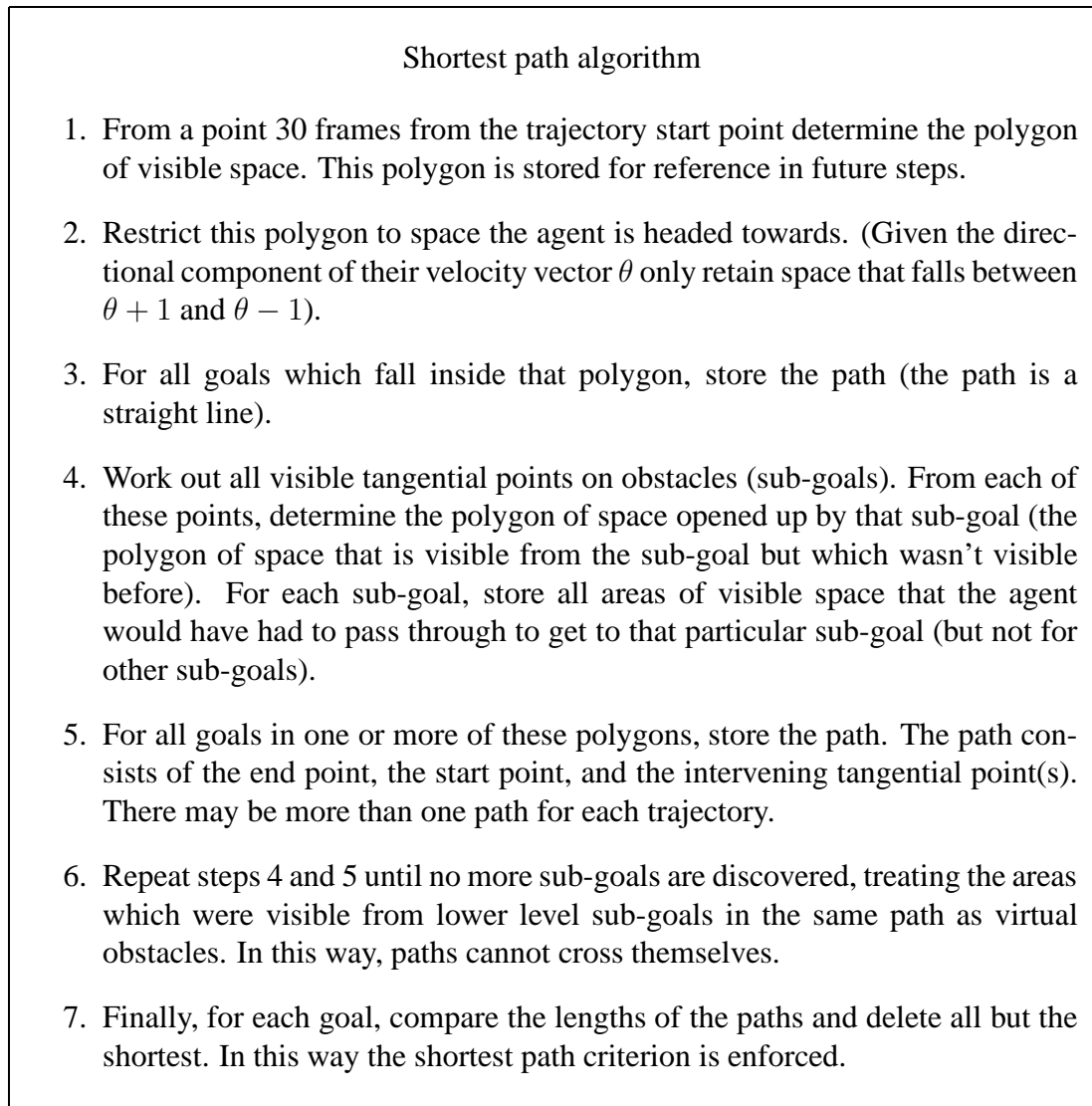


Figure 5.2: Shortest path algorithm

Illustrations of all projected simplest and shortest paths for two example agents are shown in Figure 5.3. These images show that both algorithms can result in a plausible path through the scene. The first two images also illustrate the way in which simplest path can result in more than one ideal path to a particular goal. The agents' trajectory is shown in black, simplest ideal paths in green and predicted sub-goals in yellow. The second two images depict shortest paths, with the shortest ideal paths in blue. A comparison of Figure 5.3 (a) and (b), and 5.3 (c) and (d) can highlight some of the differences between the two hypothesised navigational strategies. It is clear from these figures that simplest path and shortest path come up with different routes in certain situations.

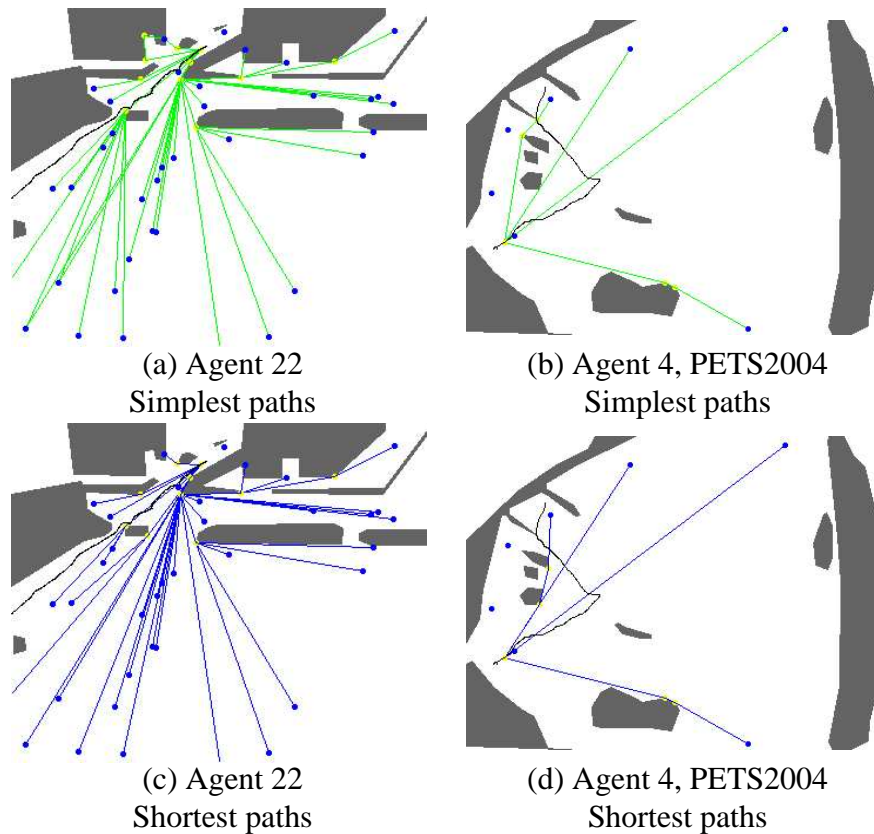


Figure 5.3: All potential simplest paths for agent 22 from the car-park dataset and agent 4 from PETS2004. Simplest paths are shown in green, and shortest paths in blue. The agent's actual trajectory is in black, with projected sub-goals in yellow. Comparing (a) and (b) with (c) and (d) shows some of the differences between the algorithms, with different routes being selected in some cases, and simplest path predicting more than one route to some goals.

As mentioned earlier, shortest and simplest paths are often the same. Calculated using the algorithms described above, in the car-park dataset over all agents, 10,964 simplest paths were predicted. The number of predicted shortest paths is smaller, at 9,675 (due to the fact that it is possible to have more than one simplest path to any particular goal). Of these 9,675 shortest paths, approximately 80% (7,824) were the same as one of the corresponding set of simplest paths.

5.2 Finding the closest path to the trajectory

This section considers various techniques for comparing the agent’s trajectory with the hypothetical paths detailed earlier. It begins with a consideration of the necessary transformation of image plane coordinates into an arbitrary ground plane coordinate system, goes on to discuss appropriate distance metrics for finding the closest path to a trajectory, and concludes with some illustrations.

5.2.1 Image to ground plane transformation

The car park scene covers a large area, and the height of a human at the front of the scene is approximately 30 pixels. The same human at the back of the scene is nearer 3 pixels in height. In order to compare distances between paths and trajectories consistently the scene over, a transformation from image to ground-plane is necessary.

In the current application such a complete camera calibration is not necessary as the recovery of relative distances between points is sufficient. In relatively flat urban scenes such as those considered here it is enough to assume a ground plane and model this as a flat surface at some angle to the image plane. The relationship between these planes can be calculated using plane-to-plane homography [58]. So to obtain “ground plane” coordinates all that is required is to project image plane coordinates back onto this plane using a 3x3 projective matrix. Following [86] this transformation can be estimated through the assumption of fixed road width - or in the case of the scenes described in this thesis, fixed width car-parking bays, and fixed width floor tiles.

5.2.2 Hausdorff measures

There are a number of ways to compare trajectories, some of which are detailed in [105] (which describes the evaluation of tracking applications). However in the current application the aim is something a little more abstract: comparison between a trajectory and a *path*. The proposed paths take the form of straight line segments through free space which

join or hinge on the tangential points of obstacles, and terminate near the start and at the end of the agents' actual trajectory. Thus if a path contains N_s sub-goals, that path can be represented by $2 + N_s$ points. A trajectory, on the other hand, is sampled at 15 frames per second and hence contains 15 points for every second the agent has been within the field of view of the camera. Clearly these are not amenable to direct point-wise comparison.

Ideally, the distance metric would serve two purposes: to predict or explain behaviour, and to enable the detection of inexplicable behaviour. The first of these requires that the metric be able to identify which of the known goals in the scene is the best explanation for the agent's trajectory. The second purpose requires that the distance metric provide some measure of fit, *goal directedness* or *intentionality*. This thesis proposes a different metric for each of these purposes: one to identify which of the ideal paths is closest to the agent's trajectory, and a second metric to determine the degree to which the agent's behaviour can be interpreted as following the sequential set of goals predicted by that ideal path.

The Hausdorff distance h is a measure of distance from a set of points X to a second set of points Y and is a maximin function defined as the maximum distance of a set of points to the nearest point in the other set, as formally set out in Equation 5.1. It is commonly used in computer vision to determine the degree of fit of a model with a set of image features, as in [66].

$$h(X, Y) = \max_{x \in X} \{ \min_{y \in Y} \{ \|x - y\| \} \} \quad (5.1)$$

This is an asymmetric measurement, and it is common for authors to take the Hausdorff distance *between* two sets - that is, to calculate the distance from each set to the other, and take the higher of the two. This measure (set out in Equation 5.2) provides an indication of the distance between two sets of points.

$$H(X, Y) = \max\{h(X, Y), h(Y, X)\} \quad (5.2)$$

Figure 5.4 (a) shows a trajectory and its associated ideal path. Calculating the Hausdorff distance between this ideal path P and the trajectory T would not provide a particularly accurate measure of the separation between the two. As the path is made up of just three points whilst the trajectory is made up of many, $h(T, P)$ is likely to be much higher than $h(P, T)$. To get around this problem the ideal path is first quantized by splitting it into m points (where m is the number of points in the corresponding trajectory). In this way $h(T, P)$ and $h(P, T)$ are made comparable.

The Hausdorff distance between two sets of points is a fundamentally directionless measure, whilst the trajectories and paths under examination have a natural progression

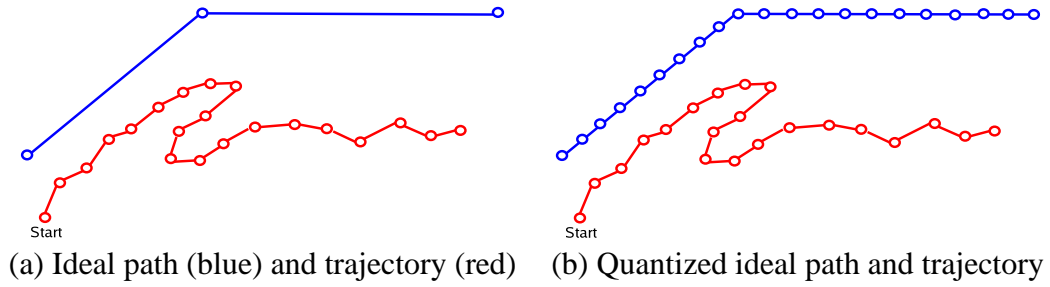


Figure 5.4: An illustration of a problematic trajectory

from point to point. Thus an alteration of the Hausdorff distance is proposed which takes into account this directionality: there are cases where the Hausdorff distance would produce a low measure, but the match between path and trajectory is very poor. An example of such a case is that of agent 267, who leaves their car, walks away, then returns directly to the car only to walk away again. In this case, the distance to the nearest point on the idealised path to the trajectory is always small. It is possible to reformulate Equation 5.1 as shown in Equation 5.3: for each member of the set X , x_i , we calculate the distance to a member y of the set of points Y based upon some function I of i . With the standard Hausdorff distance, $I(i)$ is defined such that $y_{I(i)}$ is the closest member of Y to the point x_i .

$$h(X, Y) = \max_{I \in \mathcal{I}} \left(\min_i \|y_{I(i)} - x_i\| \right) \quad (5.3)$$

With the proposed monotonic Hausdorff distance, an additional constraint is added: the point in Y selected by the function I must be the same distance or farther from the start point than its predecessor: \mathcal{I} is the set of all monotonically increasing functions $\{1, 2 \dots m\} \rightarrow \{1, 2, \dots m\}$. A diagram illustrating the difference in the selection of which point to match is shown in Figure 5.5. It is clear from Figure 5.5 (a) that with the standard Hausdorff distance, low matches can occur in situations where the agent doubles back upon themselves (as there is always a point on the idealised path near to the trajectory). Figure 5.5 (b) shows that by forcing the matched point to have a monotonically increasing distance from the start of the ideal path, this problem is avoided.

Figure 5.6 is similar to the earlier Figure 5.3, in that it shows all simplest and shortest ideal paths through the scene for the same example agents. Figure 5.6 shows the projected ideal paths with the closest path (according to the proposed monotonic Hausdorff measure) highlighted in red.

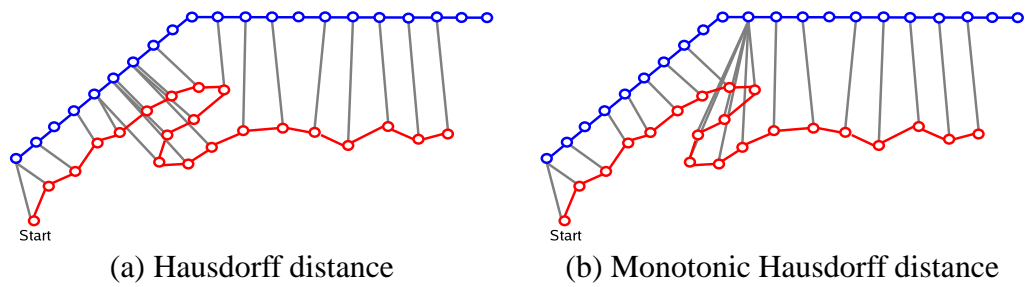


Figure 5.5: The selection of matched points in Hausdorff and Monotonic Hausdorff calculations

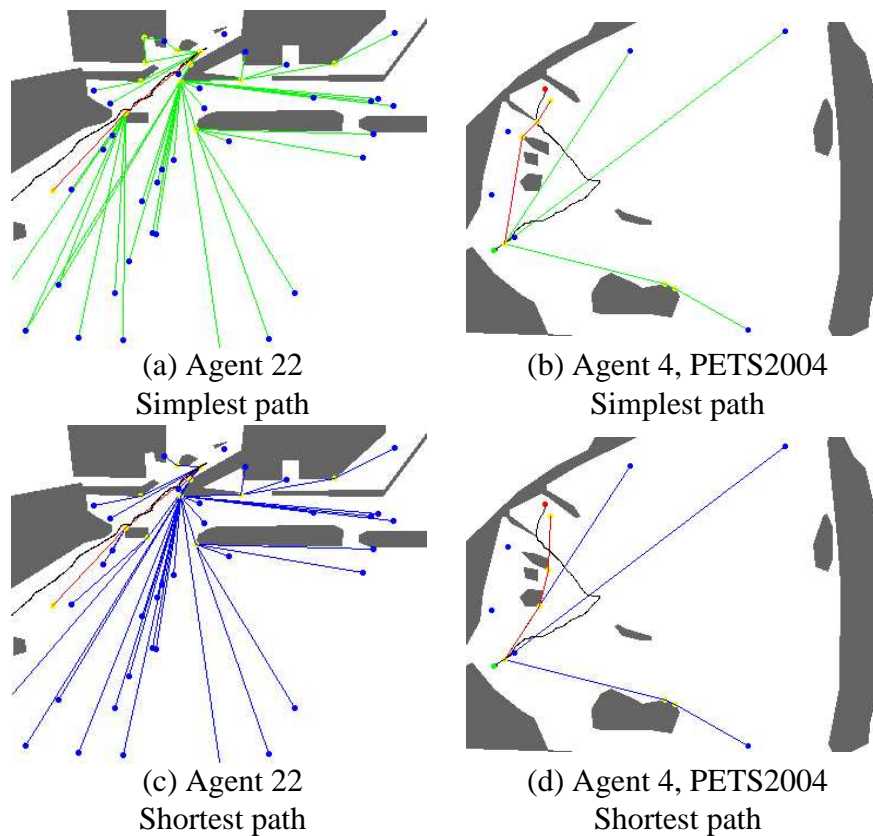


Figure 5.6: All potential simplest (green) and shortest (blue) paths, agents 22 (car-park) and 4 (PETS2004), monotonic Hausdorff closest path highlighted in red.

5.3 Measuring how closely the agent is following the path

The monotonic Hausdorff distance indicates which of the proposed routes through the scene the agent is likely to be following by determining the closest path in space. However, this distance metric is inadequate by itself: whilst it provides a measure of how *closely* the agent is following the specified path, it is not a good measure of goal-directedness. Within the car-park dataset, the monotonic Hausdorff distances range from 23.3 to 940 with an average of 136. An investigation of the trajectories confirms that for those agents with low monotonic Hausdorff distances, the trajectory and path are well matched.

The situation is not as clear cut in those cases where the trajectory and path are associated with high monotonic Hausdorff distances. The agent might be moving in the general direction of the sub-goal but (say) in parallel to the theoretical shortest or simplest path. This is still clearly goal-directed behaviour, but it is not captured by metrics which consider distance in space alone. A few examples of such agents are given in Figure 5.7. These agents have monotonic Hausdorff distance measurements in the top third of all cases, but are clearly still examples of goal-directed behaviour.

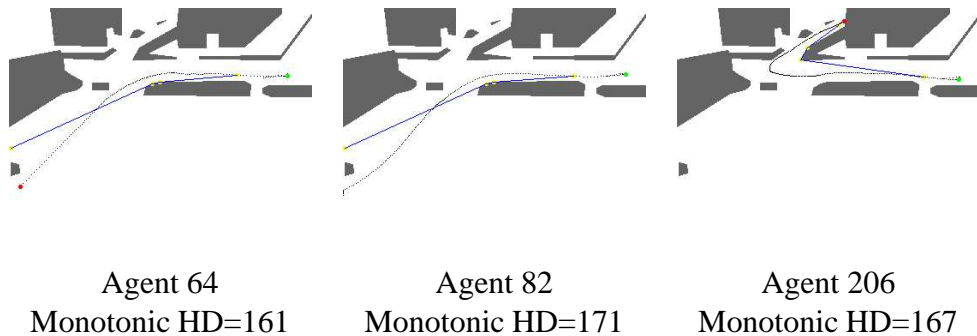


Figure 5.7: Examples of goal-directed behaviour where the monotonic Hausdorff distance is high

Each path (either simplest or shortest) is constructed as a number of straight line segments. Taking the angle of each of these segments ϕ and comparing it to the angle of motion of the agent θ provides us with an indication of whether the agent is moving in the general direction of the goal. Figures 5.8, 5.9 and 5.10 show the trajectories of the agent (in black), and the closest shortest path (in red). Also shown is a graph of the angular disparity between the agents' trajectory at each time step and each of the segments of the projected best path. Angular disparity is represented in this chapter as $\theta - \phi$, for simplicity, but the quantity being measured is in fact the difference in heading.

Agent 44, whose trajectory and ideal path are shown in Figure 5.8 is a car, and travels very smoothly through the scene. The graph of angular disparity in this case is very clear:

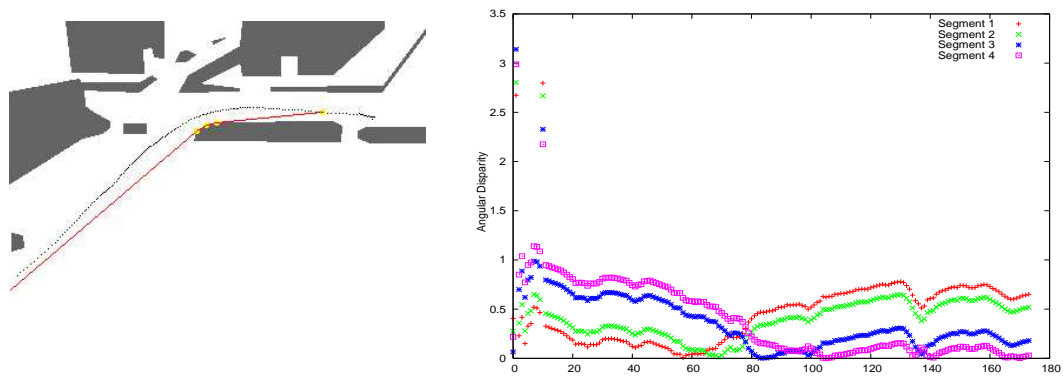


Figure 5.8: Agent 44, a trajectory identified as having 4 segments. The graph to the right shows angular disparity with each of the four segments throughout the length of the agent’s trajectory. From this graph, it is clear to see that the agent’s trajectory is closest in direction to segment 1 at the start of the trajectory, then closest to segment 2, then 3 and finally 4. This is as expected, as the predicted closest path goes through these segments in that order.

at any one point it is obvious which section has the lowest angular disparity (although it is worth noting that the noise at the start of the trajectory before the Kalman filter has stabilised is clear to see). Contrast this to the trajectory of agent 22, shown in Figure 5.9, where the two segments of the idealised path are very close in orientation, and the agent’s trajectory is noisier. In this instance, choosing where the path changes from one segment to the next is not really possible based upon angle alone. Agent 36, depicted in 5.10, shows a different type of problem. Two of the three sections of the idealised path have clearly different orientations, and choosing the point where the agent transitions from one to the next is straightforward based upon angle alone. However, the trajectory is noisy (due to the up-and-down bobbing motion of a walking person) yet clearly goal-directed.

By examining the angular disparity between the velocity of the agent and each of the segments of the ideal path identified as being closest, it is possible to determine at which point the agent moves from following one segment to following the next. These transition points between segments represent changes in the currently active goal of the agent: they move from heading towards a sub-goal to heading towards a goal, (or from one sub-goal to the next sub-goal). For example, according to the simplest path model the trajectory of Agent 44 is made up of four linear segments. The graph shown in Figure 5.8 supports this. It is possible to use angular disparity to work out where the transition in goals falls – at which point the agent reaches a sub-goal.

The trajectory could be partitioned (and “actual” sub-goal location found) by minimising over all possible sets of segment transition times the modulus of the total angular

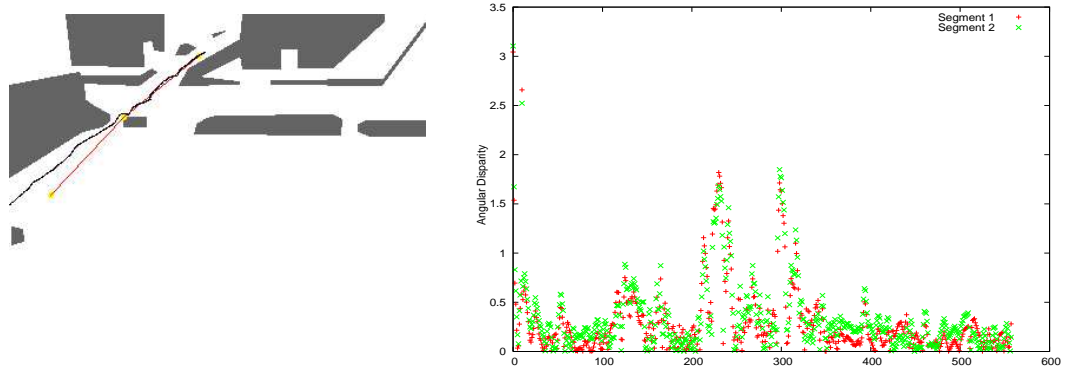


Figure 5.9: Agent 22, a trajectory identified as having 2 segments. This trajectory is noisier than that of agent 44 considered earlier. Whilst the predicted path consists of two segments, determining which segment the agent is following in a particular frame using angular disparity alone does not seem clear.

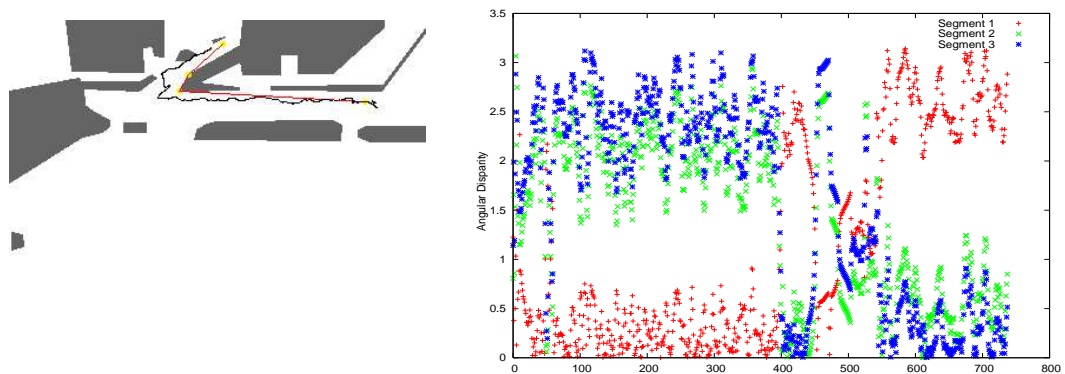


Figure 5.10: Agent 36, a trajectory identified as having 3 segments. The graph shown does indicate that for the first part of the trajectory, the agent is heading in the same direction as segment 1. Segments 2 and 3 are harder to separate as the trajectory is fairly noisy, but the general pattern of direction change does appear to fit the predicted path.

disparity between the direction θ_i of the agent at that time step and the direction of the corresponding segment ϕ_k as shown in Equation 5.4¹. This function serves two purposes: firstly, it partitions the trajectory by placing virtual vertices at times (v_k) which correspond to changes in direction (sub-goals), and secondly, it provides a measure of fit between the trajectory and the path. \mathcal{V} is the set of all ordered sequences of n transition times with $v_i = 1$ and $v_{k+1} = m + 1$. So in a trajectory such as that of agent 36, with three segments, this function has the effect of dividing the agent's trajectory into 3 segments ($n = 3$) based upon the direction of travel of the agent and the direction of each of the predicted path segments $\theta - \phi$. The result is normalised by dividing by the length of the trajectory. In the current implementation, minimisation is carried out by performing an exhaustive search over all \mathcal{V} . This is possible as the majority of ideal paths have fewer than 5 segments. Within a larger search space or a more complicated scene, techniques such as Dynamic Programming could be used to provide a quicker solution.

$$\min_{v \in \mathcal{V}} \sum_{k=1}^n \sum_{i=v_k}^{v_{k+1}-1} \frac{|\theta_i - \phi_k|}{m\pi} \quad (5.4)$$

In the majority of cases, this approach finds plausible locations for the change in direction: they fall near the sub-goals on the ideal path.

A problem with using straightforward angular disparity as a means of trajectory segmentation and path comparison is illustrated in Figure 5.11 (d). In this trajectory, the ideal path has a straight section which matches both the beginning and the end of the agent's trajectory. This has the effect that the first section matches almost the entire trajectory, and intervening sections are "pushed to the end". In this instance, angular measures alone are not sufficient and so a modified version of the distance metric has been developed.

The modified distance metric includes a penalty term taking into account the proportion of the trajectory assigned to each segment, as well as the angular disparity term shown in Equation 5.4. Equation 5.6 shows this modified distance measure, in which m represents the number of frames. The second component of this metric is a term which implies a penalty if the proportion of trajectory assigned to each path segment is not similar. In this, p_i represents the length of the i th path segment, and $\sum_n p_k$ the total length of the idealised path. As k varies, the proportion of the trajectory associated with each idealised path segment changes and s_k is the length of the k^{th} trajectory "segment" as shown in Equation 5.5. Thus $\sum_n s_j$ is the entire length of the trajectory. The second half of Equation 5.6, therefore, represents the proportion of trajectory assigned to each segment. λ is

¹It is worth noting again that the angular disparity is not, strictly speaking, a subtraction: it is the acute angular disparity between the two directions and shown here in the equations as a subtraction for simplicity.

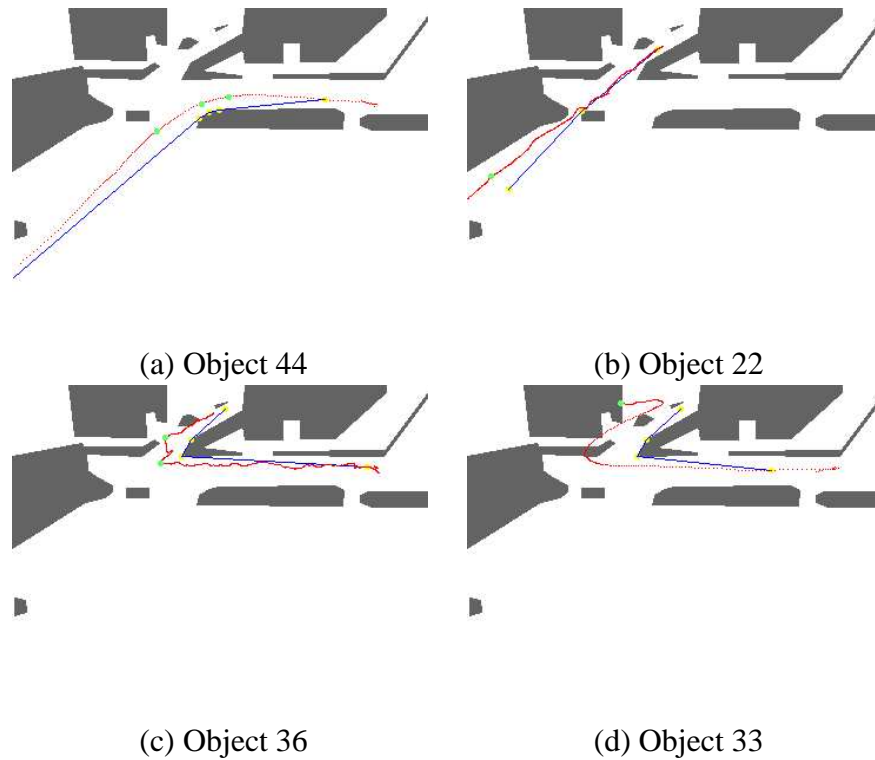


Figure 5.11: Trajectory turning points located using angular disparity alone: Object 33 shows that this is not always enough to place trajectory turning points, as in some cases, a strong match at the beginning of the trajectory can “force” turning points to the end. In the case of Object 33 shown here the problem is exacerbated by the fact that the end of the trajectory has a similar direction to the start, so that segment one of the ideal path has proved the best match for the entire trajectory.

a weighting term which is to be determined experimentally. Figure 5.12 shows the effect of varying λ . A value of 0.01 has been chosen for all experiments.

$$s_k = \sum_{i=v_k}^{v_{k+1}-1} \|x_i - x_{i+1}\| \quad (5.5)$$

$$\min_{v \in \mathcal{V}} \sum_{k=1}^n \left[\sum_{i=v_k}^{v_{k+1}-1} \left(\frac{|\theta_i - \phi_k|}{m\pi} \right) + \left(\lambda \left| \frac{s_k}{\sum_n s_j} - \frac{p_k}{\sum_n p_j} \right| \right) \right] \quad (5.6)$$

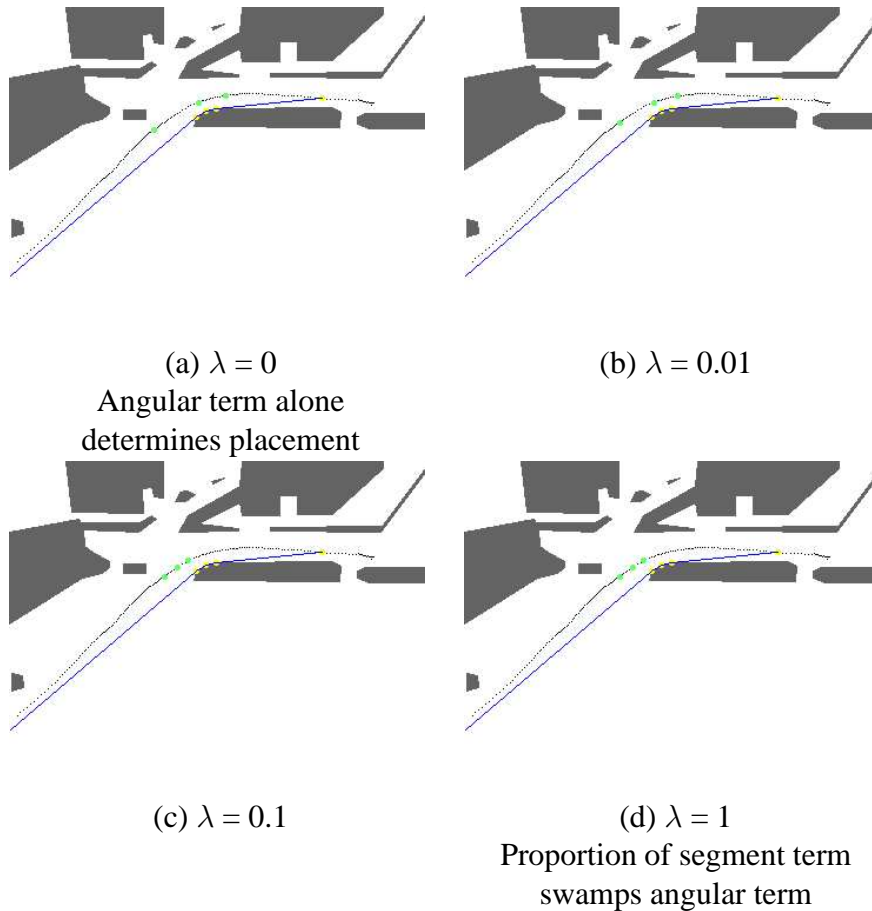


Figure 5.12: The effect of varying the weighting factor λ from 0 (where the angular term alone determines sub-goal placement) to 1 (where the proportion of segment term swamps the angular term).

Using the cost function set out in Equation 5.6 and minimising over all possible transition points from segment to segment in the ideal path (constraining segment order) creates a partitioning or segmentation of the agent's trajectory. By segmenting the trajectory into the same number of sections as the idealised path, the algorithm provides an indication of which path segment a particular individual is thought to be following at a particular time.

This, in a sense, is an *explanation*. A motive is attributed to the agent – they are heading towards a particular goal or sub-goal.

A measurement of fit, such as that provided by angular disparity measures or the cost function defined in Equation 5.6, can be thought of in two ways. Firstly, such a statistic is a measure of how *good* a particular explanation is for the behaviour at hand. It also provides a measure of intentionality. If the person in question is actually heading to the sub-goals and goals determined by the closest path, then the degree of fit with the closest path will be high. If, on the other hand, the agent is behaving erratically and not following a particular goal consistently then the fit between their trajectory and the closest path will be poor.

Table 5.1 shows some summary statistics comparing the cost function from Equation 5.6 for simplest path and shortest path. Table 5.2 shows the same information for angular disparity alone (Equation 5.4). Certain trajectories, such as that of agent 36 shown earlier in Figure 5.10, are noisy (particularly pedestrian trajectories). Both the cost function and a measure of angular disparity allow multiple small differences in angle to accumulate, leaving trajectories of this type with a relatively high cost. Agent 36, for example, has a cost score of 0.14 and an angular disparity sum of 0.438 – both of these are near the mean for the measure, indicating that the explanation is not particularly good even though the behaviour is goal-directed. Relaxing the criterion for goal-directedness and stating that an agent is heading towards a goal if the goal falls within half a radian either side of the agent’s velocity vector (as in Equation 5.7) provides a more useful measure, the summary statistics for which are shown in Table 5.3. This is computed by subtracting 0.5 radians from the angular disparity, summing over the length of the trajectory, but ignoring negative results. This final approach provides a more robust indication of goal-directedness, and the problematic agent 36’s trajectory scores 0.143: much more goal-directed than average.

$$\min_{v \in \mathcal{V}} \sum_{k=1}^n \sum_{i=v_k}^{v_{k+1}-1} \left(\frac{|\theta_i - \phi_k| - 0.5}{m\pi} \text{ if } \frac{|\theta_i - \phi_k| - 0.5}{m\pi} \geq 0 \right) \quad (5.7)$$

The shortest path metric results in explanations which score more highly, whichever of the three path-trajectory distance metrics is calculated (Equation 5.4, 5.6 or 5.7) – that is, the shortest path metric results in *better explanations*. In a small number of cases, there were no explanations produced as the agent in question was heading away from all goals at the 30 frame marker: these trajectories can be considered inexplicable. Cost scores for such trajectories were set to a high level (higher than the maximum of the auto-generated cost score for each metric). Some example explanations are discussed in the following

	Car-park		PETS2004	
	Simplest Path	Shortest Path	Simplest Path	Shortest Path
Mean	0.154	0.149	0.294	0.292
Median	0.111	0.100	0.256	0.230
Standard Deviation	0.146	0.147	0.177	0.178
Minimum	0	0	0.088	0.088
Maximum	0.712	0.712	0.661	0.661

Table 5.1: Cost function summary statistics (Equation 5.6)

	Car-park		PETS2004	
	Simplest Path	Shortest Path	Simplest Path	Shortest Path
Mean	0.507	0.475	0.935	0.916
Median	0.366	0.319	0.858	0.7065
Standard Deviation	0.464	0.467	0.558	0.562
Minimum	0	0	0.276	0.276
Maximum	2.233	2.233	2.08	2.08

Table 5.2: Angular disparity summary statistics (Equation 5.4)

	Car-park		PETS2004	
	Simplest Path	Shortest Path	Simplest Path	Shortest Path
Mean	0.231	0.214	0.541	0.525
Median	0.065	0.050	0.433	0.303
Standard Deviation	0.365	0.364	0.493	0.498
Minimum	0	0	0.406	0.0406
Maximum	1.737	1.737	1.59	1.59

Table 5.3: Angular disparity ignoring small angles summary statistics (Equation 5.7)

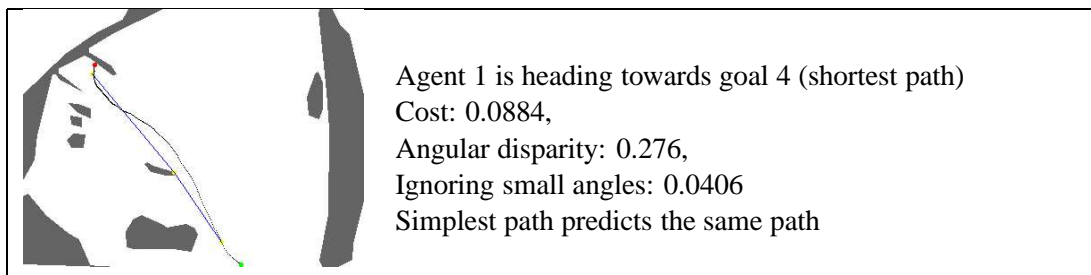


Figure 5.13: Sample explanations: Same path, low scoring (PETS2004)

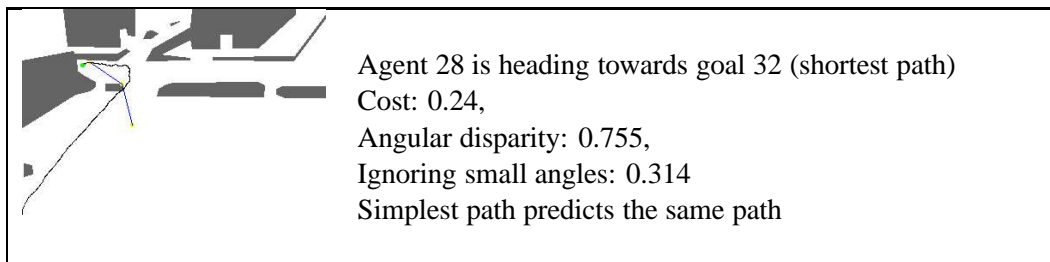


Figure 5.14: Sample explanations: Same path, high scoring (car-park)

paragraphs.

Figure 5.13 shows the trajectory and path for an agent whose predicted simplest path was the same as their predicted shortest path. This is also an example of a “good” explanation – the agent’s trajectory is close to the predicted path, and the various cost measures are low.

Figure 5.14 shows the trajectory and path for another agent whose predicted simplest path was the same as their predicted shortest path. This is an example of a “poor” explanation – the agent’s trajectory is quite far from the predicted path, and the various cost measures are above average.

Figure 5.15 shows the trajectory and paths for an agent whose predicted simplest path was different to their predicted shortest path. This is also an example of a “poor” explanation – the agent’s trajectory is quite far from the predicted path, and the various cost measures are above average. The simplest path algorithm, as in most cases, results in higher cost paths than shortest path.

Figure 5.16 also shows the trajectory and paths for an agent whose predicted simplest path was different to their predicted shortest path, although the difference is due to an artifact of the obstacle model. The long obstacle representing a hedge in the middle of the scene has a series of vertices along its left hand edge. These vertices arise during the straight line approximation stage of the obstacle model’s construction. However, the edge

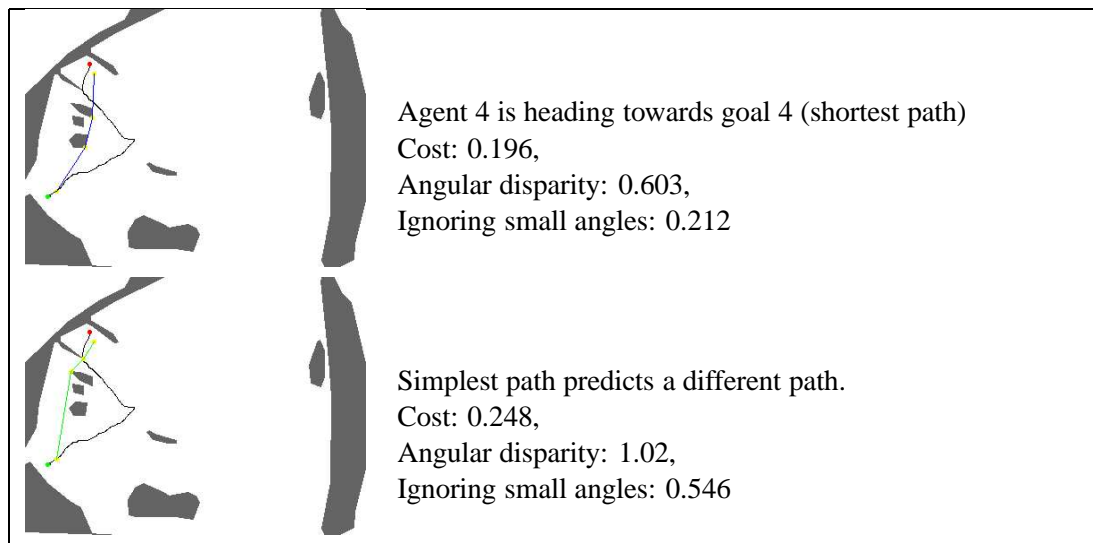


Figure 5.15: Sample explanations: Different path (PETS2004)

of the hedge should not really be considered to be a row of sub-goals: indeed, it should probably just be one. This problem occurs in a small number of cases, and is discussed in more depth in Chapter 9.

Figure 5.17 shows the trajectory for an agent whose behaviour is not consistent with any of the goals in the scene.

5.4 Concluding remarks

This chapter has described a way in which the behaviour of the agents within the scene can be modelled in terms of their intentions, and thus *explained*. A number of ways of determining how well the intentional explanation fits the agent's trajectory have also been described. Chapter 6 will describe an alternative measure of intentionality, in which a trajectory might be characterised as a "goal-set": a number of sequences of goal activations showing which of the goals in a scene an agent might be headed towards, away from, or just around.

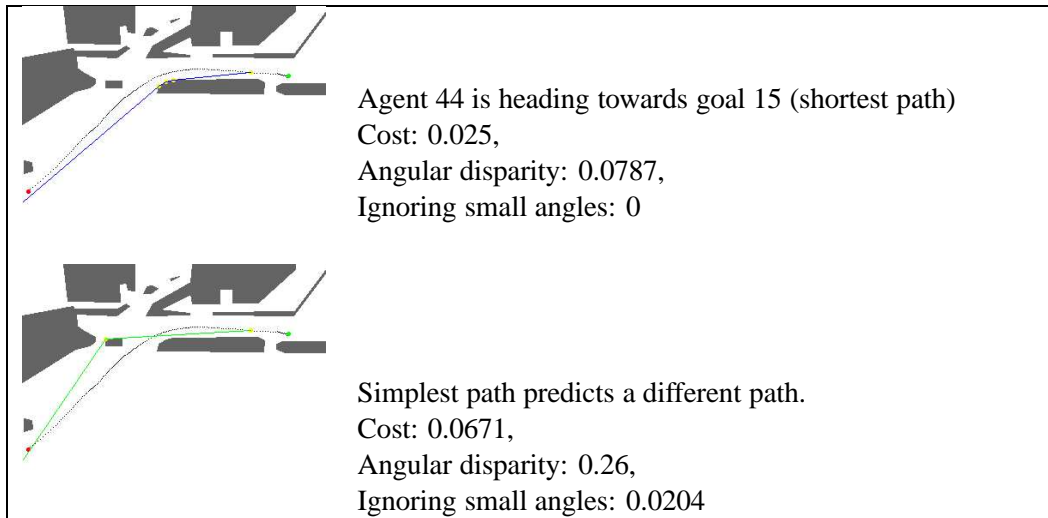


Figure 5.16: Sample explanations: Different path, problematic (car-park)

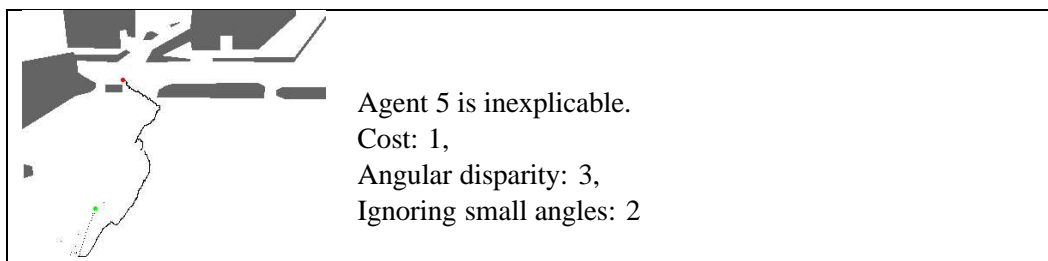


Figure 5.17: Sample explanations: inexplicable (car-park)

Chapter 6

Another way of measuring goal-directedness

In Chapter 5, two possible models of human navigation were discussed and compared, along with distance metrics for comparing agents' actual behaviour to the idealised paths predicted by these models. The output of these distance metrics was proposed as a means of measuring the intentionality of the agent. This chapter proposes a different measure of intentionality: one which develops as the agent moves through the scene. Unlike the methods set out in Chapter 5, it does not predict a complete path from a point near the start of the agent's trajectory, and hence is not as fragile. Indeed, it does not really deal with "paths" at all. With the shortest and simplest path metrics, if the path selected as closest was in fact not the path chosen by the agent, the fit between the path and the trajectory of the agent would be poor. If, for example, the agent took a sub-optimal path, or changed their mind, this would be penalised heavily. The algorithm set out in this chapter, by re-evaluating the paths and sub-goals at each time step, is more robust and allows for re-planning on the part of the agent. It is true that the methods outlined in Chapter 5 could be adapted to allow re-planning on the part of the agent, perhaps by re-calculating possible paths and distances at each time step. However, the computational cost of such an approach would be very high.

The algorithm presented here will be referred to as the *online* algorithm in this chapter, to distinguish it from the method presented in the previous chapter. It is called the *online* algorithm as there is a sense in which it re-calculates possible routes through the scene on

a frame-by-frame basis.

6.1 Goal classification

From Chapter 4, areas of the scene can be characterised in terms of their relationship to the moving agent, and it is possible to reason about that agent's possible goals. One way of conceptualising this interaction is in the form of a "goal-set" for each agent in the scene, the status of which changes over time as the agent moves through the field of view. In this goal-set a status is stored for each possible goal in the scene.

For each agent, for each frame, for each goal, it is possible to determine whether that goal is directly visible to the agent, or whether that goal is accessible to the agent by turning a corner or two. Indeed, there are four possible relationships between an agent and each goal for each frame. These can be determined from the label associated with the goal location on the agent centered map (described in Chapter 4) $Label(x_g)$, and the angle ϕ , which is the angle subtended by a line between the position of the goal x_g , the position of the agent x , and the agent's current direction θ . Within the bitmapped representation, the state is determined by investigating the pixel label at the position of the goal, and within the polygonal representation by determining which of the various polygons (visible area, or one of the sub-goal polygons, or no polygon at all) the goal falls inside. The possible states are:

1. S_0 : The goal is directly visible: $Label(x_g) = \mathbf{V}$; and the agent is heading towards it $-1 < \phi < 1$. g_1 is in this state in Figure 4.2.
2. D : The goal is directly visible to the agent: $Label(x_g) = \mathbf{V}$; but they are heading away from it: $\phi > 1$ or $\phi < -1$. g_2 is in this state in Figure 4.2.
3. N : The goal is not visible to the agent: $Label(x_g) = \mathbf{N}$ (it is on the other side of an obstacle, and is not reachable by means of a sub-goal). g_3 is in this state in Figure 4.2.
4. $S_1, S_2, S_n \dots$: The goal is visible to the agent, but only via one or more sub-goals (S_1, S_2, S_N): $Label(x_g) = \mathbf{Sn}$. g_4 is in state S_1 in Figure 4.2.

This goal-set implies how each agent might navigate to each of the specific goals (exits) identified within the scene, but does not strictly speaking specify a route.

It is worth noting that in the polygonal implementation fewer goals will in general be classified as N as there is no cap upon the depth of sub-goal analysis, so more of the scene is "opened up" by sub-goals.

6.2 Using goal classification to explain behaviour

The following stage of analysis provides a unification of these frame-by-frame classifications in order to determine whether or not a particular goal is a viable *explanation* for the trajectory as a whole. This approach relies upon the general assumption that our agent chooses a piecewise linear path between tangential points (sub-goals) with a decreasing number of remaining turns as the path progresses. Thus, if a goal is accessible by one turn and later by two, this goal has become a less likely explanation for the agent's actions. This is directly related to the Simplest Path algorithm from Chapter 5.

The relationships described in the previous section are context-free: they just depend upon the location and direction of travel of the agent in that specific frame. The next stage is to classify each goal as consistent or inconsistent with the trajectory so far. Essentially, we look at the pattern of state transitions associated with each goal in turn, asking the question “*Is this a possible explanation for the agent's behaviour?*” or “*Could they be headed towards this goal?*”. Figure 6.1 shows an example trajectory. From this picture it is clear to a human observer which goal the agent is headed towards: it is Goal II. We can describe the process of moving towards Goal II as a sequence of goal-state labels: it starts in state $S1$ as the goal is around a corner (the bottom left corner of the L-shaped obstacle in the image), becomes $S2$ for a while, then $S1$ again, and finally becomes directly visible and enters state $S0$. This progression through successive levels of sub-goal indirection implies navigation towards a goal without actually predicting a path.

With goals near the boundary between labels, noise in the direction measurement can cause noise in the categorisation. To minimise the effects of this noise, classification information is “smoothed” by voting over a five frame moving window: for each frame, the categorisation of each goal is replaced by the most common categorisation (the mode).

Our model predicts that people will move directly and purposefully towards their goal. Translating this into state transitions, we can say that those goals which are consistent explanations for the behaviour so far will be those that the agent travels towards. Those goals in $S2$ are two levels of indirection away from the current position, those in $S1$ one, and those in $S0$ zero - thus those goals which are consistent will have transitions of the sort $S2 \rightarrow S1 \rightarrow S0$, and will probably stay in any or all of these states for some number of frames. To obtain a measure of explicability, we associate a cost with those state transitions that imply a particular goal is *not* an explanation for the current trajectory. Thus, if a goal G is in state $S0$ and moves to state N , that agent was heading towards G and it was directly visible, but is now in a position where G is not visible at all. G is now less likely to be the final goal for the agent – the explanation for their behaviour.

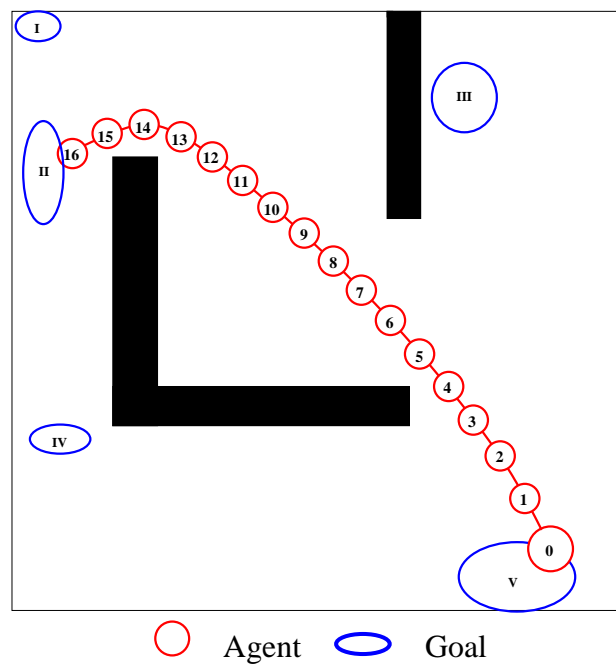


Figure 6.1: An example trajectory: frame numbers inside circles. For the first few frames, goal II is just one sub-goal away, then it becomes two sub-goals away. Around frame 11 there is just one corner between the agent and the goal and finally it is directly visible.

Figure 6.2 shows the transitions possible in the model and their associated costs, and Table 6.1 shows an illustration of the sorts of patterns of activity resulting from agent behaviour.

Overall cost is calculated for each goal within the scene and normalised by dividing by the length of the agent's trajectory. These measures will be called C , to distinguish it from the Cost function described in Equation 5.6. If this model is correct, the goal with the lowest C can be thought of as the most likely explanation for the behaviour of the agent. Lowest C is an intuitively appealing metric, which has the virtue of simplicity. Aggregate measures such as mean-goal-cost would be flawed in this situation, as perfectly goal-directed trajectories will move consistently away from some of the goals in the scene.

Consider the five fictitious goal activation patterns shown in Table 6.1 (corresponding to the behaviour pattern depicted in Figure 6.1). Goal III starts in the field of view of the agent, and the agent is heading towards it. Then the agent heads away from it, and then the agent can't see it directly at all. This goal is highly unlikely to be an explanation for the agent's behaviour. However choosing the most likely pattern of activation might not always be a case of choosing the goal with the lowest C . Compare goals I and II, which illustrate a problem with the lowest C approach: Goal I stays within the agent's field of view with the agent heading towards it throughout the length of the agent's trajectory.

Frame	Goal I Class:Score	Goal II Class:Score	Goal III Class:Score	Goal IV Class:Score	Goal V Class:Score
0	SO:-	S1:-	SO:-	SO:-	SO:-
1	SO:0	S1:0	SO:0	SO:0	D:1
2	SO:0	S2:1	SO:0	SO:0	D:1
3	SO:0	S2:0	SO:0	S2:1	D:1
4	SO:0	S1:0	SO:0	S2:0	D:1
5	SO:0	S1:0	SO:0	S2:0	D:1
6	SO:0	S1:0	S1:1	S2:0	D:1
7	SO:0	S1:0	N:1	S2:0	D:1
8	SO:0	S1:0	N:1	S2:0	D:1
9	SO:0	S1:0	N:1	S2:0	D:1
10	SO:0	S1:0	N:1	S2:0	D:1
11	SO:0	S1:0	N:1	S2:0	S3:0
12	SO:0	SO:0	N:1	S1:0	S3:0
13	SO:0	SO:0	N:1	S1:0	S2:0
14	SO:0	SO:0	N:1	S1:0	S2:0
15	SO:0	SO:0	N:1	SO:0	S1:0
16	N:1	SO:0	S2:0	SO:0	S1:0
Totals	1	1	11	1	10
C	0.0625	0.0625	0.6825	0.0625	0.625

Table 6.1: Patterns of goal activity over time, corresponding to Figure 6.1. Total row represents the number of frames for which the goal has been inconsistent with the trajectory, and Cost represents the total normalised by dividing by trajectory length.

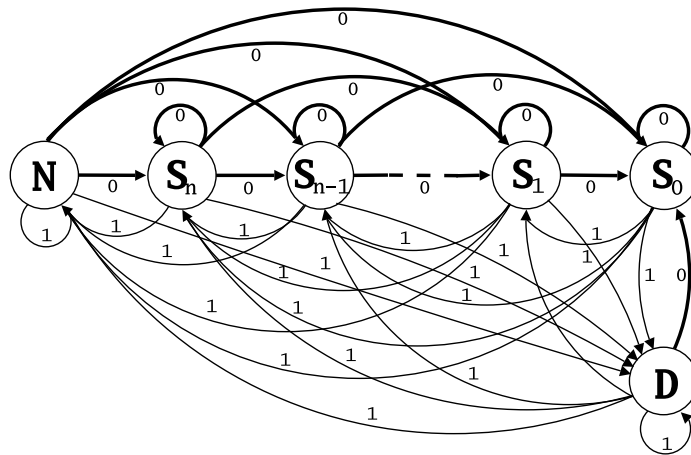


Figure 6.2: State transition diagram indicating the cost of each transition. Those transitions which are *free* (drawn with thick lines) are those associated with progress towards the particular goal; those with a cost are those associated with movement away from the goal

Goal II, however, starts off visible by one sub-goal, then the agent moves into a position from which they could reach the goal by turning two corners (the goal is in state S2). Then the agent turns the first of these corners, taking the goal to state S1 and finishes with the goal directly visible. The C of both of these trajectories (shown in Table 6.1) is 0.0625: so it is difficult to tell from C alone which of the goals is the explanation for the agent's behaviour.

Given these considerations, it is necessary to look at various metrics to determine which of the C scores to use for each trajectory. The one associated with the nearest goal to the trajectory end? Or the lowest? This is a decision to determine experimentally, and will be considered in Chapter 7.

6.3 Concluding remarks

This chapter has introduced a novel way of thinking about an agent's trajectory, by first building an agent centered representation of the scene for each time step as described in Chapter 4 and then by providing a framework for unifying the goal-directed elements of this representation over the course of an agent's entire trajectory. The result of this process is a set of "C scores" for each goal within the scene. Each C score is an indication of the extent to which the agent's trajectory is consistent with motion towards that goal. It is suggested that taking one of these C scores can provide some measure of the overall "goal-directedness" of the trajectory – a measure of intentionality. After normalising by

dividing by trajectory length, a goal with a *C* score of 0 indicates that the agent has been consistently travelling towards that goal throughout their time in the scene, and a *C* score of 1 indicates that the agent has been consistently moving away from that goal.

It is worth emphasising that the algorithm described in this chapter accounts for the history of an agent's trajectory by noting the state in which goals are classified within the agent-centered map at each time step. This has the effect that agents are penalised for changing their mind, but not greatly. If an agent were to head directly away from a goal for a small time then move around an obstacle, to return to that goal by a different path, the only part of the trajectory that would be penalised for this particular goal is the initial "heading away from" period. No record is made of what areas of the scene have been directly visible.

Contrast this with the shortest path and simplest path algorithms detailed in the previous chapter. In these, possible paths through the scene were calculated 30 frames (2 seconds) into the agent's trajectory. The tree of projected paths is fixed at this point, and if an agent were to change their mind, they would be penalised for this throughout the rest of the time they were within the camera's field of view. Which of these approaches is the best to adopt within a surveillance application is a question to be determined empirically, and that is the subject of the next chapter.

Chapter 7

The measurement of interesting behaviour

The previous chapters have provided various metrics for the measurement of intentionality or goal-directedness. Chapter 6 did so by associating a cost with each goal within the scene, and then choosing some metric for selecting a cost to represent the trajectory as a whole. Chapter 5 did so by determining all possible intentional paths through scene and measuring the distance from the agent's real trajectory to these idealised paths. This chapter concerns itself with the related hypothesis that unusual or *interesting* behaviour is that which is not obviously intentional or goal-directed. This hypothesis implies that those trajectories with particularly high cost scores or distance measurements are in some way interesting; that these metrics capture something about the real world.

One of the main contributions of this chapter is to introduce a novel evaluative schema for *interesting event detectors*. This involves comparing the performance of the algorithms described earlier against the performance of human volunteers performing a surveillance task. Thus the model is evaluated against human performance at explaining and classifying the behaviour of other humans. The previous chapters described the specific hypotheses about human navigation which are being investigated, and the current chapter evaluates the likelihood that such a model is useful in detecting the kinds of events that surveillance operatives pick out from video streams. This is being considered separately to the question of whether this family of models provides an accurate representation of human behaviour, a question which was discussed in Chapter 5.

This chapter begins by considering previous event detection systems and the way in which they have been evaluated. It goes on to propose a psychological evaluation criterion, based upon human observers' opinions of the trajectories in question. The human observation data is analysed and then compared to various software generated cost scores, for both the car-park dataset and then for the PETS2004 dataset. The chapter concludes with an evaluation of the exit model.

7.1 The evaluation of event detection

Determining the overall effectiveness of interesting event detection algorithms has historically been unsystematic. This is acknowledged by the authors of [139], who stated they were working on methods of evaluating the unusual event detection aspect of their work. Evaluative techniques for such systems have, at their simplest, involved investigating the problematic cases by hand. This involves looking at the outliers – and saying “Yes, that’s unusual” [78, 139]. One such model, trained on pedestrians, had a major outlier which turned out to be a cyclist. This serves to provide confirmation that the model provides a reasonable basis for the detection of strange pedestrian activity, however, the confirmation such evidence provides is at best anecdotal. It is also completely self-justifying – if you look at the examples which do not fit the model, and find they are odd in some way, then of course they are interesting *to you* – by definition, they don’t fit your model of what is going on.

Another means of evaluating such systems is through the use of “actors”. These people are recorded behaving in an unusual fashion, and the system in question is evaluated on its ability to single out the sequences featuring the strangely behaving actors [78, 103]. For example, in [71] interesting behaviour was defined as rapid head movements; in [157] the interesting behaviour detected was people driving in circles or zig-zags in a car-park; and in [117] suspicious behaviour is defined as behaviour which is deceptive (such as avoiding visible areas). Problems with this approach are manifold, but all hinge upon the question of *whose idea of interesting or unusual we are dealing with*. If the decision as to what constitutes unusual behaviour is left up to the actors, questions about who the actors are, what their preconceptions of the project might be, and most importantly, their links to the software designers, become paramount. If the actors are lab-mates of the researcher, do they know how the algorithm in question works? The alternative case, where the actors are instructed by the system designer on the nature of unusual behaviour, could be even worse. It is easy to imagine a scenario in which the instruction “We need some footage of suspicious behaviour, like walking from car to car across the car park in a wavy line” is

issued.

A recent step towards more systematic evaluation has been made in [163]. In the “*Challenge for real time event detection solutions*” or CREDS, researchers were invited to try their software on a specific scene. The scene is from the Paris metro and systems were tested on their ability to issue warnings when certain pre-defined events such as *walking on rails* or *dropping objects on tracks* were detected. A number of camera configurations (both visible and infra-red) and scenarios (such as *walking on rails*) were released for researchers to use as training data and to fine-tune their algorithms. The submitted software was tested for its ability to produce alarms corresponding to the hand-crafted ground truth. The systems demonstrated as part of the CREDS challenge [14, 126, 129, 136] detected some activities with ease – mostly by defining areas of scene which were forbidden unless the moving object happened to be a train. Some of the systems submitted for the challenge were fully-fledged surveillance systems which were capable of detecting events not specified in the challenge (such as that of Black et al [14], which could detect graffiti and abandoned packages). As a challenge, in which surveillance systems were evaluated against each other and against ground truth the results are interesting and a move towards more objective evaluation. However, the sequences all appear to be performed by actors, and there was no separate test dataset meaning that the systems were evaluated against the training data.

7.2 Psychological evaluation

Computer Vision systems for surveillance are generally model based. Things which do not fit the model can only be classed as unusual or interesting with respect to that model. It is not valid to claim that events which fall outside the model are interesting or unusual unless some sort of comparison with external events can be obtained. All that can really be said about them is that they don’t fit the model. What is required is a more principled way of evaluating the performance of these systems. The aim of this chapter is to propose a way out of this model-based trap by providing a form of “ground-truth” for interestingness.

Within the surveillance domain, interesting events are events which might be associated with criminal or dangerous behaviour. One recent study [149] investigates whether such events can be predicted from CCTV footage – that is, whether it is possible to distinguish sequences where a crime was about to occur from neutral sequences. The authors conclude that not only is it possible, but that naïve observers perform as well as trained security guards. This suggests that there is no learned ability to detect the type of

events security guards or surveillance operatives detect. Given this finding, benchmarking against a number of humans can be assumed to be an improvement over relying on the author, actors, or serendipity to provide some measure of the interestingness or otherwise of the data set. Indeed [149] leads us to the conclusion that those characteristics which could bias performance are associated with experience of the software in question, rather than experience or lack of experience within the surveillance domain.

The evaluative schema proposed here involves requiring a number of volunteers (in this case, undergraduate and postgraduate students with no knowledge of the project being evaluated) to rank the behaviour of each agent in each scene in question. To assist in this task, separate videos are produced for each agent containing only those frames of video encompassing the agent's trajectory. A highlight (in the form of a dot) indicates exactly the agent in question – this makes the cognitive task of those evaluating much easier in scenes with multiple, occluded agents. In the case of pedestrians, it also serves to obscure the agent. This has the benefit of forcing the evaluator to concentrate on the pattern of activity rather than the appearance of the agent.

Volunteers are asked to rate the “interestingness” of these videos on a scale of 1 to 5. The instructions given to the volunteers were as follows:

“If you were a security guard, would you regard the behaviour of the agent highlighted in this video as interesting? Please indicate on the following questionnaire, with one being uninteresting and five being interesting.”

Volunteers were also invited to note down any comments they wished to make about any of the videos.

An average of the scores from the subjects is then assumed to provide a simple measure of “interestingness”. This can then be compared directly to the output of any machine generated indication of typicality, intentionality, or any other surveillance-related metric. If a binary decision (interesting, or not) is required, we can use ROC (Receiver Operating Characteristic) graphs to assist in the determination of a threshold. ROC graphs come from signal detection theory and are a means of visualising the behaviour of a classifier. They are plots of sensitivity against 1-specificity (the fraction of true positives vs. the fraction of false positives) as a threshold is varied. They clearly show the trade-off in sub-optimal classifiers between setting the threshold for classification very high and rejecting everything resulting in no true positives, but no false positives either; and setting the threshold very low resulting in a 100% true positive detection rate and a 100% false positive rate. Chapter 8 contains ROC curves for the systems described in this thesis.

However, the mean or median is just one statistic it is possible to use: the advantage of having the opinions of a number of people is that there is a richness of information to

incorporate within the evaluative process. For example, correlation statistics can be calculated – both within the human set (to determine consistency between human rankers) and between the set of human rankings and the machine generated statistic. The statistical variance between the human rankers can provide an indication of whether there was disagreement over the behaviour of a particular agent – it might be desirable to flag up behaviours where all humans agreed that behaviour was unusual or interesting, but less important to flag up those where there was disagreement between subjects.

As well as the possibility of performing a range of statistical tests we have a wealth of qualitative information in the form of comments made by the subjects as they were ranking the dataset. These can help in instances where disagreement occurs – for example, in the car-park scenario an object was reported as being highly interesting by several subjects, uninteresting by others, and the trajectory taken by that object was very dull. Inspection of the comments on their forms revealed that it was interesting because it was an ambulance.

Whilst the ultimate aim of an automated surveillance system is a binary decision – interesting, or not – in the real world events and behaviours fall on a continuum. Footage from a car-park might be largely uninteresting, but it is still possible to say that event A is *more interesting than* event B. By using the average rating of human volunteers, this evaluative schema allows us to take advantage of this.

7.3 Correlation statistics

There are two correlation statistics applicable to this data: Spearman's Rho [28], and Kendall's Tau [81]. The data is clearly non-parametric and on different scales – that is to say that any computer generated statistic is unlikely to map directly onto a 1-5 rating of interestingness. Nevertheless, if those videos rated highly by the computer are those videos rated highly by the human volunteers this is a positive result, and so rank correlation methods are appropriate for detecting any such relationship. Spearman's Rho (r_s) is calculated by first ranking the data and then using Pearson's product moment correlation calculation on the resultant ranks. Pearson's is calculated using the formula given in Equation 7.1 [28] in which x_i and y_i are matched pairs of ranks.

$$r_s = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{[\sum_{i=1}^n (x_i - \bar{x})^2][\sum_{i=1}^n (y_i - \bar{y})^2]}} \quad (7.1)$$

r_s can be tested for significance: for small values of n , r_s has a non standard distribution and specific tables must be used. For large ($n > 10$) values of n the function

in Equation 7.2 [28] of r_s follows approximately the distribution of a t-test statistic with $n - 2$ degrees of freedom.

$$t_s = r_s \sqrt{\frac{n-2}{1-r_s^2}} \quad (7.2)$$

The resultant value t_s can be compared against any standard statistical tables for significance testing.

The second applicable correlation statistic, Kendall's Tau, operates differently. Whilst it utilises the same underlying information (ranks of scores), it is not directly comparable to Spearman's Rho. Instead of relying upon the numerical difference between ranks, it only takes account of the relative orderings of ranks. To calculate Kendall's Tau (T_k), calculate the total number of concordant ranks (agreements in ordering, i.e., cases where Judge A ranked Object 1 more highly than Object 4, and Judge B also ranked Object 1 more highly than Object 4), and the total number of discordant ranks. Tied ranks are ignored for the purposes of determining concordance or discordance. In a case with no tied ranks, T_k only requires knowledge of concordant and discordant judgements and the number of judgements. With tied ranks, the number of tied ranks in each set of judgements is also required. The formula for calculating T_k is given in Equation 7.3 [133], in which T_x and T_y are the terms correcting for tied ranks.

$$T_k = \frac{\text{concordant} - \text{discordant}}{\sqrt{n(n-1) - T_x} \sqrt{n(n-1) - T_y}} \quad (7.3)$$

As is the case with r_s , the distribution for T_k is also known for the null hypothesis of no relationship between variables. Indeed, T_k is approximately normally distributed and using Equation 7.4 [133] can be converted to z scores then compared with standard statistical tables in cases where $n > 10$.

$$z = \frac{3T_k \sqrt{n(n-1)}}{\sqrt{2(2n+5)}} \quad (7.4)$$

Numerically, T_k and r_s are not directly comparable, having different underlying scales. There is an approximate relation between the two measures expressed by Equation 7.5, and they use the same amount of information and have the same sensitivity. However, the way in which they are calculated is obviously different, and they can be interpreted differently [82]. Spearman's Rho is a special case of the Pearson's correlation – a significant result according to Spearman's Rho tells us that there is a linear relationship between the *ranks* of the X and Y data (and so a relationship, perhaps nonlinear, between the *values*).

A significant value of Kendall's Tau, on the other hand, tells us about the probability of X and Y being in the same order in the observed population [133]. Other than the differences in interpretation just listed, and some concerns about the reliability of Spearman's Rho with small sample sizes [81], there seem to be no clear situations in which one statistic is preferable to the other and so both will be presented here. Generally r_s is larger than T_k and the relationship between the two measures is described approximately by the inequality shown in Equation 7.5 [133].

$$-1 \leq 3T_k - 2r_s \leq 1 \quad (7.5)$$

7.4 Between-human correlations

For each agent, a separate movie containing only those frames of video which encompass that agent's trajectory was produced with the agent of interest clearly highlighted throughout. Volunteers were asked to rate the "interestingness" of these videos on a scale of 1 to 5 as detailed in Section 7.2. For the car-park dataset, the number of volunteers was 7 ($n_s = 7$) and for the PETS2004 dataset, the number of volunteers was 12 ($n_s = 12$). Before comparing these rankings to any machine generated statistic, it is worth investigating the amount of discord and determining whether the naïve subjects are in agreement about what constitutes interesting.

7.4.1 Ranking of the car-park dataset

The car-park dataset used in the experiment includes 256 ($n = 256$) trajectories, including 6 performed by actors. As described earlier, there are inherent problems associated with the use of actors in this type of work. However, the main aim of this evaluation is to abstract away from the specific opinions of an individual and instead compare the opinions of many judges about the behaviour in question. Given this, and given the great difficulty in obtaining footage of genuine suspicious or criminal behaviour, it is believed that the use of actors in this situation is acceptable.

All between human correlations were positive and significant at the 0.0001 level for $n = 256$. These between-human correlations are shown in Table 7.1 for Spearman's Rho, and in Table 7.2 for Kendall's Tau. Throughout this chapter, results significant at the 0.0001 level are shown in **boldface**, and those significant at 0.001 in *italics*. The cut-off points for significance with $n = 256$ at the 0.0001 level are 0.164 for T_k , and 0.241 for R_s . For the 0.001 (0.1%) level, they are 0.139 for T_k and 0.205 for R_s . At the bottom

of each correlation matrix, the correlation of each subject's ranks with the mean rank is also given. Such a correlation is of limited statistical validity, as each subject's ranks have contributed to the mean. For this reason, also included is a correlation with the mean of the *other* human rankers on a leave-one-out basis (i.e., the ranks from volunteer number 1 have been correlated the average of volunteers 2 through to 7).

H	1	2	3	4	5	6	7	Mean
1	1							
2	0.64	1						
3	0.75	0.71	1					
4	0.54	0.43	0.51	1				
5	0.67	0.62	0.61	0.43	1			
6	0.59	0.58	0.61	0.32	0.78	1		
7	0.56	0.62	0.57	0.36	0.63	0.59	1	
Mean	0.57	0.7	0.61	0.32	0.72	0.75	0.84	1
Leave-one-out	0.55	0.62	0.59	0.32	0.68	0.67	0.66	N/A

Table 7.1: Between-human Spearman's correlation matrix, car-park dataset

H	1	2	3	4	5	6	7	Mean
1	1							
2	0.63	1						
3	0.75	0.7	1					
4	0.54	0.43	0.5	1				
5	0.66	0.6	0.6	0.42	1			
6	0.58	0.56	0.6	0.31	0.76	1		
7	0.54	0.6	0.56	0.35	0.61	0.57	1	
Mean	0.55	0.68	0.59	0.31	0.7	0.72	0.82	1
Leave-one-out	0.53	0.6	0.56	0.3	0.65	0.64	0.64	N/A

Table 7.2: Between-human Kendall's correlation matrix, car-park dataset

7.4.2 Ranking of the PETS2004 dataset

The PETS2004 dataset consists of pedestrian footage filmed in a foyer situation, with actors performing various roles such as meeting, walking, fighting and browsing. Included in the dataset are various people we assume are bystanders. Agents whose trajectories are only partially covered by the video, and those agents who hover on the periphery have been excluded. In short, only the main actors in each scene, and those bystanders whose trajectories are shown in full are analysed. This leaves a total of 23 agents from 12

movies. Given the small size of this dataset it is possible to visualise it in its entirety. Such an illustration alongside summary information (mean score, standard deviation, median score) is given in Table 7.3.

The 12 original videos are used to produce 23 ($n = 23$) labelled videos. These were presented to 12 subjects ($n_s = 12$), who rated each on the 1-5 scale as detailed in Section 7.2. Spearman's Rho and Kendall's Tau were calculated for each pair of human raters, giving correlation matrices with 66 entries ($\frac{n_s^2 - n_s}{2}$). All of these correlations were positive, and are shown in tables 7.4 for Spearman's Rho and 7.5 for Kendall's Tau. Those significant at the 0.001 level are shown in *italics*, and those significant at the 0.0001 level are shown in **boldface**. The cut-off points for significance with $n = 23$ at the 0.0001 level are 0.583 for T_k and 0.723 for R_s . At the less significant 0.001 (0.1%) level, the thresholds are 0.641 for R_s and 0.493 for T_k . As with the previously shown correlation matrices in Tables 7.1 and 7.2, at the end of each table, each subject's correlation with the mean and with the mean of all other subjects are given as an indicator of subject reliability.

7.4.3 Consideration of high-variance cases

Within the car-park dataset, the vast majority of cases (243/256) had little or no disagreement between rankers (with the difference between maximum and minimum rank being 2 or less). This shows remarkable levels of agreement between the 7 subjects, and will be at least in part due to the fact that the vast majority of trajectories within the car-park dataset were considered uninteresting. Within the PETS2004 dataset, there were more disagreements between subjects (11/22 cases had little or no disagreement).

It is interesting to take a closer look at the behaviour of those agents where the human rankers were in disagreement – where the standard deviation of the human scores is high. Some of these were due to partial trajectories, and to the inclusion of people such as Agent 0 from the PETS2004 dataset, who entered the scene then immediately turned around and left (it can be assumed he was a passer-by, perhaps put off by the camera). In particular, there are five cases where the human rankings range from lowest (1) to highest (5) and it is worth investigating these in a little more detail:

- **Car-park Agent 98:** Standard deviation = 1.57, mean 4.14. In this movie, a car is parked in a particularly unusual fashion. Comments indicate that those who thought it odd considered the parking to be very poorly executed (one subject wrote “Are they on drugs?”). The one subject who thought the clip uninteresting did not comment.
























ID	Image	Description	Mean score	SD of scores	Median score
0		Walks in, waves at camera, goes back through same door	3.33	1.07	3.5
1		Walks slowly across scene	1.25	0.45	1
2		Walks out, turns around, walks back through same door	2.08	1.16	2
3		Walks slowly across scene	1.58	0.67	1.5
4		Enters, meets, shakes hands, changes direction, exits	1.92	1.16	1.5
5		Enters, meets, shakes hands, changes direction, exits	1.92	1.16	1.5
6		Enters in a wobbly fashion, falls over, gets up and leaves	4.67	0.65	5
7		Leaves scene, re-enters, slumps on floor, leaves scene again	3	1.48	3
8		Walks towards person, shakes hands, turns, leaves scene	2.5	1.62	2
9		Walks towards person, shakes hands, turns, leaves scene	2.33	1.56	2
10		Walks in straight line across scene	1.33	0.78	1
11		Walks in straight line across scene	1.5	0.67	1
12		Walks in relatively straight line across scene	1.5	1	1
13		Walks in relatively straight line across scene	1.58	1.16	1
14		Walks in, fights, runs out	4.75	0.62	5
15		Hangs around, Walks in, fights, runs out	4.67	0.65	5
16		Walks in, fights, runs in circles, runs out	4.75	0.62	5
17		Enters, gets fought with and knocked over, leaves	4.33	1.15	5
18		Wanders aimlessly	2.08	0.9	2
19		Wanders aimlessly	1.92	0.9	2
20		Walks directly across scene	1.17	0.39	1
21		Walks in, waves at camera, leaves	2.75	0.97	3
22		Wanders towards bookshelves, browses, leaves	1.67	0.78	1.5

Table 7.3: Overview of the PETS2004 dataset

h	1	2	3	4	5	6	7	8	9	10	11	12
1	1											
2	0.93	1										
3	0.65	0.55	1									
4	0.84	0.76	0.6	1								
5	0.75	0.76	0.66	0.68	1							
6	0.83	0.77	0.65	0.79	0.79	1						
7	0.68	0.69	0.69	0.77	0.74	0.76	1					
8	0.63	0.6	0.77	0.63	0.5	0.62	0.6	1				
9	0.92	0.82	0.61	0.94	0.69	0.85	0.74	0.62	1			
10	0.83	0.75	0.53	0.88	0.71	0.77	0.71	0.55	0.85	1		
11	0.71	0.67	0.8	0.68	0.67	0.53	0.65	0.62	0.68	0.55	1	
12	0.89	0.8	0.84	0.8	0.83	0.8	0.72	0.7	0.82	0.79	0.79	1
Mean	0.94	0.86	0.76	0.81	0.76	0.8	0.68	0.72	0.87	0.78	0.81	0.92
Leave-one-out	0.94	0.8	0.67	0.79	0.71	0.65	0.66	0.64	0.85	0.71	0.78	0.86

Table 7.4: Between-human Spearman's correlation matrix, PETS2004 dataset

H	1	2	3	4	5	6	7	8	9	10	11	12
1	1											
2	0.87	1										
3	0.54	0.44	1									
4	0.79	0.7	0.51	1								
5	0.65	0.66	0.58	0.59	1							
6	0.73	0.67	0.57	0.72	0.71	1						
7	0.62	0.63	0.63	0.73	0.67	0.71	1					
8	0.52	0.46	0.67	0.56	0.38	0.55	0.55	1				
9	0.84	0.73	0.5	0.89	0.59	0.76	0.68	0.53	1			
10	0.72	0.65	0.43	0.82	0.63	0.68	0.66	0.46	0.75	1		
11	0.63	0.59	0.69	0.64	0.61	0.45	0.6	0.49	0.61	0.48	1	
12	0.83	0.71	0.76	0.74	0.75	0.73	0.67	0.6	0.73	0.68	0.71	1
Mean	0.84	0.73	0.6	0.71	0.65	0.68	0.58	0.58	0.75	0.65	0.68	0.82
Leave-one-out	0.84	0.67	0.53	0.68	0.59	0.54	0.56	0.49	0.72	0.58	0.66	0.76

Table 7.5: Between-human Kendall's correlation matrix, PETS2004 dataset

- **PETS2004 Agent 0:** Standard deviation = 1.07, mean 3.33. In this movie clip, the agent walks out and waves at the camera, then leaves the scene by the same door. The actor in this clip is presumably signalling to the camera person that they are ready to go, although this was not clear from context.
- **PETS2004 Agent 7:** Standard deviation = 1.48, mean 3.0. In this movie, the agent walks out of the scene (the clip clearly starts before the actor is ready) then re-enters, crosses to the object on the left, then sits on the floor for a short while before leaving. Some of the subjects think that sitting on the floor is uninteresting.
- **PETS2004 Agent 8:** Standard deviation = 1.62, mean 2.5. This clip and that of Agent 9 (see below) feature agents entering the scene from different doors, meeting in the middle, and then leaving from different doors. Comments by those subjects who rated these clips highly indicate that they thought a package was passed between the two actors – which would have been suspicious given the instructions to subjects.
- **PETS2004 Agent 9:** Standard deviation = 1.56, mean 2.33. See the entry for Agent 8 above.
- **PETS2004 Agent 17:** Standard deviation = 1.15, mean 4.33. This agent enters the scene, gets into a fight, is knocked over and then leaves. It is difficult to determine why one subject did not find this clip interesting, as they did not comment.

7.4.4 Concluding remarks upon the human ranks

These considerations (both the correlation results and the consideration of individual disagreements) suggest a very high level of agreement between subjects about what constitutes *interesting* behaviour. Using r_s , 48 out of 66 PETS2004 between human correlations are significant at the 0.001 (0.1%) level. Using T_k , slightly more results are significant with 57 out of 66 reaching the same level of agreement. With the car-park dataset there is even more agreement, as all between-human correlations are significant at the higher 0.0001 (0.01%) level. The higher level of agreement within the car-park dataset is probably due to the dull nature of an hour's car-park footage – very little of interest occurs.

Whether this level of agreement is because the human subjects are looking for the same types of behaviour pattern – whether there is some common underlying cause – is not a conclusion we can draw from correlation results alone and therefore remains an open question, if intuitively likely. A more interesting open question arises if we assume such

an underlying agreement exists, which is whether or not subjects' judgements hinge upon the *intentionality* or otherwise of the agents. This is a particularly difficult hypothesis to test.

That there was some disagreement between the human subjects on some of the clips should not be seen as a drawback to this evaluative schema - indeed, one of the reasons for including a number of subjects is to allow for such differences and disagreements. These help provide a richer framework against which to evaluate the software. It is also worth noting that in several of the worst cases of inter-subject disagreement, this was due to just one subject ranking the agent's behaviour as uninteresting. As the disagreement of one subject does not affect the mean rank unduly, it is reasonable to take mean rank as the basis for evaluation of the various sub-goal algorithms.

7.5 Comparing human rankings with computer-generated scores: Shortest path and simplest path

Three measures of intentionality were proposed in Chapter 5 for the comparison of a projected ideal path with an agent's trajectory. All of these first require the determination of the closest ideal path, and then for the trajectory to be segmented into the same number of sections as the corresponding ideal path. The simplest of the three measures is angular disparity (hereafter *AD*), in which the agent's direction of travel is compared to the direction of the relevant path segment and absolute difference values are summed over the length of the agent's trajectory (Equation 5.4). The second to consider is angular disparity, but ignoring small angles (hereafter *IS*): this is calculated by subtracting 0.5 from the angular disparity before addition and ignoring values under 0 (as set out in Equation 5.7). The final metric to evaluate is Cost, which takes into account relative proportions of the path as well as angles, as set out in Equation 5.6.

Correlations between the mean of the human rankers and each of these metrics for the car-park dataset are set out in Table 7.6 and for the PETS2004 dataset in Table 7.7. All car-park correlations are positive and significant at the 0.0001 (0.01%) level, showing that there is a strong positive relationship between the scores and the human perception of "interestingness". The results from the PETS2004 dataset show a lower level of significance: throughout this chapter, correlations are shown in boldface if they are significant at the 0.0001 level or better and in italics if they are significant at the 0.001 level but not the 0.0001. The PETS2004 correlations with shortest and simplest path metrics are significant at the lower 0.5% (0.005) level, thus they are not displayed in any altered font,

but are still representative of a strong correlation. The cut-off points for significance at the 0.005 level with $n = 23$ are 0.565 for Spearman’s Rho and 0.421 for Kendall’s Tau.

	<i>AD</i>	<i>AD</i>	<i>IS</i>	<i>IS</i>	Cost	Cost
	Shortest	Simplest	Shortest	Simplest	Shortest	Simplest
R_s	0.4	0.38	0.44	0.4	0.37	0.37
T_K	0.31	0.3	0.35	0.32	0.29	0.29

Table 7.6: Correlations between use of shortest and simplest path metrics and the human rankers, car-park dataset

	<i>AD</i>	<i>AD</i>	<i>IS</i>	<i>IS</i>	Cost	Cost
	Shortest	Simplest	Shortest	Simplest	Shortest	Simplest
R_s	0.61	0.6	0.63	0.62	0.6	0.6
T_K	0.44	0.43	0.45	0.45	0.44	0.44

Table 7.7: Correlations between use of shortest and simplest path metrics and the human rankers, PETS2004 dataset

Comparing the three metrics, the most highly correlated is *IS*: angular disparity ignoring small angles. The cost function and the simple angular disparity function (*AD*) perform similarly. Shortest path metrics correlate as well or better than simplest path metrics in all cases. This supports the conclusions of Chapter 5 in which shortest path metrics were found to provide more plausible ideal paths than simplest path metrics.

Correlation results between these path metrics and each of the individual human rankers are presented in Appendix A, in Tables A.1 and A.2 for the car-park dataset, and Tables A.8 and A.9 for the PETS2004 dataset.

7.6 Comparing human rankings with computer-generated scores: The online algorithm

In this section, correlations between the mean human rank for each agent and various computer generated cost scores for the online algorithm will be presented. C will be defined as the cost of a particular goal in the scene, calculated as set out in Chapter 6, normalised by the length of the trajectory (in frames). The questions to be answered in this section are as follows:

- Does the polygonal representation provide any improvements over the bitmapped representation?

- Is it better to use the cost of the lowest-cost-goal or the cost of the nearest goal to the trajectory end?
- What are the effects of limiting or extending the depth of search of the sub-goal algorithm?

Each of these different variations on the C score will be denoted by subscripts – p for polygonal, b for bitmapped, l for lowest cost, c for closest cost, and a number to denote the depth of the sub-goal searching. Thus, C_{bl2} is the cost of the lowest cost goal in the bitmapped representation, with sub-goal search capped to 2 levels of look-ahead, and C_{pc} is the cost of the closest goal to the trajectory end, calculated using a polygonal representation with no limit to sub-goal search depth.

7.6.1 Bitmapped or polygonal representation?

Table 7.8 gives the correlation results between the C score, each human and the mean human, for both the bitmap based model C_b and the polygonal model C_p . Those results which are significant at the 0.0001 (0.01%) level are shown in **boldface**, and those which are significant at the 0.001 (0.1%) in *italics*. Due to computational considerations, the bitmapped representation only performs two levels of sub-goal analysis. In order to perform a direct comparison between the bitmapped and polygonal representations, the polygonal model was also capped at two levels of analysis whilst generating these results. The lowest cost goal is used.

Car-park Subject	Bitmapped model C_{bl2}		Polygonal model C_{pl2}	
	Spearman's	Kendall's	Spearman's	Kendall's
Mean Human (Car-park)	0.40	0.32	0.43	0.36
Mean Human (PETS)	0.63	0.48	0.74	<i>0.56</i>

Table 7.8: Correlation statistics for the car park dataset comparing polygonal and bitmapped implementations

From Table 7.8 it is clear that the implementation based upon a polygonal obstacle model outperforms that based upon the bitmapped obstacle model in all situations. The limitations of the bitmapped model hinge upon the problem with obstacles with curved edges, where the saw-toothed nature of a bitmap (at the pixel level) led to rows of sub-goals being formed close to each other. The polygonal model does not suffer from this deficiency and hence the sub-goal structure created within such a model is a more accurate representation of a piecewise linear navigation through the scene. Correlation results

comparing the choice between bitmapped or polygonal models and each of the individual human rankers are presented in Appendix A, in Table A.3 for the car-park dataset, and Table A.10 for the PETS2004 dataset.

7.6.2 Lowest-cost goal or closest goal to trajectory end?

The output of the sub-goal algorithm described in Chapter 6 is a cost associated with each goal in the scene. In order to have a measure of intentionality or explicability for the trajectory as a whole, it is necessary to choose either some function of these costs, or to choose one particular cost as the measure for that trajectory. Given that any number of goals within the scene might incur maximum cost (the agent might be headed away from them from the start of the trajectory to the finish), any function which creates some aggregate goal cost is going to be influenced heavily by any such goals. It is also true that a perfectly explicable trajectory might just have one explicable goal.

The two obvious choices for a measure of explicability for the trajectory as a whole are the *lowest cost goal* and the *closest goal to the trajectory end*. The lowest cost goal is of interest as it is the goal most consistent with the trajectory to date. The closest goal to the trajectory end is of interest as presumably that is the goal the agent was pursuing throughout the trajectory.

A major disadvantage of choosing the closest goal to the trajectory end is that to do this, you need to know where the trajectory ends. This is in a way begging the question, and would preclude any online use of the system. On the other hand, using the lowest cost goal as a measure enables the online calculation of C scores on a frame-by-frame basis. This has obvious advantages for an interesting behaviour detection system. If the software were to be used to ring an alarm in a surveillance situation, it would be much less useful if it were only able to raise the alarm after the agent in question has left the scene.

Figures 7.1, 7.2, 7.3 and 7.4 show some trajectories selected to illustrate different possible configurations of closest-goal and lowest-cost-goal with the obstacle model drawn in green, the closest goal to the trajectory end marked as a red dot and the lowest cost goal marked as a blue dot. Figure 7.1 shows two trajectories where the closest and lowest goal were the same or very near to each other, which is what we would expect and also the most common occurrence (more than half of all trajectories have this configuration).

Figure 7.2 shows a fairly common occurrence, in which a long trajectory “over-shoots” a corner. In these cases, the closest goal to the trajectory end incurs cost for the portion of the trajectory where they “over shoot” the corner, as for this section of their

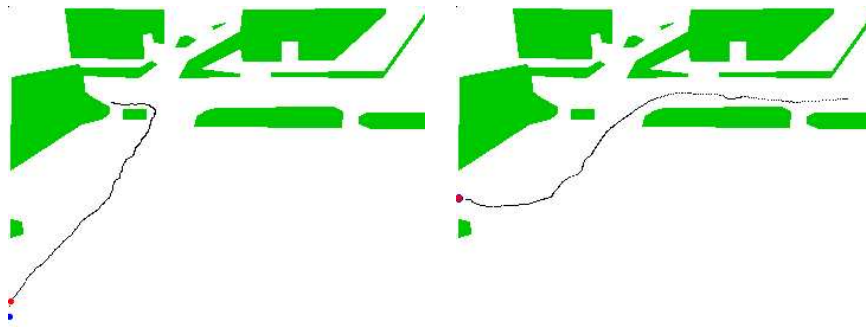


Figure 7.1: Examples where closest goal to finish (red) and lowest cost goal (blue) were the same or very near to each other

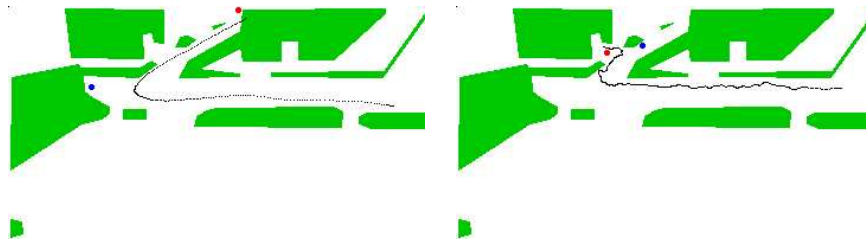


Figure 7.2: Illustrations showing the effect of over-shooting a corner upon goal cost. Lowest cost goal (blue) is consistent with early parts of the trajectory. Closest goal to finish shown in red.

journey the agent is effective *heading away* from the goal. Thus, in these cases, the lowest cost goal is consistent with the majority of the trajectory.

Figure 7.3 shows the situation where a trajectory has more than one lowest cost goal (often, in these situations, one of the lowest cost goals is also the closest).

Figure 7.4 shows two of the most “interesting” trajectories, in which a complicated path leads to lowest cost goals some distance from the closest cost goal. It is unsurprising that the algorithms designed to model intentional behaviour provide strange results in cases like these, in which the trajectory of the agent is not indicative of simple geographically goal-directed behaviour.

Table 7.9 shows the correlation results comparing cost scores using closest goal and lowest cost goal with the mean human rating. All correlations are positive, and in the case of the car-park dataset most are also significant. Somewhat surprisingly, the closest-goal correlation results are lower than the lowest-cost-goal correlation results in all situations.

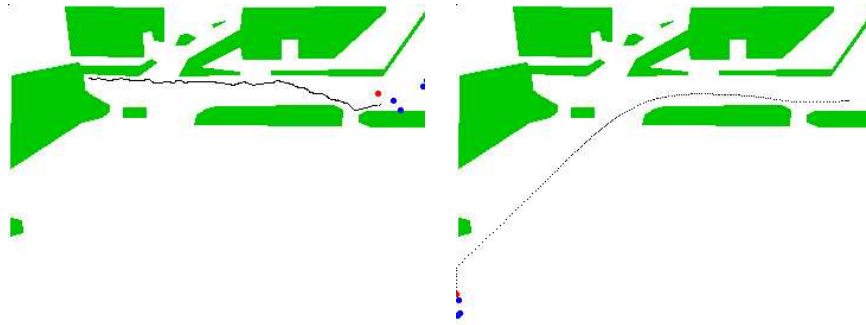


Figure 7.3: In some circumstances, there were many lowest-cost-goals

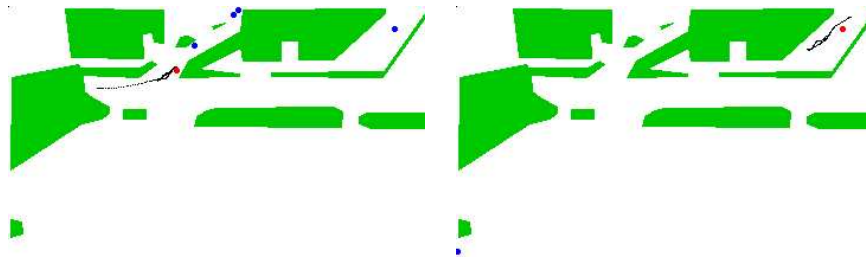


Figure 7.4: In a few circumstances (usually in cases where the trajectory was particularly complicated), the lowest cost goal is nowhere near the trajectory finish

One possible explanation for this is that those trajectories which are interesting are those where the goal-analysis does not match the behaviour (for example, trajectories such as those shown in Figure 7.4).

Person	Closest cost C_{pc2}		Lowest cost C_{pl2}	
	R_s	T_k	R_s	T_k
Mean Human (Car-park)	0.33	0.26	0.43	0.36
Mean Human (PETS)	0.67	0.5	0.74	0.56

Table 7.9: Correlation statistics comparing closest and lowest cost goal

Correlation results comparing the effect of choosing the closest or the lowest cost goal with each of the individual human rankers are shown in Appendix A, in Table A.4 for the car-park dataset and Table A.11 for the PETS2004 dataset.

7.6.3 The effect of limiting depth of search

Due to computational limitations, the bitmapped implementation of this algorithm stops analysis at 2 levels of sub-goal analysis. The polygonal model continues until there are no more areas of scene accessible by sub-goals, and hence could continue indefinitely. The actual depth to which an uncapped sub-goal search descends is very much dependant upon the layout of the scene in question. In order to investigate the effect of a cap on sub-goal depth, C scores have been calculated capping the search at various depths of sub-goal analysis. The results of correlating these scores with the human mean ratings are shown in Table 7.10.

Capped at...	1	2	3	4	5	6	7
r_s Car-park	0.44	0.43	0.42	0.42	0.42	0.42	0.42
T_k Car-park	0.36	0.35	0.35	0.35	0.35	0.34	0.34
r_s PETS2004	0.77	0.74	0.74	0.74	0.74	0.74	0.74
T_k PETS2004	0.61	0.57	0.57	0.57	0.57	0.57	0.57

Table 7.10: Human mean correlations with depth limited search: both datasets

It was expected that increasing the depth of search would provide increased levels of correlation. From Table 7.10 it is clear that this is not the case. Why is it the case that higher level sub-goals do not make a difference to the correlation results? One explanation hinges upon the nature of the ground truth – if it is the case that the human observers are predicting paths and determining the goal-directedness of the agents within the scene it

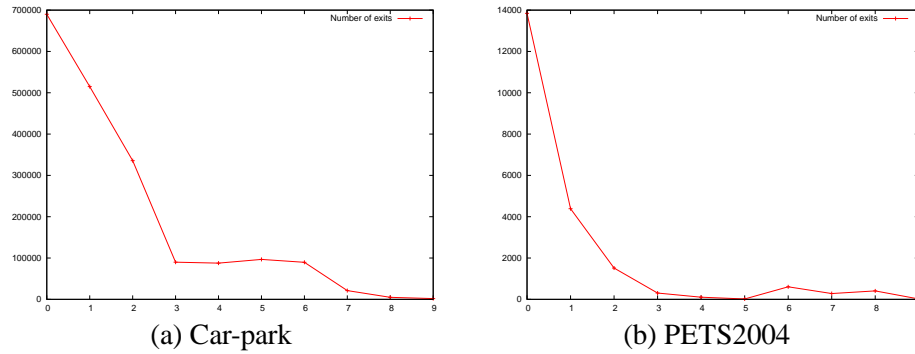


Figure 7.5: The number of goals classified as being at each level of sub-goal analysis during uncapped search. For the PETS2004 scene in particular, there are very few goals which are more than one or two sub-goals away.

becomes necessary to consider the level of look-ahead these human judges are doing. Another explanation for this unexpected result is that the trajectories being analysed are not actually very complicated. If the majority of trajectories can be explained in terms of one or two sub-goals, the addition of more complicated navigational hypotheses involving a large number of sub-goals could be confusing matters and keeping alive unrealistic paths through the scene.

Figure 7.5 shows the result of plotting the number of goals at each level of sub-goal analysis for each scene. This graph goes some way towards explaining why higher level sub-goals did not have a great effect, especially within the PETS2004 scene. The simple answer is that for most paths, higher level sub-goals were never hypothesised in the first place.

Figures 7.6 looks at a few trajectories in depth, showing an example frame with the sub-goal structure postulated for each agent capped at 1, 3, 5 and 7 levels of sub-goal. In these images, each successive level of sub-goal analysis is shown as a darker shade of grey, with sub-goals themselves in yellow and paths between sub-goals in green. The agent is shown as a red circle with a line indicating direction of travel, and the central point of known goals within the scene are shown as blue dots. Large blue dots represent geographical goals (the means from the Gaussian mixture model) and smaller blue dots represent parked cars. Areas directly visible to the agent are shown in white. The obstacle model in these illustrations is implicit – it is clear from this that projected paths radiate out from the agents' position to fill the scene.

Correlation results comparing the effect of capping search depth with each of the human rankers are given in Appendix A, in Tables A.5 and A.6 for the car-park dataset,

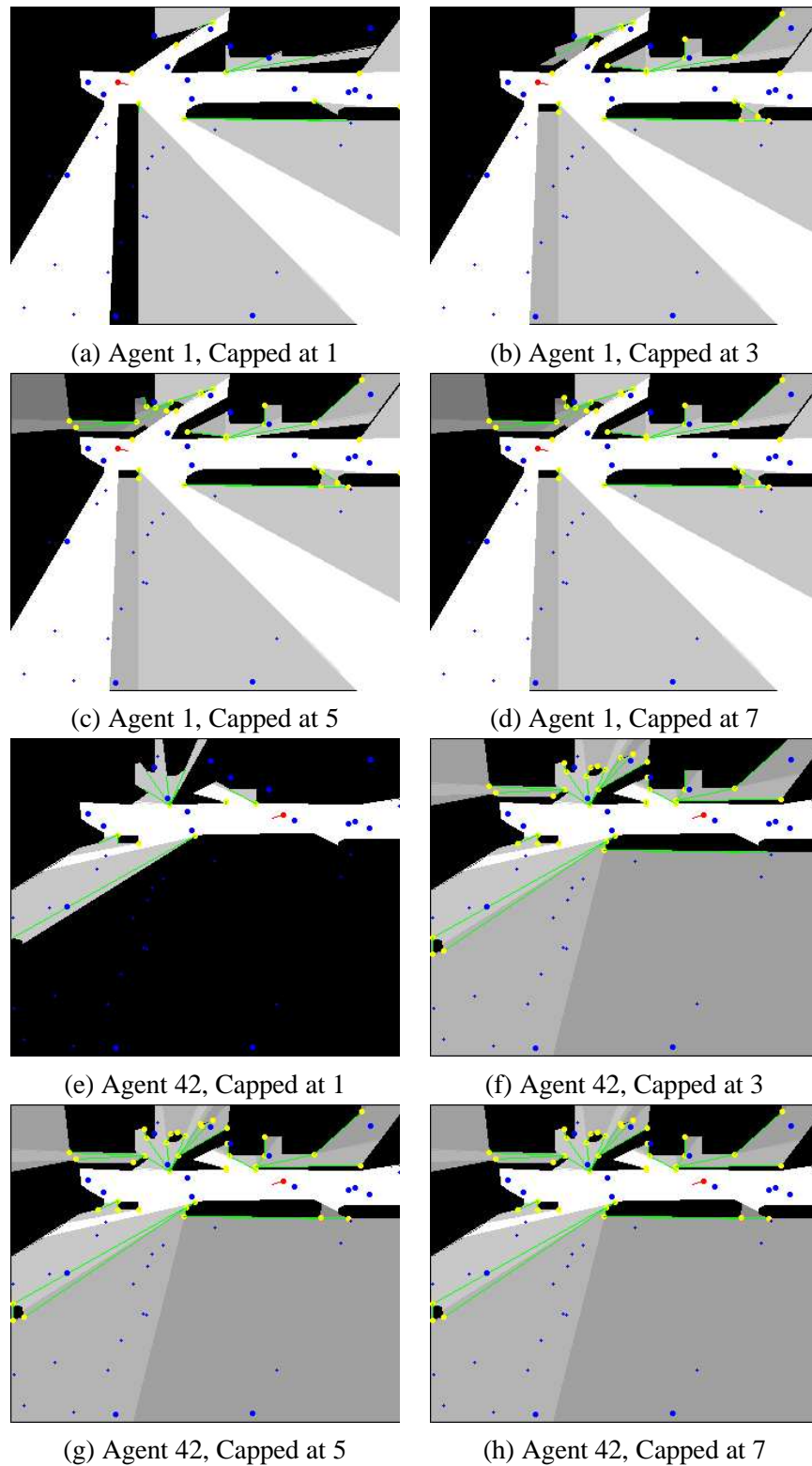


Figure 7.6: Example output from depth capped search showing that in many cases, higher level sub-goals do not add much to the analysis.

and Tables A.12 and A.13 for the PETS2004 dataset.

7.7 An evaluation of the exit model

The final comparison to be made in this chapter is between the learned exit model, described in depth in Chapter 3, the hand-crafted exit model created using knowledge of the scene, and no exit model at all. The *no exit model* condition still has to represent goals in some way, as the algorithms described in this thesis require some indication of goal location to function. So in the *no exit model* condition, “goals” were placed at even intervals around the scene, represented as points, and parked cars were not incorporated into the model as goals. This comparison is carried out in the car-park scene as there is no learned exit model for the PETS2004 scene. A graph of the correlation statistics (correlating with the human mean rating) is presented in Figures 7.7 for R_s and 7.8 for T_k . A table showing the figures upon which these graphs are based is presented in Appendix A at Table A.7.

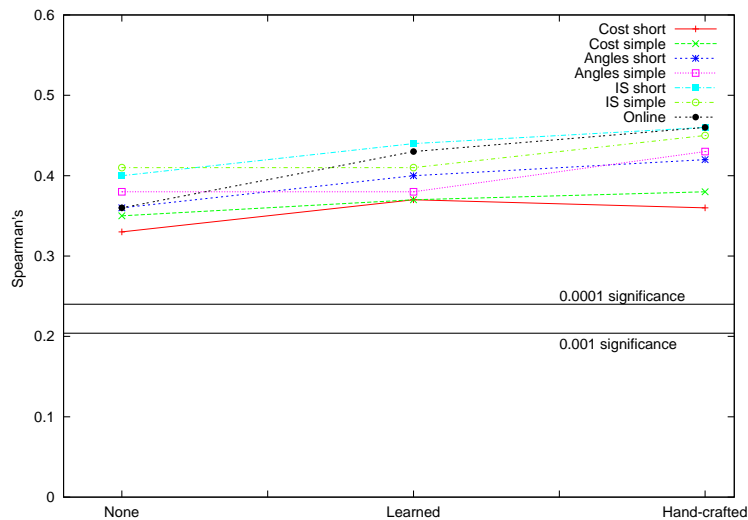


Figure 7.7: Graph showing Spearman’s Rho for evenly spaced exits, learned exits and hand-crafted exits

As expected, Figures 7.7 and 7.8 show that the exit model makes a noticeable difference to the level of correlation. Results are weakest with no exit model at all, although the system still produces correlation statistics significant at the 0.01% level.

The hypothesis that without an exit model (but with evenly spaced goals), these algorithms would still provide a measure of general intentionality has been supported. The *no exit model* algorithms still provide useful results from a surveillance perspective, correlating strongly with the human rankers. Whilst it does not perform quite as well as

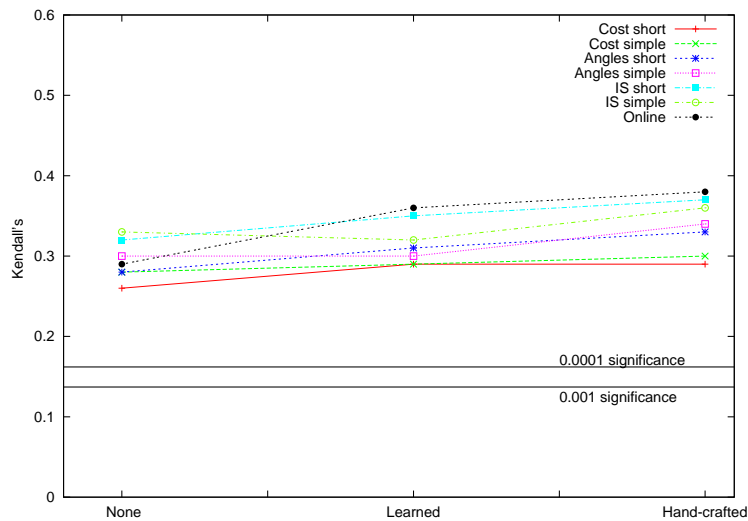


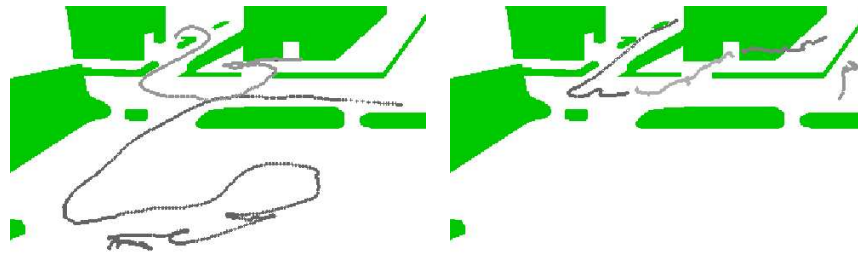
Figure 7.8: Graph showing Kendall's Tau for evenly spaced exits, learned exits and hand-crafted exits

the other systems, the no exit model implementation is considerably simpler than those which incorporate an exit model and a model of car location, as it has no need for object permanence or to keep track of parked cars. The hand-crafted exit model provides slightly stronger correlations with the human mean than the learned model with almost all measurements, using both R_s and T_k . From these results, it is possible to conclude that knowing *where* the goals in the scene lie helps determine intentional behaviour, but is not vital.

7.8 Consideration of high-variance cases

Due to the size of the car-park dataset, it is not practical to perform a qualitative evaluation of each agents' trajectory. However those agents which were subject to disagreement between human rankers or between the human rankers and the machine generated **C** statistic are worth investigating further. A selection of these are pictured in Figure 7.9. Figure 7.9(a) shows the trajectories with a high **C** statistic and high variance between human rankers: these trajectories feature vehicles parking in a rather roundabout fashion – it is clear from this picture alone that in neither case did the parking manoeuvre proceed smoothly. Figure 7.9(b) shows the opposite cases, where the **C** statistic was low but there was an amount of disagreement between rankers. These cases involved people using rarely used car-parks, or parking in rarely-used spaces, and in one case (the track to the far right of the image) an ambulance, which was not moving in an interesting or odd way

but was thought interesting just because it was an ambulance.



(a) High C , high variance

(b) Low C , high variance

Figure 7.9: Trajectories with a high level of disagreement between human and machine ranks from the car park dataset: (a) consists mainly of complicated parking manoeuvres and (b) of unusual areas of the scene

7.9 Concluding remarks

In this chapter a number of different variations on the theme of measuring intentionality have been compared against the performance of humans in a surveillance task. Various distance metrics applied to models of simplest and shortest path defined in Chapter 5 have been evaluated, along with various cost functions based on the online model from Chapter 6. These metrics are designed to measure the goal-directedness or intentionality of the agents within the scene, and it has been shown that they correlate strongly with human judgements of interestingness.

Within the car-park dataset, with its natural behaviours and mostly intentional activity, all models performed well in comparison to the humans (with all correlations with the mean and most of the individual correlations highly significant at the 0.0001% level). Figure 7.10 shows a summary of the correlations described in this chapter for the car-park dataset. Correlation results presented in this figure are for Kendall's Tau.

The behaviour found in the PETS2004 dataset was handled less well by these algorithms. In particular, the simplest and shortest path algorithms correlated relatively poorly with human performance. One possible explanation for this is in the nature of the PETS2004 dataset: the actions being performed involve a lot of changing of direction, and the simplest and shortest path algorithms imply a heavy penalty for this. The online algorithm, in contrast, penalises re-planning much less severely. This effect is clear

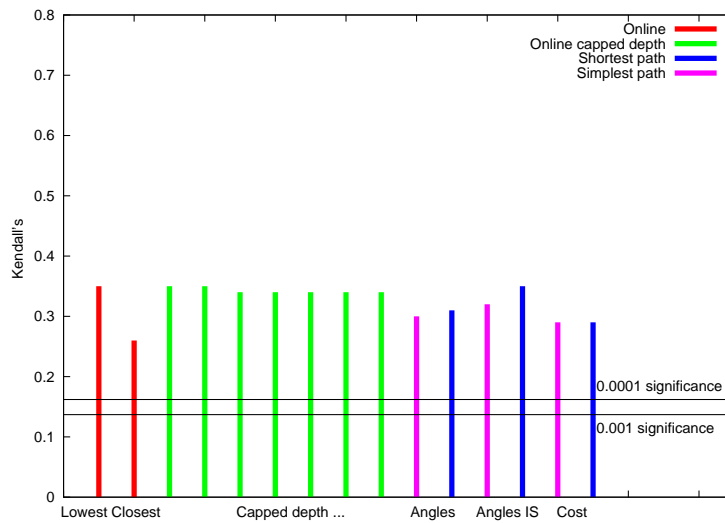


Figure 7.10: Correlation with the mean human result: overview of the car-park dataset

from Figure 7.11, a chart showing the performance of the various metrics against the human mean rank. As with Figure 7.10, correlation results presented in this table are for Kendall's Tau.

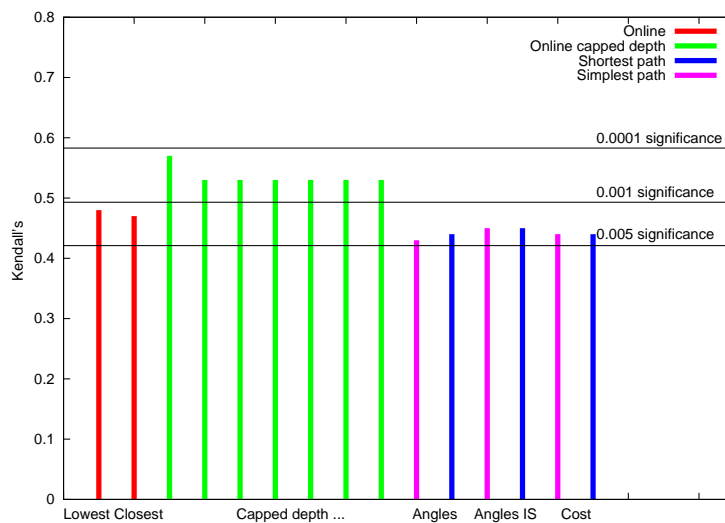


Figure 7.11: Correlation with the mean human result: overview of the PETS2004 dataset

Chapter 8

An example surveillance application

Chapter 7 detailed ways in which measurements of intentionality can be compared with human judgements about behaviour, and demonstrated a general pattern in which human ratings of interestingness correlate strongly with computer generated measurements of goal-directedness. This chapter describes a potential surveillance application based upon the measurements described in earlier chapters. This application provides a binary decision (interesting, or not interesting) by thresholding on the intentionality score of a trajectory or frame.

As described in Section 2.1, real-world surveillance installations typically involve one operative monitoring an unfeasibly large number of screens. Whilst it would be desirable to have a system that automatically rang an alarm when an unusual or interesting event occurred, the problem of repeated false alarms and the danger of missing an event mean that such a system is not within reach given current results. However, a pragmatic alternative presents itself. Instead of creating an *interesting behaviour detector*, this thesis proposes creating a *boring behaviour rejector*. By ignoring those trajectories or frames in which the behaviour of agents is explicable – those trajectories where the various scores presented earlier are very low – it is possible to cut down the number of frames of footage that an operative would have to inspect. This distinction is based upon the assumption that failing to draw attention to genuinely criminal behaviour is a much more costly error than that of accidentally drawing attention to behaviour which is not of interest. Within the surveillance domain, this is a common assumption (see, for example, [31]).

8.1 Choosing a threshold

The proposed application involves simple thresholding on one of the metrics presented earlier. Given the strong correlations between the various metrics and the ratings of the human volunteers, any of the metrics (Shortest path, simplest path or online) could be used to indicate explicability for trajectories as a whole. The online algorithm's C score has been chosen, as this score also allows for frame-by-frame measures.

Thresholding upon this score allows the removal of frames or trajectories in which people are behaving in a straightforwardly goal-directed fashion. The choice of threshold is determined with reference to the human ranks described in the previous chapter, that is, the decision about which trajectories the system *ought* to ignore is guided by the decisions of humans. If a suitably low threshold is set upon the human ranks (T_H) we will be left with those scenes in which absolutely nothing of interest is occurring. It is then possible to determine whether or not the C score can be used to automatically reject some proportion of these clearly dull behaviour patterns by thresholding again on C (T_C). The particular C score being used here is C_{pl} (polygonal model, lowest cost). There are two ways in which this thresholding can be performed:

1. *By trajectory*: is the simpler of the measures – we have trajectory-by-trajectory indications of both C and human opinion. This is less realistic than the second option as it fails to take into account situations where more than one person is in the scene.
2. *By frame*: is a more complicated measure, as it involves converting by-trajectory measures of cost and human rank into by-frame measures. This is achieved by taking the highest scoring trajectory per frame as a measure of intentionality for that frame. It is a more realistic approach, as within real surveillance situations filtering would need to be based upon whole scenes rather than individual trajectories.

The threshold chosen for T_H should be very low. The aim is to provide a filter which will remove a proportion of completely uninteresting footage whilst retaining as much as possible – preferably all – of the interesting footage. Examination of ROC curves for various values of T_H can help guide the choice of threshold, and these curves are shown in Figure 8.1.

ROC curves show true positive rate plotted against false positive rate. In this application, the ROC curve is being used to determine values for a filter on surveillance data. As the aim is to keep as many of the interesting trajectories as possible whilst rejecting those

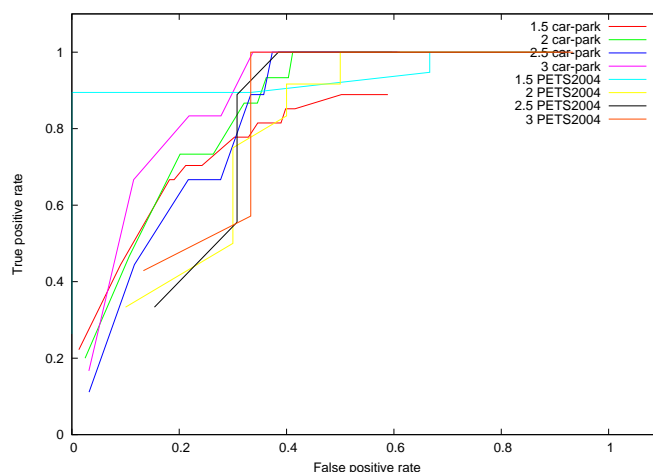


Figure 8.1: ROC curves for various values of T_H . Thresholds should be selected which maximise the true positive rate.

which are uninteresting, true positive rate is defined as the number of interesting trajectories kept which are actually interesting, divided by the the total number of interesting trajectories. (those over T_C and also over T_H , divided by the total number of trajectories over T_H .) False positive rate is the opposite of this – that is, the number of those under T_H and also over T_C which have erroneously been kept, divided by the total number under T_H . The curves shown in Figure 8.1 are generated by keeping T_H constant for each curve and varying T_C from a value of 0.001 (preserving nearly all the trajectories - the points towards the top right of the graph) to 0.3 (rejecting nearly all - the points towards the bottom left).

Usually, when examining ROC curves the indication of a good discriminator is a point towards the top-left of the curve, which has the effect of maximising true positives and minimising false positives. However in this application it is more important to maximise the true positives, as it would be much less of a problem to preserve uninteresting footage than it would be to accidentally filter out genuinely interesting or problematic behaviour.

Two further factors can be used to influence the choice of T_H . One of these is investigation of the human ranks, and the other is consideration of the comments made by the volunteers. Taking these three factors into consideration, the value of 2 was chosen. This value provides good discrimination in both datasets, and excludes most of the trajectories considered interesting purely because they consisted of people walking in unusual places within the scene.

Setting T_H at 2 provides us with 15 trajectories within the car-park dataset which are considered to be *interesting* and 243 which we would wish to filter out. Within the

PETS2004 dataset the same threshold has 12 trajectories which we would wish to keep and 10 which we should ignore¹. The charts shown in Figure 8.2 provide a more in-depth illustration of some of the information shown in the earlier ROC chart Figure 8.1, with a value of 2 for T_H and values of T_C varying between 0.07 and 0.1.

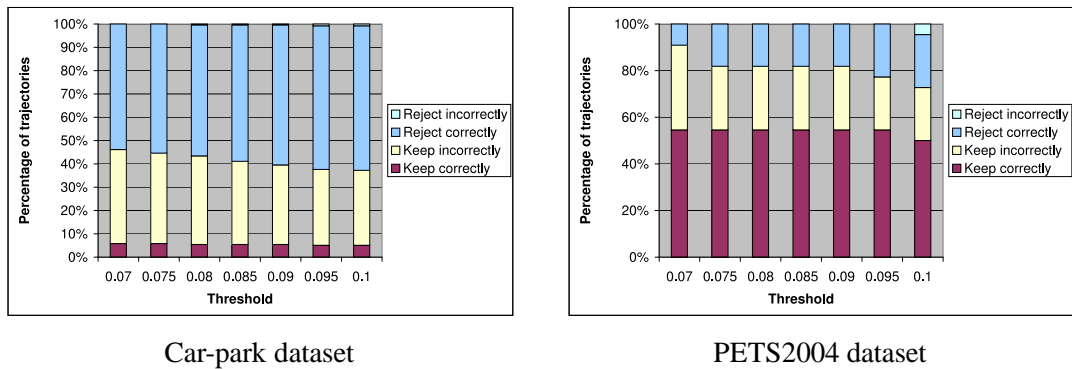
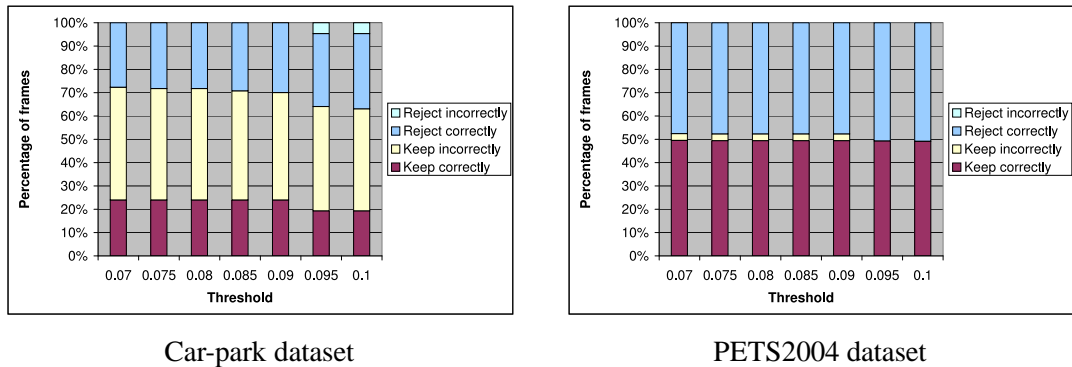


Figure 8.2: The effect of thresholding by trajectory T_C

From the charts shown in Figure 8.2 it is clear that filtering on C scores as suggested would preserve a number of uninteresting trajectories as well as those which are considered interesting – it would not be a good deal of use as an *interesting behaviour detector* as the number of false positives is high. However, given the stated aim of creating a *boring behaviour rejector* instead the results are much more promising. It is important to reject a significant proportion of uninteresting trajectories whilst rejecting *no* interesting trajectories by mistake, and this appears to be possible. From the charts, a threshold of around 0.09 seems to be the most effective, rejecting none of the interesting trajectories within the PETS2004 dataset, and rejecting only one (mildly) interesting trajectory from the car-park dataset (which upon inspection turns out to be a person using an unusual shortcut).

The results for thresholding on a frame-by-frame basis (shown in Figure 8.3) rather than upon entire trajectories show that a smaller proportion of the footage would be rejected. One possible factor in this is that those trajectories which are considered to be *interesting* tend to be longer, and hence tend to influence the C scores of more frames. Agents who cross the scene purposefully heading directly to their goal do not take long to cross the field of view of the camera. The charts shown in Figure 8.3 support our earlier suggestion of a threshold on C of around 0.09. It is worth noting here that these statistics are for frames in which there were actually moving agents, and the 10% or so of empty

¹The large difference in proportion of interesting behaviours between the datasets is due to the slightly contrived nature of the PETS2004 dataset



Car-park dataset

PETS2004 dataset

Figure 8.3: The effect of thresholding by frame

frames would obviously be rejected as well.

8.2 Concluding remarks

This chapter has briefly described a practical application based upon the algorithms presented in earlier chapters. By ignoring the most explicable trajectories, it is possible to filter out around 60% of behaviour patterns from the car-park scene.

Chapter 9

Conclusions

9.1 Summary

This thesis has presented a novel approach to the problem of behaviour modelling, with specific application to the surveillance domain. Previous work, as outlined in Chapter 2 has either concentrated upon modelling a scene (e.g., [93, 98]), working out statistically where people have previously walked (e.g., [78, 139]) or has involved *a priori* ideas of what constitutes unusual behaviour (e.g., [71, 117]). Unlike this previous work, the algorithms presented here in Chapters 4, 5 and 6 aim to model human behaviour at the level of individual psychology: the level of goals and intentions. Chapter 4 described the construction of a scene representation making explicit the relationships between an agent within the scene and possible known goals. Chapter 5 built upon this representation and described two alternative ways of navigating through the scene (inspired by work from within Psychology into human navigation) and also described a number of ways of measuring how well a particular trajectory matches this type of model. Chapter 6 described a different way of measuring intentionality, also based upon the representation from Chapter 4, in which changing patterns of activity at known goal sites are used to determine which goals are consistent with the behaviour of the agent. Chapter 7 described a novel way of evaluating event detection systems for surveillance, enabling the creation of a form of *ground truth* for such systems, and then uses this new evaluation criterion to judge the systems described in earlier chapters. Finally, Chapter 8 provides a brief outline of a

way in which these systems could be brought together to provide a practical surveillance application.

This thesis has shown that by attributing *goals* to people within surveillance situations, their behaviour can be *explained* automatically. The quality of these explanations can be measured in a number of ways: either by incurring a cost for each frame in which the goal is inconsistent with the agent's behaviour, or by comparing the agent's actual trajectory with some ideal path using various distance metrics. These assessments of explanation quality have been shown to correlate strongly with human judgements of the *interestingness* of the agent's behaviour, suggesting that they would be of use in a surveillance application.

9.2 Discussion

The initial insight upon which this thesis is based is that when engaged in a surveillance task, we try to explain what the people in the scene are doing. If we can come up with an explanation ("He's going to that car over there", for example, or "She's going round the hedge so she can get to that exit") then we can ignore the behaviour. It is only when the behaviour is *inexplicable* that our interest is piqued. The thesis then proceeded with the job of automatically determining some simplistic explanation in terms of known goal sites within the scene, under the assumption that for this type of simple goal attribution, folk psychology and the intentional stance usually work.

Philosophical problems with intentionality – the assumption of rationality, for example – do not seem to have caused any problems for the very simple application of goal-directed reasoning used in this thesis. The attribution of geographical goals to the agents, and the measurement of how well those goals actually *explain* the agents behaviour does not require any tricky or controversial belief attribution. It also does not require that the agents be perfectly rational. Indeed, the comparison of simplest path with shortest path in Chapter 5 is in a sense a comparison between an irrational and a rational model of human navigation.

Nevertheless, it could be the case that the agents under consideration may not be behaving rationally at all. In familiar situations, people might not engage in intentional reasoning about their navigational strategies and instead behave habitually. Some of the people observed going about their daily business in the car-park scene might tread the same path every day without thinking about their route at all. Habitual behaviours, however, grow out of a history of learning. Whilst the present instance of navigation might be thoughtless and habitual, it can be argued that when the agent in question was first

navigating around the university campus in question they *did* have to consider how they were to get from A to B. This initial reasoning might have been based on any number of strategies – simplest path, shortest path, fewest hills, prettiest buildings. . . They may even have been shown the way by a colleague: but at some point someone will have thought about which route to take from a starting point to a goal position, and that reasoning will have been intentional.

In Chapter 7 measures of goal-directedness are compared to the performance of human volunteers. There are two assumptions made in this chapter: firstly, that naïve observers are as good at such tasks as trained operators, and secondly, that the judgements of human beings ranking behaviour for *interestingness* is the sort of thing we should want computerised surveillance systems to correlate with. The first of these assumptions has been dealt with in depth in the chapter, but the second bears more discussion. Whose intentionality is being modelled here anyway? When we are engaged in surveillance activity, we try to explain the agent's behaviour in terms of known goals. This can be characterised as a straightforward case of adopting intentional reasoning towards the people moving around the car-park. However, it can also be accounted for by simulation theorists, and the person engaged in surveillance isn't actually doing intentional reasoning as described by Dennett [35] but putting themselves in the place of the agent. It could be that the correlations are high through having built an accurate model not of the behaviour of the agent, but of the behaviour of some idealised surveillance operative? It could be argued that the system models the goal attributions of the watchers, rather than the goals of the agents themselves.

9.3 Future work and possible extensions

The work described in this thesis could be extended in a number of directions. There are some obvious enhancements which could improve the performance, such as enhancing the exit model, enabling the incorporation of cars as obstacles, and including a more sophisticated account of object permanence. Working within a larger scene area would allow more interesting experiments about intentionality to be conducted.

9.3.1 Modelling the scene: learning and extending

A richer scene model could enhance the explanatory depth of this approach. As mentioned in Section 3.2, a side effect of using the velocity component of the Kalman filter as an indication of agents' direction of travel is that in the absence of data this measurement

varies wildly due to the noise estimates incorporated into the filter. Therefore, the goals associated with agents who stop change a great deal. In the scenes described here, this effect is desirable as there are few locations in the scene where agents might want to stop. But in different scenes this may not be the case. Inactivity zones could be identified (such as those described by McKenna and Nait Charif in [98,99,104]) at places such as benches, cash machines or any other location where people are known to linger. Such places could also be incorporated into the goal model, perhaps with temporal characteristics. For example, people approach a cash machine (it is a legitimate goal), are still for a length of time, and then move off. Similar patterns of activity occur at park benches (although the time constraint would be different).

The learning of an obstacle model is another interesting line of research to pursue: simply by looking at the patterns of trajectories through an unknown scene it is possible for humans to identify obstacles from tracker output. See, for example, Figure 9.1, which features one hour's worth of tracks from a pedestrianised area. In this image, it is clear that there are areas of scene which are effectively obstacles. It should be possible to develop algorithms which can automatically detect such areas, given enough observational data.



Figure 9.1: Tracks and scene from a pedestrian area: the location of the obstacles could be inferred from the tracks alone, although there are areas of scene which are not obstacle but are still fairly empty.

Sumpter, in [145], describes an experiment using simulated data in which paths through a maze are learned over time. By training a neural network on legitimate paths through a maze, it can be thought of as inferring the existence of the maze walls as obstacles [145] p. 86. The neural network can be used to predict paths from a particular location, and as the learned behaviour patterns do not include any examples of walking through walls, the output predictions do not either. There is a sense in which Johnson and Hogg's system [75, 78] learns the location of obstacles in a similar way, in that the learned patterns

through behaviour space do not cross obstacles, and hence predicted behaviours will not cross obstacles. Within these sort of approaches, obstacle location is implicit. For an accurate “obstacle model” the burden falls on the nature of the training set: it must contain examples of all possible paths, and must not contain examples of any impossible paths.

The algorithms described in this thesis have the useful quality of handling trajectories which are unusual: routes through the scene which are unpopular, and hence do not appear often, are not penalised because of their novelty. The obstacle model plays a vital role in this, as without an obstacle model it would be impossible to talk about the way people really navigate. Indeed, within this thesis, the positioning of sub-goals has been determined by the obstacle model alone. It has been assumed that tangential points on obstacles are the places where people choose to change direction. This assumption is a useful fiction, and certainly an oversimplification.

In Section 5.3, it was noted that for some agents the difference between simplest and shortest path was due to the shape of the obstacle in the centre of the car-park scene - the hedge. Figure 9.2 shows another example of a case where this effect can be observed.

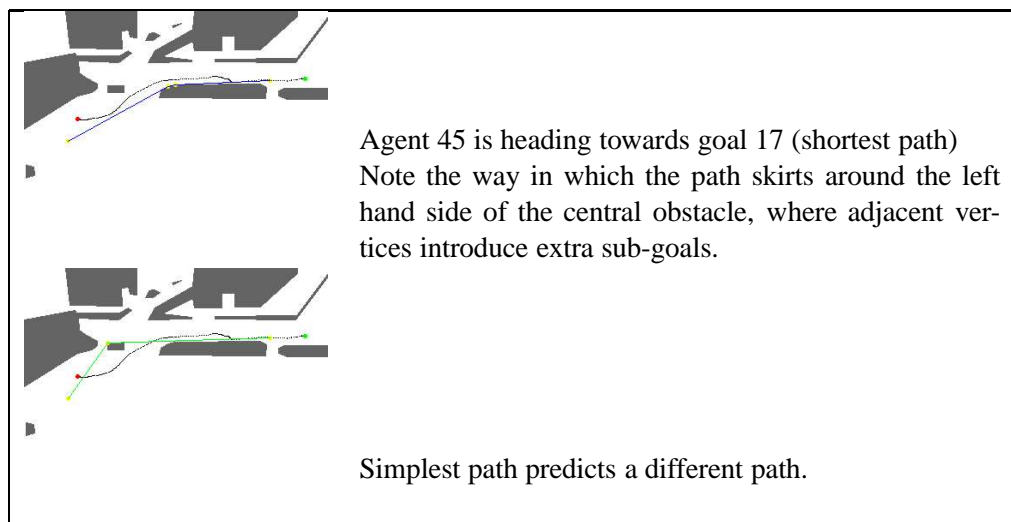


Figure 9.2: An example of an agent where path difference is due to an artifact of the obstacle model

Possible solutions to this problem involve either reformulating the obstacle model (in such a way as to ensure that turning points were marked as such) or reformulating the way in which sub-goals are constructed. It would be possible, for example, to count multiple adjacent sub-goals as just one sub-goal.

Another way to approach this issue is to reconsider the placement of sub-goals completely. In Section 5.3, a means of partitioning the trajectory into the same number of

segments as its corresponding ideal path was presented. Taking the sub-goal locations as determined by the trajectory partitioning provides an indication of where the agent's trajectories *actually* change direction: in a sense, where the sub-goals *actually are*. It could be possible to use something like this to determine the actual location of sub-goals, which would in turn provide a better model of the way people actually navigate through a scene.

Plotting these points results in the image shown in Figure 9.3, showing where the trajectory partitioning algorithm placed the sub-goals. Whilst this shows an indication of where people turned, it does not provide a means for determining real turning points in the absence of some indication of sub-goal location. The trajectory partitioning algorithm itself requires knowledge of the ideal path and how many segments that path contains, which in turn requires the sub-goal locations. However, the learning of sub-goals should be possible, and perhaps some form of iterative solution, starting with tangential points on obstacles and gradually moving towards real turning points would be successful.

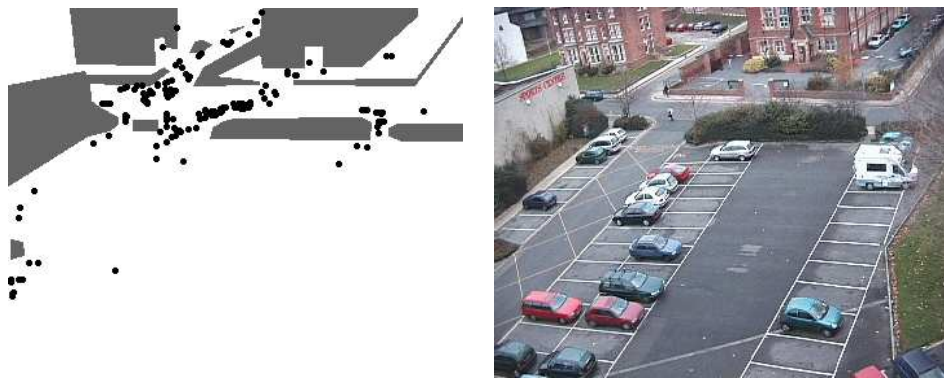


Figure 9.3: Locations of sub-goals determined by path partitioning, shown next to the car-park scene.

The trajectory partitioning procedure consists of minimising a function with two parts (angular disparity, and relative proportion of trajectory length). The relative influence of these parts is governed by a constant λ . Sub-goals will obviously fall on trajectories and by varying λ they can be moved up and down the length of an agent's route through the scene. This information could be used to derive a probability density function, and over time, the positions of the sub-goals could be learned from trajectory data. Such an approach would lead to sub-goals being defined as places in the scene where agents choose to change direction, rather than as points on obstacles where an agent might choose to change direction.

Learning the location of sub-goals in this fashion would have its drawbacks. Firstly, the training set would need to contain enough examples of behaviour to capture the loca-

tion of all possible sub-goals within the scene. With the current approach this is not the case, therefore novel trajectories and unusual routes are handled with ease even though the dataset by definition does not contain many examples of these behaviour patterns. A learned model of sub-goal location would also exclude the use of moving obstacles: whilst the current implementations do not incorporate a changing obstacle model they do leave open the possibility of implementing such a thing.

9.3.2 Working within a larger scene area

As the algorithms in this thesis described work on positional information alone, it would be possible to implement the work described here coupled to any multi-camera system capable of providing frame-by-frame object location in space (such as that described by Black et al in [13]). With a multi-camera system a larger area can be investigated as it is possible to track individuals across cameras, in this way investigating the application of the intentional algorithms in a larger scene.

In particular, there are certain hypotheses about human navigation which have been investigated by psychologists using map-based studies e.g. [51, 52], and virtual environment studies e.g. [29, 72]. The question of whether features in a scene contribute to perceived length of path has not been investigable within the current thesis, as the number of “landmarks” or noticeable features in the scenes under consideration is small. It might be possible, within a larger, more complicated scene, to determine if people’s choice of shortest or simplest path were influenced by the number of features along the route, although the difficulty of defining what makes a feature a landmark could still prove problematic.

The effect of trip-chaining on route selection has been impossible to investigate as there is no indication of where the agents have been before they enter the scene, and no way of knowing where they go when they leave. With a large area surveillance system it ought to be possible to carry out more naturalistic studies on the way in which people *really* navigate when they have three different shops to visit.

Within a different (but not necessarily larger) scene it would be possible to perform a direct comparison between this work and work based upon machine learning techniques such as that of Johnson and Hogg [78]. The scenes described in this thesis do not provide sufficient training data for such a comparison to be performed: the PETS2004 scene does not feature a large enough number of people; and the car-park scene is too unconstrained (with its wide open space and moving obstacles). Indeed, one of the strengths of the approach described in this thesis is that it can handle such unconstrained scenes.

9.3.3 Other possible extensions

The potential for using an intentional analysis of behaviour for prediction or generation has not been exploited in this thesis. Synthesised agents have been produced from video using various statistical methods including the auto regressive process [23], Gaussian mixture models [76], variable length Markov models [48, 75], and Markov chains [38]. The advantages of basing some form of behaviour simulation upon an intentional analysis would be the ability to generate plausible behaviours which have never been seen before. Current techniques for behaviour generation and synthesis are able to produce plausible paths through behaviour space from observed behaviour to observed behaviour (e.g., come in through the door, approach the television, and then sit on the sofa). If there happened to be a box of chocolates in a different part of the room, a system capable of generating behaviour based upon intentions could produce completely novel and completely plausible trajectories that simply would not occur in a model based upon learned patterns of motion alone.

Within the current thesis no effort has been made to distinguish between pedestrians and cars, and the two classes of agent have been treated as identical. This is a reasonable simplification to make as the behaviour of a car is usually intentional (as intentional as the behaviour of its driver). However there are a few instances where this simplification causes problems, as some of the gaps between obstacles are too small for a car to pass through. These are occasionally predicted as simplest or shortest paths and the subsequent analysis shows the agent isn't following the best path through the scene. Distinguishing between people and cars and then incorporating a size constraint could get around this problem. There is a number of techniques for telling the difference between cars and pedestrians, ranging from the approach of using a separate (model based) tracker for each (as they do in [113]), to using simple blob based measures such as dispersedness (people are usually taller and thinner, so have a higher perimeter² to area ratio [85]).

The speeds of the agents within the scene are not referred to at all by the algorithms outlined in this thesis. The inclusion of inactivity zones, within which the speed of the agents could drop to zero, has already been mentioned as a possible extension. Noting changes in speed could also be a useful additional feature. Considering the hypothetical case of a person running from the scene of a crime, it would be possible for an agent's trajectory to be perfectly goal directed but still interesting or unusual, but in the temporal domain rather than the spatial domain. This sort of behaviour is still goal-directed, but the goal is to *leave the scene quickly*. Differentiating this sort of behaviour from the behaviours already modelled would be a useful extension, and could be dealt with alongside the modelling of inactivity zones as both problems are associated with changes in

velocity.

Appendix A

Correlation results with individual humans

In Chapter 7, correlations with the mean score were presented. This appendix contains correlation results with individual human rankers for completeness. To maintain consistency with the convention used in Chapter 7, results which are significant at the 0.001 level (0.1%) are shown in **boldface**, and those which are significant to the 0.01 (1%) level but not to the 0.001 level in *italics*.

A.1 The car-park dataset

The first set of tables compares shortest path and simplest path, using angular disparity, angular disparity ignoring small angles and cost function (all as defined in Chapter 5). Spearman's Rho (R_s) is in Table A.1, Kendall's Tau (T_K) is in Table A.2. Correlation statistics comparing polygonal and bitmapped implementations are shown in Table A.3. Statistics comparing closest and lowest cost goal are in Table A.4. Statistics demonstrating the effect of limiting depth of search are in Table A.5 for R_s and A.6 for T_K .

	Angular disparity Shortest	Angular disparity Simplest	Ignoring small angles Shortest	Ignoring small angles Simplest	Cost Shortest	Cost Simplest
1	0.32	0.31	0.32	0.32	0.27	0.28
2	0.39	0.39	0.39	0.38	0.36	0.35
3	0.33	0.32	0.34	0.33	0.29	0.29
4	0.15	0.15	0.15	0.15	0.15	0.16
5	0.32	0.31	0.33	0.32	0.28	0.28
6	0.32	0.31	0.34	0.32	0.28	0.29
7	0.3	0.29	0.33	0.31	0.26	0.27
Mean Human	0.4	0.38	0.44	0.4	0.37	0.37

Table A.1: Individual correlations: Carpark dataset, shortest vs. simplest path, R_s

	Angular disparity Shortest	Angular disparity Simplest	Ignoring small angles Shortest	Ignoring small angles Simplest	Cost Shortest	Cost Simplest
1	0.26	0.26	0.27	0.26	0.22	0.22
2	0.32	0.31	0.32	0.31	0.29	0.29
3	0.27	0.26	0.27	0.27	0.23	0.24
4	0.12	0.12	0.12	0.13	0.12	0.13
5	0.25	0.25	0.27	0.26	0.22	0.23
6	0.26	0.25	0.28	0.26	0.23	0.23
7	0.24	0.23	0.27	0.25	0.21	0.21
Mean Human	0.31	0.3	0.35	0.32	0.29	0.29

Table A.2: Individual correlations: Carpark dataset, shortest vs. simplest path, T_K

Car-park Subject	Bitmapped model C_{b12}		Polygonal model C_{p12}	
	Spearman's	Kendall's	Spearman's	Kendall's
1	<i>0.23</i>	0.19	0.28	0.23
2	0.30	0.25	0.30	0.25
3	0.26	0.21	0.28	0.23
4	0.13	0.11	0.17	<i>0.17</i>
5	0.32	0.26	0.36	0.30
6	0.29	0.24	0.34	0.29
7	0.37	0.30	0.40	0.33
Mean Human	0.40	0.32	0.43	0.36

Table A.3: Correlation statistics for the car park dataset comparing polygonal and bitmapped implementations

Person	Closest cost C_{pc2}		Lowest cost C_{pl2}	
	R_s	T_k	R_s	T_k
1	0.16	0.13	0.28	0.23
2	0.3	0.24	0.30	0.25
3	0.21	0.17	0.28	0.23
4	0.041	0.033	0.17	0.17
5	0.22	0.18	0.36	0.30
6	0.22	0.18	0.34	0.29
7	0.29	0.23	0.40	0.33
Mean Human	0.33	0.26	0.43	0.36

Table A.4: Correlation statistics for the car-park dataset comparing closest and lowest cost goal

Capped at...	1	2	3	4	5	6	7
1	0.25	0.27	0.26	0.26	0.26	0.25	0.25
2	0.32	0.29	0.28	0.28	0.28	0.28	0.28
3	0.27	0.28	0.27	0.27	0.27	0.26	0.26
4	0.15	0.16	0.14	0.14	0.14	0.13	0.13
5	0.32	0.35	0.33	0.33	0.33	0.33	0.33
6	0.31	0.33	0.32	0.32	0.32	0.31	0.31
7	0.39	0.4	0.39	0.39	0.39	0.38	0.38
Mean	0.43	0.43	0.42	0.42	0.42	0.41	0.41

Table A.5: Correlation statistics for the car-park dataset comparing depth limited search: R_s

Capped at...	1	2	3	4	5	6	7
1	0.21	0.23	0.22	0.22	0.21	0.21	0.21
2	0.26	0.24	0.24	0.24	0.23	0.23	0.23
3	0.22	0.23	0.22	0.22	0.22	0.22	0.22
4	0.12	0.13	0.12	0.12	0.12	0.11	0.11
5	0.27	0.29	0.28	0.28	0.28	0.27	0.27
6	0.26	0.27	0.26	0.26	0.26	0.26	0.26
7	0.33	0.33	0.32	0.32	0.32	0.32	0.32
Mean	0.35	0.35	0.34	0.34	0.34	0.34	0.34

Table A.6: Correlation statistics for the car-park dataset comparing depth limited search: T_K

Statistic	No exit model	Learned exit model	Hand crafted exit model
Online model R_s	0.36	0.43	0.46
Online model T_k	0.29	0.36	0.38
Cost, Short R_s	0.33	0.37	0.36
Cost, Short T_k	0.26	0.29	0.29
Cost, Simple R_s	0.35	0.37	0.38
Cost, Simple T_k	0.28	0.29	0.3
Angles, Short R_s	0.36	0.4	0.42
Angles, Short T_k	0.28	0.31	0.33
Angles, Simple R_s	0.38	0.38	0.43
Angles, Simple T_k	0.3	0.3	0.34
IS, Short R_s	0.4	0.44	0.46
IS, Short T_k	0.32	0.35	0.37
IS, Simple R_s	0.41	0.41	0.45
IS, Simple T_k	0.33	0.32	0.36

Table A.7: A comparison of regularly-spaced, learned and hand-crafted exit models within the carpark dataset: Correlations with the human mean

A.2 The PETS2004 dataset

The first set of tables compares shortest path and simplest path, using angular disparity, angular disparity ignoring small angles and cost function (all as defined in Chapter 5). Spearman’s Rho (R_s) is in Table A.8, Kendall’s Tau (T_K) is in Table A.9. Correlation statistics comparing polygonal and bitmapped implementations are shown in Table A.10. Statistics comparing closest and lowest cost goal are in Table A.11. Statistics demonstrating the effect of limiting depth of search are in Table A.12 for R_s and A.13 for T_K .

	Angular disparity Shortest	Angular disparity Simplest	Ignoring small angles Shortest	Ignoring small angles Simplest	Cost Shortest	Cost Simplest
1	0.6	0.6	0.62	0.62	0.6	0.6
2	0.43	0.42	0.46	0.45	0.42	0.42
3	0.33	0.34	0.33	0.34	0.33	0.33
4	<i>0.69</i>	<i>0.69</i>	0.72	0.72	<i>0.69</i>	<i>0.69</i>
5	0.3	0.31	0.34	0.35	0.3	0.3
6	0.3	0.28	0.3	0.29	0.29	0.29
7	0.28	0.28	0.31	0.31	0.28	0.28
8	0.29	0.28	0.29	0.27	0.29	0.29
9	<i>0.66</i>	<i>0.66</i>	<i>0.69</i>	<i>0.68</i>	<i>0.66</i>	<i>0.66</i>
10	0.47	0.47	0.49	0.49	0.46	0.46
11	0.63	0.63	<i>0.66</i>	<i>0.66</i>	0.63	0.63
12	0.43	0.44	0.45	0.46	0.42	0.42

Table A.8: Correlations between shortest and simplest path metrics and the human rankers, R_s , PETS2004 dataset

	Angular disparity Shortest	Angular disparity Simplest	Ignoring small angles Shortest	Ignoring small angles Simplest	Cost Shortest	Cost Simplest
1	0.48	0.46	<i>0.5</i>	0.48	0.47	0.47
2	0.32	0.3	0.35	0.33	0.31	0.31
3	0.29	0.29	0.28	0.29	0.28	0.28
4	<i>0.54</i>	<i>0.54</i>	<i>0.56</i>	<i>0.56</i>	<i>0.54</i>	<i>0.54</i>
5	0.22	0.23	0.24	0.24	0.22	0.22
6	0.23	0.22	0.23	0.22	0.23	0.23
7	0.23	0.23	0.26	0.26	0.23	0.23
8	0.23	0.22	0.23	0.22	0.23	0.23
9	<i>0.5</i>	0.49	<i>0.52</i>	<i>0.51</i>	<i>0.5</i>	<i>0.5</i>
10	0.33	0.33	0.34	0.34	0.32	0.32
11	0.45	0.45	0.47	0.47	0.45	0.45
12	0.35	0.36	0.37	0.38	0.35	0.35

Table A.9: Correlations between shortest and simplest path metrics and the human rankers, R_s , PETS2004 dataset

PETS2004 Subject	Bitmapped model C_{bl2}		Polygonal model C_{pl2}	
	Spearman's	Kendall's	Spearman's	Kendall's
1	0.64	0.53	0.73	0.58
2	0.69	0.55	0.71	0.59
3	0.46	0.36	0.52	0.41
4	0.41	0.34	0.68	0.56
5	0.56	0.45	0.64	0.52
6	0.54	0.41	0.61	0.47
7	0.38	0.32	0.51	0.43
8	0.37	0.26	0.51	0.38
9	0.43	0.37	0.66	0.55
10	0.37	0.29	0.54	0.43
11	0.50	0.45	0.69	0.56
12	0.64	0.55	0.67	0.55
Mean Human	0.63	0.48	0.74	0.56

Table A.10: Correlation statistics for the PETS2004 dataset comparing polygonal and bitmapped implementations

PETS2004 Person	Closest Cost C_{pc2}		Lowest Cost C_{pl2}	
	R_s	T_k	R_s	T_k
1	0.67	0.53	0.73	0.58
2	0.58	0.44	0.71	0.59
3	0.37	0.28	0.52	0.41
4	0.68	0.55	0.68	0.56
5	0.47	0.37	0.64	0.52
6	0.47	0.36	0.61	0.47
7	0.28	0.23	0.51	0.43
8	0.42	0.33	0.51	0.38
9	0.63	0.49	0.66	0.55
10	0.55	0.42	0.54	0.43
11	0.58	0.45	0.69	0.56
12	0.62	0.5	0.67	0.55
Mean	0.67	0.5	0.74	0.56

Table A.11: Correlation statistics for the PETS2004 dataset comparing closest and lowest cost goal

Capped at...	1	2	3	4	5	6	7
1	0.77	0.72	0.72	0.72	0.72	0.72	0.72
2	0.69	0.6	0.6	0.6	0.6	0.6	0.6
3	0.53	0.54	0.54	0.54	0.54	0.54	0.54
4	0.75	0.67	0.67	0.67	0.67	0.67	0.67
5	0.68	0.68	0.68	0.68	0.68	0.68	0.68
6	0.7	0.7	0.7	0.7	0.7	0.7	0.7
7	0.55	0.47	0.47	0.47	0.47	0.47	0.47
8	0.47	0.44	0.44	0.44	0.44	0.44	0.44
9	0.77	0.71	0.71	0.71	0.71	0.71	0.71
10	0.63	0.6	0.6	0.6	0.6	0.6	0.6
11	0.64	0.58	0.58	0.58	0.58	0.58	0.58
12	0.67	0.67	0.67	0.67	0.67	0.67	0.67
Mean	0.77	0.74	0.74	0.74	0.74	0.74	0.74

Table A.12: Correlation statistics for the PETS2004 dataset comparing depth limited search: R_s

Capped at...	1	2	3	4	5	6	7
1	0.64	0.58	0.58	0.58	0.58	0.58	0.58
2	0.55	0.48	0.48	0.48	0.48	0.48	0.48
3	0.43	0.42	0.42	0.42	0.42	0.42	0.42
4	0.62	0.55	0.55	0.55	0.55	0.55	0.55
5	0.56	0.54	0.54	0.54	0.54	0.54	0.54
6	0.55	0.55	0.55	0.55	0.55	0.55	0.55
7	0.47	0.39	0.39	0.39	0.39	0.39	0.39
8	0.35	0.34	0.34	0.34	0.34	0.34	0.34
9	0.65	0.58	0.58	0.58	0.58	0.58	0.58
10	0.5	0.48	0.48	0.48	0.48	0.48	0.48
11	0.51	0.44	0.44	0.44	0.44	0.44	0.44
12	0.57	0.56	0.56	0.56	0.56	0.56	0.56
Mean	0.61	0.57	0.57	0.57	0.57	0.57	0.57

Table A.13: Correlation statistics for the PETS2004 dataset comparing depth limited search: T_K

Bibliography

- [1] Allen G.L. ‘A developmental perspective on the effects of “subdividing” macrospatial experience.’ *Journal of Experimental Psychology: Human Learning and Memory*, Vol 7(2), pp. 120–132, 1981.
- [2] Allen G.L. and Kirasic K.C. ‘Effects of the cognitive organization of route knowledge on judgments of macrospatial distance.’ *Memory and Cognition*, Vol 13(3), pp. 218–227, 1985.
- [3] Andreas Schadschneider K.N. Ansgar Kirchner. ‘CA approach to collective phenomena in pedestrian dynamics.’ In: *International Conference on Cellular Automata for Research and Industry*, pp. 239–248. 2002.
- [4] Armitage R. *To CCTV or not to CCTV? A review of current research in the effectiveness of CCTV systems in reducing crime*. NACRO, London, 2002.
- [5] Aronoff S. *Geographical Information Systems: A management perspective*. W.D.L. publications, Ottawa, Canada, 1989.
- [6] Baumberg A. and Hogg D.C. ‘An efficient method for contour tracking using active shape models.’ In: *Proc. of the IEEE workshop on Motion on Non-Rigid and Articulated Objects*, pp. 194–199. 1994.
- [7] Baumberg A. and Hogg D.C. ‘Learning flexible models from image sequences.’ In: *Proc. European Conference on Computer Vision (ECCV)*, Vol 1, pp. 299–308. 1994.
- [8] Baumberg A. and Hogg D.C. ‘Learning spatiotemporal models from examples.’ *Image and Vision Computing*, Vol 14(8), pp. 525–532, 1996.
- [9] Benhamou S. and Poucet B. ‘Landmark use by navigating rats (*rattus norvegicus*): Contrasting geometric and featural information.’ *Journal of Comparative Psychology*, Vol 112, pp. 317–322, 1998.

- [10] Berendt B. and Jansen-Osmann P. ‘Feature accumulation and route structuring in distance estimations - an interdisciplinary approach.’ In: *Spatial information theory: A theoretical basis for GIS*, pp. 279–296. 1997.
- [11] Bird N.D., Masoud O., Papanikolopoulos P. and Isaacs A. ‘Detection of loitering individuals in public transportation areas.’ *IEEE Transactions on intelligent transportation systems*, Vol 6(2), pp. 167–177, 2005.
- [12] Black J., Ellis T. and Makris D. ‘A hierarchical database for visual surveillance applications.’ In: *Proc. IEEE International Conference on Multimedia and expo*, pp. 1571–1574. Taipei, Taiwan, 2004.
- [13] Black J., Ellis T. and Rosin P. ‘Multi view image surveillance and tracking.’ In: *IEEE Workshop on motion and Video computing*, pp. 169–174. Orlando, FL, 2002.
- [14] Black J., Velastin S. and Boghossian B. ‘A real-time surveillance system for metropolitan railways.’ In: *Proc. International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 189–194. Como, Italy, 2005.
- [15] Borenstein J. and Koren Y. ‘Real-time obstacle avoidance for fast mobile robots.’ *IEEE Transactions on Systems, Man, and Cybernetics*, Vol 19(5), pp. 1179–1187, 1989.
- [16] Borenstein J. and Koren Y. ‘Real-time obstacle avoidance for fast mobile robots in cluttered environments.’ In: *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, pp. 572–577. Cincinnati, Ohio., 1990.
- [17] Brand M. and Kettner V. ‘Discovery and segmentation of activities in video.’ *IEEE transactions on Pattern Analysis and Machine Intelligence (PAMI)*, Vol 22(8), pp. 747–757, 2000.
- [18] Brand M., Oliver N. and Pentland A. ‘Coupled Hidden Markov Models for complex action recognition.’ In: *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 994–999. 1997.
- [19] Brock O. and Khatib O. ‘Elastic strips: A framework for motion generation in human environments.’ *International Journal of Robotics Research*, Vol 21(12), pp. 1031–1052, 2002.
- [20] Buxton H. ‘Generative models for learning and understanding dynamic scene activity.’ In: *1st International Workshop on Generative-Model-Based Vision*, pp. 77–81. Copenhagen, Denmark, 2002.

- [21] Buxton H. 'Learning and understanding dynamic scene activity: a review.' *Image and Vision Computing*, Vol 21(1), pp. 125–136, 2003.
- [22] Buxton H. and Gong S. 'Visual surveillance in a dynamic and uncertain world.' *Artificial Intelligence*, Vol 78(1-2), pp. 431–459, 1995.
- [23] Campbell N., Dalton C., Gibson D. and Thomas B. 'Practical generation of video textures using the auto-regressive process.' In: *Proc. British Machine Vision Conference (BMVC)*, pp. 434–443. Cardiff, UK, 2002.
- [24] Canny J.F. and Lin M.C. 'An opportunistic global path planner.' *Algorithmica*, Vol 10, pp. 102–120, 1993.
- [25] Chong K.S. and Kleeman L. 'Feature-based mapping in real, large scale environments using an ultrasonic array.' *International Journal of Robotics Research*, Vol 18(2), pp. 3–19, 1999.
- [26] Churchland P.M. 'Folk psychology and the explanation of human behavior.' In: J.D. Greenwood (editor), *The Future of Folk Psychology: Intentionality and cognitive science*, pp. 51–69. Cambridge University Press, Cambridge, 1991.
- [27] Churchland P.M. and Churchland P.S. 'Stalking the wild epistemic engine.' *Nous*, Vol 17, pp. 5–18, 1983.
- [28] Clarke G.M. and Cooke D. *A basic course in statistics*. Edward Arnold, London, 1992, 3rd edition.
- [29] Conroy Dalton R. 'The secret is to follow your nose: Route path selection and angularity.' *Environment and Behavior*, Vol 35(1), pp. 107–131, 2003.
- [30] Cootes T. and Taylor C. 'A mixture model for representing shape variation.' In: *Proc. British Machine Vision Conference (BMVC)*, pp. 110–119. Essex, UK, 1997.
- [31] Davis A.C. and Velastin S.A. 'A progress review of intelligent CCTV surveillance systems.' In: *Third IEEE Workshop on Intelligent Data Acquisition Systems: Technology and Applications*, pp. 417–423. Sofia, Bulgaria, 2005.
- [32] Davison A.J. and Murray D.W. 'Simultaneous localisation and map-building using active vision.' *IEEE transactions on Pattern Analysis and Machine Intelligence (PAMI)*, Vol 24(7), pp. 865–880, 2002.

- [33] Dempster A., Rubin D. and Laird N. ‘Maximum likelihood from incomplete data via the EM algorithm.’ *Journal of the Royal Statistical Society*, Vol 39, pp. 1–38, 1977.
- [34] Denis M., Pazzaglia F., Cornoldi C. and Bertolo L. ‘Spatial discourse and navigation: An analysis of route directions in the city of Venice.’ *Applied Cognitive Psychology*, Vol 13, pp. 145–174, 1999.
- [35] Dennett D.C. *The Intentional Stance*. The MIT Press/Bradford Books, Cambridge, MA, 1987, reprinted 2002.
- [36] Dennett D.C. ‘True believers: The intentional strategy and why it works.’ In: W.G. Lycan (editor), *Mind and Cognition: A Reader*, pp. 150–167. Blackwell, Cambridge, MA, 1990.
- [37] Dennett D.C. *Consciousness Explained*. Little-Brown, 1991.
- [38] Devin V.E. and Hogg D.C. ‘Reactive memories: an interactive talking head.’ In: *Proc. British Machine Vision Conference (BMVC)*, pp. 603–612. Manchester, UK, 2001.
- [39] Dijkstra E.W. ‘A note on two problems in connexion with graphs.’ *Numerische Mathematik*, Vol 1, pp. 269–271, 1959.
- [40] Ditton J. and Short E. ‘Evaluating Scotland’s first town centre CCTV scheme.’ In: C. Norris, J. Moran and G. Armstrong (editors), *Surveillance, closed circuit television and social control*, pp. 155–173. Ashgate, Aldershot, 1998.
- [41] Duckham M. and Kulik L. ‘“Simplest paths”: Automated route selection for navigation.’ In: W. Kuhn, M.F. Worboys and S. Timpf (editors), *Spatial Information Theory: Foundations of Geographic Information Science*, pp. 182–199. Springer, Berlin, 2003.
- [42] Ellis T. and Xu M. ‘Object detection and tracking in an open and dynamic world.’ In: *IEEE CVPR workshop on Performance Evaluation of Tracking and Surveillance*. Kawai, Hawaii, 2001.
- [43] Faltings B. and Pu P. ‘Applying means-ends analysis to spatial planning.’ In: *AAAI Spring Symposium on Reasoning with Diagrammatic Representations*. 1992.
- [44] Fodor J. *Psychosemantics*. The MIT Press/Bradford Books, Cambridge, MA, 1987.

- [45] Fraichard T. ‘Trajectory planning in dynamic workspaces: a state-time space approach.’ *Advanced Robotics*, pp. 75–94, 1999.
- [46] Fraichard T. and Ahuactzin J.M. ‘Smooth path planning for cars.’ In: *Proc. IEEE International Conf. on Robotics and Automation*. 2001.
- [47] Galata A., Cohn A.G., Magee D.R. and Hogg D.C. ‘Modeling interaction using learnt qualitative spatio-temporal relations and variable length Markov Models.’ In: *Proc. European Conference on Artificial Intelligence (ECAI)*, pp. 741–745. Lyon, France, 2002.
- [48] Galata A., Johnson N. and Hogg D.C. ‘Learning variable length Markov Models of behaviour.’ In: *Computer Vision and Image Understanding: CVIU*, pp. 398–413. 2001.
- [49] Gallistel C.R. ‘Animal cognition: The representation of space, time and number.’ *Annual Review of Psychology*, Vol 40, pp. 155–189, 1989.
- [50] Gibbins D., Newsam G.N. and Brooks M.J. ‘Detecting suspicious background changes in video surveillance of busy scenes.’ In: *Proc. 3rd IEEE Workshop on Applications of Computer Vision*, pp. 22–26. 1996.
- [51] Golledge R. ‘Path selection and route preference in human navigation: a progress report.’ In: *Proc. Spatial Information Theory: Foundations of GIS (COSIT)*, pp. 207–222. 1995.
- [52] Golledge R.G. ‘Defining the criteria used in path selection.’ Technical Report UCTC No. 78, University of California Transportation Center, 1995.
- [53] Gong S. and Xiang T. ‘Recognition of group activities using dynamic probabilistic networks.’ In: *Proc. International Conference on Computer Vision (ICCV)*, pp. 742–749. Nice, France, 2003.
- [54] Graves A. and Gong S. ‘Wavelet based holistic sequence descriptor for generating video summaries.’ In: *Proc. British Machine Vision Conference (BMVC)*, pp. 167–176. Kingston, UK, 2004.
- [55] Greenhill D., Renno J., Orwell J. and Jones G.A. ‘Occlusion analysis: Learning and utilising depth maps in object tracking.’ In: *Proc. British Machine Vision Conference (BMVC)*, pp. 467–476. Kingston, UK, 2004.

- [56] Grimson W.E.L., Stauffer C., Romano R. and Lee L. ‘Using adaptive tracking to classify and monitor activities in a site.’ In: *Proc. Computer Vision and Pattern Recognition (CVPR)*, pp. 246–252. Santa Barbara, CA, 1998.
- [57] Haritaoglu I., Harwood D. and Davis L.S. ‘W4: Real-time surveillance of people and their activities.’ *IEEE transactions on Pattern Analysis and Machine Intelligence (PAMI)*, Vol 22(8), pp. 809–830, 2000.
- [58] Hartley R.I. and Zisserman A. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2000.
- [59] Hongeng S. ‘Unsupervised learning of multi-object event classes.’ In: *Proc. British Machine Vision Conference (BMVC)*, pp. 487–496. Kingston, UK, 2004.
- [60] Hongeng S. and Nevatia R. ‘Multi-agent event recognition.’ In: *Proc. International Conference on Computer Vision (ICCV)*, pp. 84–91. Vancouver, Canada, 2001.
- [61] Horgan T. and Woodward J. ‘Folk psychology is here to stay.’ *Philosophical Review*, Vol 94, pp. 197–226, 1985.
- [62] Howarth R.J. and Buxton H. ‘Visual surveillance monitoring and watching.’ In: *Proc. European Conference on Computer Vision (ECCV)*, pp. 321–224. Cambridge, UK, 1996.
- [63] Howarth R.J. and Buxton H. ‘Conceptual descriptions from monitoring and watching image sequences.’ *Image and Vision Computing*, pp. 105–135, 2000.
- [64] Huang T. and Russell S. ‘Object identification in a Bayesian context.’ In: *Proc. International Joint Conference on Artificial Intelligence(IJCAI)*, pp. 1276–1283. Nagoya, Japan, 1997.
- [65] Hung H. and Gong S. ‘Detecting and quantifying unusual interactions by correlating salient action.’ In: *Proc. International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 46–51. Como, Italy, 2005.
- [66] Huttenlocher D.P., Klanderman G.A. and Rucklidge W.J. ‘Comparing images using the Hausdorff distance.’ *IEEE transactions on Pattern Analysis and Machine Intelligence (PAMI)*, Vol 15(9), pp. 850–683, 1993.
- [67] Intille S.S. and Bobick A.F. ‘Recognising planned, multiperson action.’ *Computer Vision and Image Understanding (CVIU)*, Vol 81, pp. 414–445, 2001.

- [68] Isard M. and Blake A. ‘A mixed-state CONDENSATION tracker with automatic model-switching.’ In: *Proc. International Conference on Computer Vision (ICCV)*, pp. 107–112. Bombay, India, 1998.
- [69] Isard M. and MacCormick J. ‘BraMBLe: A Bayesian multiple-blob tracker.’ In: *Proc. International Conference on Computer Vision (ICCV)*, pp. 34–41. Vancouver, Canada, 2001.
- [70] Ivanov Y.A. and Bobick A.F. ‘Recognition of visual activities and interactions by stochastic parsing.’ *IEEE transactions on Pattern Analysis and Machine Intelligence (PAMI)*, Vol 22(8), pp. 852–872, 2000.
- [71] Jan T., Piccardi M. and Hintz T. ‘Detection of suspicious pedestrian behavior using modified probabilistic neural network.’ In: *Proc. of Image and Vision Computing*, pp. 237–241. Auckland, New Zealand, 2002.
- [72] Jansen-Osman P. and Berendt B. ‘What makes a route appear longer? An experimental perspective on features, route segmentation and distance knowledge.’ *To appear in: The quarterly Journal of Experimental Psychology*, Vol 58A, 2005.
- [73] Jansen-Osmann P. and Wiedenbauer G. ‘The influence of turns on distance cognition: New experimental approaches to clarify the route-angularity effect.’ *Environment and Behavior*, Vol 36(6), pp. 790–813, 2004.
- [74] Johnson N. *Learning object behaviour models*. Ph.D. Thesis, University of Leeds, 2000.
- [75] Johnson N., Galata A. and Hogg D. ‘The acquisition and use of interaction behaviour models.’ In: *Proc. Computer Vision and Pattern Recognition (CVPR)*, pp. 866–871. Santa Barbara, CA, 1998.
- [76] Johnson N. and Hogg D. ‘Representation and synthesis of behaviour using Gaussian mixtures.’ *Image and Vision Computing*, Vol 20(12), pp. 889–894, 2002.
- [77] Johnson N. and Hogg D.C. ‘Learning the distribution of object trajectories for event recognition.’ In: *Proc. British Machine Vision Conference (BMVC)*, pp. 583–592. 1995.
- [78] Johnson N. and Hogg D.C. ‘Learning the distribution of object trajectories for event recognition.’ *Image and Vision Computing*, Vol 14(8), pp. 609–615, 1996.

- [79] Kalman R. 'A new approach to linear filtering and prediction problems.' *Transactions of the ASME - Journal of Basic Engineering*, pp. 35–45, 1960.
- [80] Kashiwagi N. 'On the use of the Kalman filter for spatial smoothing.' *Annals of the Institute of Statistical Mathematics*, Vol 45(1), pp. 21–34, 1993.
- [81] Kendall M.G. and Gibbons J.D. *Rank correlation methods*. Edward Arnold, London, 1990.
- [82] Kruskal W.H. 'Ordinal measures of association.' *Journal of the American Statistical Association*, Vol 53 No. 284., pp. 814–861, 1958.
- [83] Kukla R., Kerridge J., Willis A. and Hine J. 'PEDFLOW: Development of an autonomous agent model of pedestrian flow.' In: *80th Annual Meeting TRB - Spatial analysis in urban activity and travel demand modelling*. Washington, DC, 2001.
- [84] Liberty. 'CCTV.', 2005. [Http://www.liberty-human-rights.org.uk/privacy/cctv.shtml](http://www.liberty-human-rights.org.uk/privacy/cctv.shtml).
- [85] Lipton A.J., Fujiyoshi H. and Patil R.S. 'Moving target classification and tracking from real-time video.' In: *IEEE workshop on applications of computer vision*, pp. 129–136. 1998.
- [86] Magee D.R. 'Tracking multiple vehicles using foreground, background and shape models.' *Image and Vision Computing*, Vol 22, pp. 143–155, 2004.
- [87] Magee D.R. and Boyle R.D. 'Building shape models from image sequences using piecewise linear approximation.' In: *Proc. British Machine Vision Conference (BMVC)*, pp. 398–408. Southampton, UK, 1998.
- [88] Magee D.R. and Boyle R.D. 'Detecting lameness using 're-sampling condensation' and 'multi-stream cyclic Hidden Markov Models'.' *Image and Vision Computing*, Vol 20(8), pp. 581–594, 2002.
- [89] Mahajan D., Kwatra N., Jain S., Kalra P. and Banerjee S. 'A framework for activity recognition and detection of unusual activities.' In: *Proc. Indian Conference on Computer Vision, Graphics and Image Processing*. 2004.
- [90] Makris D. and Ellis T. 'Finding paths in video sequences.' In: *Proc. British Machine Vision Conference (BMVC)*, pp. 263–272. Manchester, UK, 2001.

- [91] Makris D. and Ellis T. ‘Spatial and probabilistic modelling of pedestrian behaviour.’ In: *Proc. British Machine Vision Conference (BMVC)*, pp. 557–566. Cardiff, UK, 2002.
- [92] Makris D. and Ellis T. ‘Automatic learning of an activity based semantic scene model.’ In: *Proc. International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 183–188. Miami, FL, 2003.
- [93] Makris D. and Ellis T. ‘Learning semantic scene models from observing activity in visual surveillance.’ *IEEE Transactions on Systems, Man and Cybernetics*, Vol 35(3), pp. 397–408, 2005.
- [94] Makris D. and Ellis T.J. ‘Path detection in video surveillance.’ *Image and Vision Computing*, Vol 20(12), pp. 895–903, 2002.
- [95] McCahill M. and Norris C. ‘CCTV in Britain.’ In: *On the threshold to Urban Panopticon?: Analysing the Employment of CCTV in European Cities and Assessing its Social and Political Impacts*. Technical University Berlin, 2003.
- [96] McCahill M. and Norris C. ‘CCTV systems in London: Their structures and practices.’ In: *On the threshold to Urban Panopticon?: Analysing the Employment of CCTV in European Cities and Assessing its Social and Political Impacts*. Technical University Berlin, 2003.
- [97] McCloskey M. ‘Intuitive physics.’ *Scientific American*, Vol 248(4), pp. 122–130, 1983.
- [98] McKenna S.J. and Nait Charif H. ‘Learning spatial context from tracking using penalised likelihoods.’ In: *Proc. International Conference on Pattern Recognition (ICPR)*, pp. 138–141. Cambridge, UK, 2004.
- [99] McKenna S.J. and Nait Charif H. ‘Summarising contextual activity and detecting unusual inactivity in a supportive home environment.’ *Pattern Analysis and Applications*, Vol 7(4), pp. 386–401, 2004.
- [100] Medioni G., Cohen I., Bremond F., Hongeng S. and Nevatia R. ‘Event detection and analysis from video streams.’ *IEEE transactions on Pattern Analysis and Machine Intelligence (PAMI)*, Vol 23(8), pp. 873–889, 2001.
- [101] Meng M. and Kak A.C. ‘Mobile robot navigation using neural networks and non-metrical environment models.’ In: *IEEE Control Systems*, pp. 30–39. 1993.

- [102] Montello D.R. ‘The perception and cognition of environmental distance: Direct sources of information.’ In: *Spatial information theory: A theoretical basis for GIS*, pp. 297–311. 1997.
- [103] Morris R.J. and Hogg D.C. ‘Statistical models of object interaction.’ *International Journal of Computer Vision*, Vol 37(2), pp. 209–215, 2000.
- [104] Nait Charif H. and McKenna S.J. ‘Activity summarisation and fall detection in a supportive home environment.’ In: *Proc. International Conference on Pattern Recognition (ICPR)*. Cambridge, UK, 2004.
- [105] Needham C.J. and Boyle R.D. ‘Performance evaluation metrics and statistics for positional tracker evaluation.’ In: *Proc. International Conference on Computer Vision Systems*, pp. 278–289. Austria, 2003.
- [106] Norris C. and Armstrong G. *The Unforgiving Eye: CCTV Surveillance in Public Space*. University of Hull, Hull, 1997.
- [107] Norris C. and Armstrong G. *The Maximum Surveillance Society*. Berg, Oxford, 1999.
- [108] Norris V., McCahill M. and Wood D. ‘Editorial: The growth of CCTV: a global perspective on the international diffusion of video surveillance in publicly accessible space.’ *Surveillance and Society*, Vol 2(2/3), pp. 110–135, 2004.
- [109] Oliver N., Rosario B. and Pentland A. ‘Statistical modeling of human interactions.’ In: *Proc. IEEE CVPR Workshop on the Interpretation of Visual Motion*, pp. 39–46. Santa Barbara, CA, 1998.
- [110] Oliver N.M., Rosario B. and Pentland A.P. ‘A Bayesian computer system for modeling human interactions.’ *IEEE transactions on Pattern Analysis and Machine Intelligence (PAMI)*, Vol 22(8), pp. 831–843, 2000.
- [111] O’Toole A.J., Harms J., Snow S.L., Hurst D.R., Pappas M.R., Ayyad J.H. and Abdi H. ‘A video database of moving faces and people.’ *IEEE transactions on Pattern Analysis and Machine Intelligence (PAMI)*, Vol 27(5), pp. 812–816, 2005.
- [112] Pasula H., Russell S., Ostland M. and Ritov Y. ‘Tracking many objects with many sensors.’ In: *Proc. International Joint Conference on Artificial Intelligence(IJCAI)*, pp. 1160–1171. Stockholm, Sweden, 1999.

- [113] Remagnino P., Baumberg A., Grove T., Hogg D.C., Tan T., Worrall A. and Baker K. 'An integrated traffic and pedestrian model-based vision system.' In: *Proc. British Machine Vision Conference (BMVC)*, pp. 380–389. Essex, UK, 1997.
- [114] Remagnino P., Tan T. and Baker K. 'Agent orientated annotation in model based visual surveillance.' In: *Proc. International Conference on Computer Vision (ICCV)*, pp. 857–862. Bombay, India, 1998.
- [115] Remagnino P., Tan T. and Baker K. 'Multi-agent visual surveillance of dynamic scenes.' *Image and Vision Computing*, Vol 16, pp. 529–532, 1998.
- [116] Renner M. 'Contributions of the honey bee to the study of timesense and astronomical orientation.' In: *Cold Spring Harbor Symposia on Quantitative Biology*, pp. 361–368. New York, NY, 1960.
- [117] Rowe N.C. 'Detecting suspicious behaviour from positional information.' In: *Modelling Others from Observations Workshop at IJCAI*. Edinburgh, Scotland, 2005.
- [118] Sacks H. 'Notes on police assessment of moral character.' In: D. Sudnow (editor), *Studies in Social Interaction*, pp. 280–293. Free Press, New York, 1972.
- [119] Sadalla E.K., Burroughs W.J. and J. S.L. 'Reference points in spatial cognition.' *Journal of Experimental Psychology: Human Learning and Memory*, Vol 6(5), pp. 516–28, 1980.
- [120] Sadalla E.K. and Magel S.G. 'The perception of traversed distance.' *Environment and Behavior*, Vol 12, pp. 65–79, 1980.
- [121] Sage K.H. and Buxton H. 'Joint spatial and temporal structure learning for task based control.' In: *Proc. International Conference on Pattern Recognition (ICPR)*, pp. 48–51. Cambridge, UK, 2004.
- [122] Säisä J., Svenson-Gärling A., Gärling T. and Lindberg E. 'Intraurban cognitive distance: The relationship between judgements of straight-line distances, travel distances and travel times.' *Geographical Analysis*, Vol 18(2), pp. 167–174, 1986.
- [123] Santos-Victor J., Sandini G., Curotto F. and Garibaldi S. 'Divergent stereo in autonomous navigation: From bees to robots.' *International Journal of Computer Vision*, Vol 14(2), pp. 159–177, 1995.

- [124] Scheuer A. and Fraichard T. ‘Continuous-curvature path planning for car-like vehicles.’ In: *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, Vol 2, pp. 997–1003. 1997.
- [125] Schneider L.F. and Taylor H.A. ‘How do you get there from here? Mental representations of route descriptions.’ *Applied Cognitive Psychology*, Vol 13, pp. 415–441, 1999.
- [126] Schwerdt K., Maman D., Bernas P. and Paul E. ‘Target segmentation and event detection at video-rate: the eagle project.’ In: *Proc. International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 183–188. Como, Italy, 2005.
- [127] Scödl A. and Essa I. ‘Depth layers from occlusions.’ In: *Proc. Computer Vision and Pattern Recognition (CVPR)*, pp. 339–644. Kawai, Hawaii, 2001.
- [128] Senior A. ‘Tracking people with probabilistic appearance models.’ In: *IEEE workshop on Performance Evaluation of Tracking and Surveillance*, pp. 48–55. Copenhagen, Denmark, 2002.
- [129] Seyve C. ‘Metro railway security algorithms with real world experience adapted to the ratp dataset.’ In: *Proc. International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 177–182. Como, Italy, 2005.
- [130] Shao H., Li L., Xiao P., Maylor K. and Leung H. ‘ELEVIEW: An active elevator video surveillance system.’ In: *Workshop on Human Motion*, pp. 67–72. 2000.
- [131] Sherrah J. and Gong S. ‘Automated detection of localised visual events over varying temporal scales.’ In: *Proc. European Workshop on Advanced Video-based Surveillance Systems*, pp. 215–227. Kingston, UK, 2001.
- [132] Sherrah J. and Gong S. ‘Continuous global evidence-based Bayesian modality fusion for simultaneous tracking of multiple objects.’ In: *Proc. International Conference on Computer Vision (ICCV)*, pp. 42–49. Vancouver, Canada, 2001.
- [133] Siegal S. and Castellan N.J. *Nonparametric statistics for the behavioral sciences*. McGraw Hill, Singapore, 1988, 2nd edition.
- [134] Skinns D. ‘Crime reduction, diffusion and displacement: evaluating the effectiveness of CCTV.’ In: C. Norris, J. Moran and G. Armstrong (editors), *Surveillance, closed circuit television and social control*, pp. 175–188. Ashgate, Aldershot, 1998.

- [135] Smith G.J.D. ‘Behind the screens: Examining constructions of deviance and informal practices among CCTV control room operators in the UK.’ *Surveillance and Society*, Vol 2(2/3), pp. 376–395, 2004.
- [136] Spirito M., Regazzoni C.S. and Marcenaro L. ‘Automatic detection of dangerous events for underground surveillance.’ In: *Proc. International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 195–200. Como, Italy, 2005.
- [137] Stauffer C. ‘Automatic hierarchical classification using time-based co-occurrences.’ In: *Proc. Computer Vision and Pattern Recognition (CVPR)*, pp. 333–339. Ft. Collins, CO, 1999.
- [138] Stauffer C. ‘Estimating tracking sources and sinks.’ In: *Proc. 2nd IEEE workshop on event mining*, pp. 259–266. Madison, WI, 2003.
- [139] Stauffer C. and Grimson E. ‘Learning patterns of activity using real-time tracking.’ *IEEE transactions on Pattern Analysis and Machine Intelligence (PAMI)*, Vol 22(8), pp. 747–757, 2000.
- [140] Stauffer C. and Grimson W. ‘Adaptive background mixture models for real-time tracking.’ In: *Proc. Computer Vision and Pattern Recognition (CVPR)*, pp. 246–252. Fort Collins, CO, 1999.
- [141] Stefanakis E. and Kavouras M. ‘On the determination of the optimum path in space.’ In: *Proc. Spatial Information Theory: Foundations of GIS (COSIT)*, pp. 241–257. 1995.
- [142] Stich S. *From folk psychology to cognitive science: the case against belief*. The MIT Press/Bradford Books, Cambridge, MA, 1983.
- [143] Still K.G. *Crowd Dynamics*. Ph.D. Thesis, University of Warwick, 2000.
- [144] Streeter L.A., Vitello D. and Wonsiewicz S.A. ‘How to tell people where to go: comparing navigational aids.’ *International Journal of Man-Machine Studies*, Vol 22, pp. 549–562, 1985.
- [145] Sumpter N. *The Robotic Sheepdog: Modelling animal behaviour from image sequences*. Ph.D. Thesis, University of Leeds, 1999.
- [146] Sumpter N. and Bulpitt A. ‘Learning spatio-temporal patterns for predicting object behaviour.’ *Image and Vision Computing*, Vol 18(9), pp. 697–704, 1999.

- [147] Thrun S., Fox D. and Burgard W. ‘A probabilistic approach to concurrent mapping and localisation for mobile robots.’ *Machine Learning*, Vol 31(1-3), pp. 29–53, 1998.
- [148] Tilley N. ‘Evaluating the effectiveness of CCTV schemes.’ In: C. Norris, J. Moran and G. Armstrong (editors), *Surveillance, closed circuit television and social control*, pp. 139–153. Ashgate, Aldershot, 1998.
- [149] Troscianko T., Holmes A., Stillman J., Mirmehdi M., Wright D. and Wilson A. ‘What happens next? the predictability of natural behaviour viewed through CCTV cameras.’ *Perception*, Vol 33(1), pp. 87–101, 2004.
- [150] Viola P., Jones M.J. and Snow D. ‘Detecting pedestrians using patterns of motion and appearance.’ In: *Proc. International Conference on Computer Vision (ICCV)*, pp. 734–741. Nice, France, 2003.
- [151] Vogler C. and Metaxas D. ‘A framework for recognising the simultaneous aspects of american sign language.’ *Computer Vision and Image Understanding (CVIU)*, Vol 81, pp. 358–384, 2001.
- [152] Wallace R. *Finding natural clusters through entropy minimization*. Ph.D. Thesis, CMU, 1989.
- [153] Waller D., Loomis J.M., Golledge R.G. and Beall A.C. ‘Place learning in humans: The role of distance and direction information.’ *Spatial Cognition and Computation*, Vol 2(4), pp. 333–354, 2000.
- [154] Walter M., Psarrou A. and Gong S. ‘Learning prior and observation augmented density models for behaviour recognition.’ In: *Proc. British Machine Vision Conference (BMVC)*, pp. 23–32. Nottingham, UK, 1999.
- [155] Willis A., Kukla R., Kerridge J. and Hine J. ‘Laying the foundations: the use of video footage to explore pedestrian dynamics in PEDFLOW.’ In: *Pedestrian Evacuation and Dynamics*, pp. 181–186. Dursburg, Germany, 2001.
- [156] Winter S. ‘Weighting the path continuation in route planning.’ In: *Proceedings of the ninth ACM international symposium on Advances in geographic information systems*, pp. 173–176. Atlanta, Georgia, USA, 2001.
- [157] Wu G., Wu Y., Jiao L., Wang Y. and Chang E. ‘Multicamera spatio-temporal fusion and biased sequence-data learning for security surveillance.’ In: *Proc. of ACM*

- International Conference on Multimedia, November 2003.*, pp. 528–538. Berkeley, CA, 2003.
- [158] Xu M. and Ellis T. ‘Partial observation vs. blind tracking through occlusion.’ In: *Proc. British Machine Vision Conference (BMVC)*, pp. 777–786. Cardiff, UK, 2002.
- [159] Yagi Y., Imai K., Tsuji K. and Yachida M. ‘Iconic memory-based omnidirectional route panorama navigation.’ *IEEE transactions on Pattern Analysis and Machine Intelligence (PAMI)*, Vol 27(1), pp. 78–87, 2005.
- [160] Yeap W.K. and Jefferies M. ‘On early cognitive mapping.’ *Spatial cognition and computation*, Vol 2, pp. 85–116, 2000.
- [161] Zhao W., Chellappa R., Phillips P.J. and Rosenfield A. ‘Face recognition: A literature survey.’ *ACM computing surveys*, Vol 35(4), pp. 399–458, 2003.
- [162] Zheng J.Y. and Tsuji S. ‘Panoramic representation for route recognition by a mobile robot.’ *International Journal of Computer Vision*, Vol 9(1), pp. 55–76, 1992.
- [163] Zilani F., Velastin S., Porikli F., Marcenaro L., Kelliher T., Cavallaro A. and Bruneaut P. ‘Performance evaluation of event detection solutions: the CREDS experience.’ In: *Proc. International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 201–206. Como, Italy, 2005.