Learning Crowd dynamics using Computer Vision

Kingston University London

Beibei Zhan

Faculty of Computing,Information Systems and Mathematics

Kingston University

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

Doctor of Philosophy

September 2008

This work is sponsored by British Telecom and Kingston University

Acknowledgements

I would like to express my deepest appreciation to all who have contributed to the completion of this thesis.

I am very grateful to my director of study - Dr Paolo Remagnino, for his warm consistent encouragement and guidance throughout my research work.

Thanks go to my supervision team Dr Sergio Velastin and Dr Dorothy Monekosso for their continuous support, kind help and great suggestions.

I would like to express my sincere gratitude to Dr Li-qun Xu, my industrial supervisor from British Telecom Research for his inspiring discussion and warm encouragement.

I would like to acknowledge British Telecom Research and Kingston University for the financial support of my research work. I also would like to acknowledge European Office of Aerospace Research and Development (EOARD) project for supporting the group analysis work and Legion Ltd for providing video data for extreme crowded analysis.

Thanks to ORION project team of INRIA (France), Prof Yoshinori KUNO from Saitama University (Japan) and Artificial Intelligence research group of University of Roma-La Sapienza (Italy) for hosting me as a visiting student during my research work which helps me to broaden my horizons and gives me invaluable experience in international research environments.

Thanks to my colleagues from Digital Imaging Digital Imaging Research Centre for their support and advices.

Lastly, but not in the least, to my partner for providing me his untiring care and encouragement; to my parents for their support far away from China, I am greatly thankful and grateful.

Abstract

An increase of violence in public spaces has prompted the introduction of more sophisticated technology to improve the safety and security of very crowded environments. Research disciplines such as civil engineering and sociology, have studied the crowd phenomenon for years, employing human visual observation to estimate the characteristics of a crowd. Computer vision researchers have increasingly been involved in the study and development of research methods for the automatic analysis of the crowd phenomenon. Until recently, most existing methods in computer vision have been focussed on extracting a limited number of features in controlled environments, with limited clutter and numbers of people. The main goal of this thesis is to advance the state of the art in computer vision methods for use in very crowded and cluttered environments. One of the aims is to devise a method that in the near future would be of help in other disciplines such as socio-dynamics and computer animation, where models of crowded scenes are built manually on painstaking visual observation.

A series of novel methods is presented here that can learn crowd dynamics automatically by extracting different crowd information from real world crowded scenes and modelling crowd dynamics using computer vision. The developed methods include an individual behaviour classifier, a scene cluttering level estimator, two people counting schemes based on colour modelling and tracking, two algorithm for measuring crowd motion by matching local descriptors, and two dynamics modelling methods - one based on statistical techniques and the other one based on a neural network. The proposed information extracting methods are able to gather both macro information, which represents the properties of the whole crowd, and micro information, which is different from individual (location) to individual (location). The statistically-based dynamics modelling models the scene implicitly. Furthermore, a method for discovering the main path of the crowded scene is developed based on it. Self-Organizing Map (SOM) is chosen in the neural network approach of modelling dynamics; the resulting SOMs are proven to be able to capture the main dynamics of the crowded scene.

List of Publications

Journal Papers

1. B. Zhan, N.D. Monekosso, P. Remagnino, S.A. Velastin, L Xu, "Crowd Analysis: a Survey" in 'Machine Vision and Applications', (2008).

Chapters in Books

1. B. Zhan, P. Remagnino, N.D. Monekosso, S.A. Velastin, Chapter "The Analysis of Crowd Dynamics: From Observations to Modelling" in 'Collaborative Computational Intelligence', Edited by C.L. Mumford and L.C. Jain, Springer, accepted.
2. B. Zhan, N.D. Monekosso, S. Rush, P. Remagnino, S.A. Velastin, Chapter "Augmenting Professional Training, an Ambient Intelligent Approach" in 'Intelligent Environments, Methods, Algorithms and Applications', Advanced Information and Knowledge Processing Edited by D.N.Monekosso, P.Remagnino and Y.Kuno, Springer, (2008).

Conferences

1. B. Zhan, P. Remagnino, N.D. Monekosso, S.A. Velastin, "Self-Organizing Maps for the Automatic Interpretation of Crowd", International Symposium on Visual Computing (ISVC), Dec., Las Vegas, (2008).
2. B. Zhan, P. Remagnino, S.A. Velastin, N.D. Monekosso, L Xu, "A quantitative comparison of two new motion estimation algorithms", International Symposium on Visual Computing (ISVC), Springer, November, Lake Tahoe, Nevada, (2007).
3. B. Zhan, N.D. Monekosso, P. Remagnino, T Rukhsana, A Mansur, Y Kuno, "Skin Patches Trajectories as Scene Dynamics Descriptors", IAPR Conference on Machine Vision Applications 2007, (2007).

4. B. Zhan, P. Remagnino, S.A. Velastin, F. Bremond, M Thonnat, "Matching gradient descriptors with Topological Constraints to characterise the Crowd dynamics", IET Visual Information Engineering 2006, IET, September, pp. 441-445. ISBN / ISSN 0863416713, (2006).

5. B. Zhan, P. Remagnino, L Xu, S.A. Velastin, N.D. Monekosso, "Motion Estimation with Edge Continuity Constraint for Crowd Scene Analysis", International Symposium on Visual Computing (ISVC) 2006, (2006).

6. B. Zhan, P. Remagnino, S.A. Velastin, "Analysing Crowd Intelligence", Second AIxIA Workshop on Ambient Intelligence, September, Milan, Italy, (2005).

7. B. Zhan, P. Remagnino, S.A. Velastin, "Mining paths of complex crowd scenes", Advances in Visual Computing: First International Symposium, ISVC 2005 (Eds. G Bebis, R Boyle, D Koracin, B Parvin), Lecture Notes in Computer Science (Vol. 3804/2005) Springer-Verlag GmbH, December, Nevada, USA, pp. 126-133. ISBN/ISSN 3-540-30750-8 (2005).

8. B. Zhan, P. Remagnino, S.A. Velastin, "VISUAL ANALYSIS OF CROWDED PEDESTRIAN SCENES", XLIII Congresso Annuale AICA 2005, October, Udine, Italy, pp. 549-555. (2005).

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The overall objective of this work is to develop computer vision methods that are able to learn the dynamics of real world crowded scenes. "Dynamics" refers to the directions, velocity and diversions of people, and all the related information of the crowd in the scene. This chapter includes a concise introduction to the background of the problem, a discussion of the motivation for the work, the challenges which need to be overcome and a summary of the contributions to the state of the art. The structure of the thesis is presented at the end of this chapter.

1.1 Background

Crowd analysis work can be traced back to the 19^{th} century when the work was mainly from a psychology perspective (93) (85). During the last half of the 20th century, interpretations of crowd dynamics, including using computational descriptions of the crowd, were proposed by civil engineers and sociologists (58)(57). Human observations play a very important role in the above work; all the crowd features used in the analysing work are extracted manually, which is not efficient at all.

The explosion of the global population, along with the world's urbanisation from the late 20^{th} century, have had an impact on the frequent occurrence of the crowd phenomenon. Consequently, there is an increasing concern about people's quality of life. Crowd vision scientists have begun to seek a way of automatically

learning crowd dynamics; however, until now most of the work has been working with a group of people (around 10 people) rather than working with real crowd scenes.

The definition of a crowd is introduced here as: "a temporary collection of a large number of individuals who come together in a common place for a common purpose" (125). This definition gives three elements of the crowd: "large number of individuals", "common place" and "common purpose". In addition, this definition suggests that people in crowds are purposive, and their dynamics can be modelled.

It is a common experience that in a crowded situation an individual has to consider space limitation, interactions with others and sometimes the information passed on by the crowd - all of which would not be of any consequence if they were walking alone. A crowd also increases the risks in public safety and security, especially when very dense.

G. Keith Still, the founder of Crowd Dynamics Ltd, proposes the relationship between crowd speed and density as a major factor of crowd dynamics in his PhD thesis(128). The area of the relationship between crowd speed and density is from the work of John J. Fruin(45). Fruin defined the level of service concept where the density and speed relationship are stated as guidelines for comfort and safety. A brief summary of the level of service for a walkway can be found as follows:

- When the crowd density equals the plan area of a human body, individual control is lost, as a person becomes an "involuntary" part of the mass;
- With an occupancy of about 7 people per square metre, a crowd becomes almost a fluid mass;
- Shock waves can be propagated through the mass, sufficient to lift people off their feet and propel them a distance of 3 metres or more.

According to the level of service, crowd dynamics is highly related to the density of the crowd. In this work, crowd video data is considered to have gravity in terms of the number of people included in the camera view. Particularly, videos with group scenes are defined as those recorded in the medium field of view, where

(a) A frame from a group scene

(b) A frame from an extremely crowded scene

Figure 1.1: Example frames from videos with group and extremely crowded scenes.

for every individual, most of his or her body parts can be isolated from all the others in the scene. Videos with crowd scenes are defined as recorded in the far field of view, where only some of the body parts (typically the head and shoulders) of each individual can be seen. Figure 1.1 illustrates two frames from group and extremely crowded scenes[1].

1.2 Motivation and Challenges

The records on Legion's website (87) show that every year from 1990, there have been hundreds of people hurt or killed in crowd disasters. It is also concluded on the website that the failures in both the design and management and crowd "mis" behaviour can be the primary cause of accidents and incidents. Nowadays, crowd safety is becoming a major public safety issue, meaning that better designs/management strategies and methods to avoid or prevent crowd "mis" behaviours are highly desired. Computer vision methods can provide an automatic way to determine crowd dynamics and an instantaneous way of validating and adjusting the captured dynamics. Furthermore, a visual surveillance system with knowledge of "normal" behaviour of crowd is able to detect crowd "mis" behaviour and improve the crowd's safety. The increasing requirement of efficient

[1]The video data with the extremely crowded scene used in this thesis is kindly provided by Legion Ltd.

ways to improve crowd safety and security is one of the major reasons why the work of this thesis is being carried out.

In addition, understanding crowd behaviour is a very important component for computer vision systems to understand the real world. With the information, models and crowd behaviours, a computer vision system will be able to provide a way of improving daily experiences for humans. Moreover, the study of crowd dynamics in the real world can be used in crowd simulations in computer graphics and virtual environments.

In summary, an automatic framework of crowd dynamics analysis by computer vision methods is extremely useful for different kinds of applications, from visual surveillance and crowd management through to computer graphics.

However, the analysis of crowd dynamics is a challenging topic in the field of computer vision research, not only because of the complicated nature of the problem but also because video data with real world crowded scenes are hard to find in the public domain. As a result, until recently the existing literature in computer vision was still rare and crowd analysis work in computer vision was still at an initial stage. Challenges mainly come from following different aspects:

Crowd analysis itself is a challenging problem. For different types of video data, the desirable information is different. For video data with group scenes, individual activities are of the same importance for collective activities. Meanwhile, for video data with extreme crowded scenes, individual activities are much less important, or even not important at all, when compared to collective activities. The problem is how to identify different requirements for the different types of video data and how to fulfil these requirements.

Technically, there is a lack of motion extraction techniques. Traditional motion detection techniques such as background removal and optical flow do not work well in medium and extremely crowded situations. In crowded scenes with an increasing number of foreground objects, the visibility of the background is decreases. Further, serious occlusions and the high density of the foreground objects make techniques such as optical flow and feature matching less efficient.

The challenge also comes from the gap between computer vision crowd research

and traditional crowd research. Most of the existing scene interpretation and behaviour analysis methods developed in computer vision techniques focus on individual behaviour. Crowd behaviour is very different from individual behaviour. Crowd dynamics has very distinctive properties and the literature from the computer vision of individual behaviour analysis can hardly be applied to it. On the other hand, existing crowd models from traditional research are being developed in an entirely different procedure, and it is not possible to apply the models directly to the computer vision approach.

In a word, the challenges arise from the nature of the crowd analysis problem, from the lack of previous work and from how different crowd research methodologies (traditional research and computer vision research) can be bridged.

1.3 Contributions to the State of the Art

The main contributions this work brings are methods that can determine the dynamics from crowded scenes. The methods are developed for retrieving information and interpretations of crowd behaviour from video data captured in the real world. As discussed, the videos used in this work are roughly divided into two types: the medium field of view for a group situation and the far field of view for extreme crowded situations. Algorithms are designed to work with the two types of the situations and extract different kinds of information. The extracted information and the dynamics model can be used in many applications such as visual surveillance and ambient intelligence. By providing an online analysis of the dynamics of the crowd video, the methods can be used to avoid crowd emergencies in public events, to automatically monitor a crowded scene, or in assisting intelligent systems to help the crowd. Furthermore, the built model of a scene can be used in offline analysis for improving crowd management in the scene. In summary, the contributions of this PhD work are:

A comprehensive review of related crowd analysis work from different research fields. Traditional crowd analysis, working in disciplines such as civil engineering or psychology, are based on human observations.

The analysis work is focussed on investigating the rules of how different crowd features influence each other. In the area of computer vision, crowd analysis is a fairly new topic; most existing work is based on the previous work of tracking individuals. In this thesis, most of the existing work is discussed and a framework of crowd analysis, in which the existing literature is functionally positioned, has been proposed. This is the first review work on crowd analysis from the computer vision perspective covering the different research areas. The review work gives a clear view of the relationship of the research from different fields.

Methods for retrieving macro information from crowded scenes. In this thesis, macro information refers to the information that represents the collective information of a crowded scene. In this PhD, the retrieved macro information includes a scene cluttering level and the main path of a crowded scene. Entropy is employed to estimate the level of cluttering in the scene where the cluttering depends on the distribution of the people within the scene. The main path of a crowded scene is retrieved by combining foreground coverage and motion frequency information. Macro information refers to the properties of a crowded scene when the crowded scene is regarded as a single research object. The chaotic level represents the current the status of the crowded scene while the main path preserves the dynamics of a crowded scene over time.

Methods for retrieving micro information from crowded scenes. In contrast to macro information, micro information refers to the information that describes the individual/local information of a crowded scene. Instead of regarding a crowded scene as a single research object, information is processed and gathered separately to represent the different dynamics of different individuals and locations from the scene. In this PhD work, methods for retrieving micro information include an individual behaviour classifier for a group scene and two motion estimation algorithms for extremely crowded scenes. The individual behaviour classifier is able to estimate the degree of people's interest in the group scene by analysing the individual's trajectory. The classification of individual behaviour can help the computer vision sys-

tem to analyse the group scene. For extremely crowded scenes where most tracking methods are not valid, two novel motion estimation methods employing local descriptors and refined constraints are proposed. The motion estimation methods are able to work with videos of a crowded scene. These methods are able to work with the sort of low frame rate videos that most existing methods fail to work with. Moreover, the performance comparison of the two motion estimations is also presented.

Methods for individual detection and counting people in a group scene. The number of people in the scene is a significant feature as the level of service can be inferred by the feature. In order to count people, individuals must be detected and separated from each other. As a result, the retrieved information includes both micro information (the locations of individuals) and macro information (the total number of individuals in the scene). In this thesis, besides the brief introduction of a simple and efficient counting method, a more accurate method is proposed, discussed and tested. The distinguishing advantage of the latter method is that it has only one assumption of the spatial distribution of the people in the scene, while it can work with a single static camera without any knowledge of the camera calibration, nor any other information. The proposed methods allow the computer vision system to be able to automatically determine and learn the total number in the scene.

Methods for crowd modelling. Crowd modelling captures and represents the recurrent dynamics of the crowded scene. In this thesis, a statistical approach for group modelling and a neural network approach for general crowd modelling are proposed. In the statistical approach, Probability Density Functions (PDFs) are used for modelling the foreground objects and the motions of the foreground objects over time. Self Organizing Map (SOM)-based modelling methods are able to capture the main dynamics of the crowded scene and represent it in a grid format. The models can be used to classify different crowd scenes. Furthermore, the built models can produce a better image of the dynamics in a particular scene.

The new methods proposed in this thesis are to retrieve and model crowd dynamics. The retrieved macro information includes the level of cluttering and the main path of the scene. The retrieved micro information includes the classifications of the individual behaviours and motions. In addition, methods are proposed for individual detection, counting people, and for the modelling of crowd dynamics.

1.4 Synopsis of Chapter 2 - 6

In chapter 2, a literature review covering all crowd related research is presented, although the focus is on computer vision methodologies. An overview of crowd research from all related disciplines is given, followed by a detailed review of the methodologies that can be adopted into computer vision-based crowd analysis. The review includes both computer vision and traditional methodologies.

Chapter 3 presents the methods developed for extracting and analysing the dynamics for a group of people. In this chapter, research is mainly carried out for an experimental environment with complex human activities. The behaviour classifier, scene chaotic estimator and the two people counting algorithms are included in this chapter.

Chapter 4 proposes the methods developed for measuring crowd dynamics. This chapter presents the two methods for retrieving motion from crowd scenes. The details of the methods, including the backgrounds of the local descriptors and the refined matching constraints, are introduced in the chapter. Moreover, a performance comparison of the two methods is also presented.

Chapter 5 discusses the methods developed for modelling group and crowd dynamics. The statistical approach using two Probability Density Functions (PDF) to model the group dynamics, and discovering the main path of the crowded scene, is based on the PDF models. The neuron network approach using Self-Organizing Maps to capture and model the major crowd dynamics is presented in this chapter.

Chapter 6 draws from the previous chapters to reach a conclusion and to summarise the major achievements of the work. It also discusses potential improvements to the work and, finally, future research.

Chapter 2

Literature Review

In 1999, the world population reached 6 billion - doubling the previous census estimate of 1960. Recently, the United States Census Bureau issued a revised forecast for the world population, showing a projected growth to 9.4 billion by 2050 (137). Different research disciplines have studied the crowd phenomenon and its dynamics from a social, psychological and computational standpoint, respectively. This chapter presents a survey on the crowd analysis methods employed in computer vision research and discusses perspectives from other research disciplines and how they can contribute to the computer vision approach. The aim of this chapter is to provide a comprehensive overview of the problems within the field before going on to the details of the PhD work.

2.1 Introduction

Steady population growth, along with worldwide urbanisation, has made the crowd phenomenon more frequent. It is not surprising, therefore, that crowd analysis has received attention from technical and social research disciplines. The crowd phenomenon is of great interest in a large number of applications:

Crowd Management: Crowd analysis can be used for developing crowd management strategies, especially for increasingly more frequent and popular events such as sports matches, large concerts, public demonstrations and so on, to avoid crowd related disasters and ensure public safety.

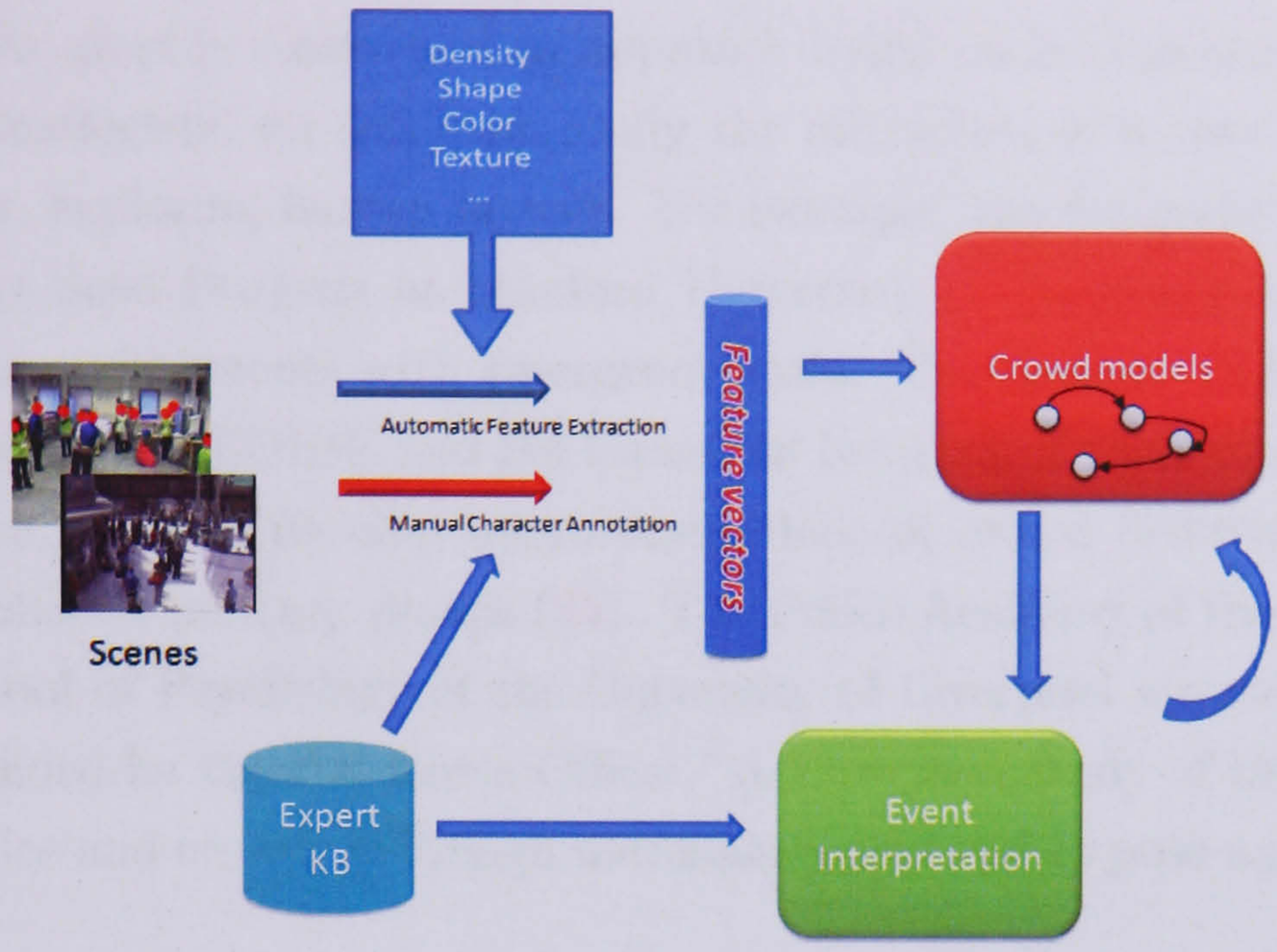


Figure 2.1: A framework for Crowd analysis.

Public Space Design: Crowd analysis can provide guidelines for the design of public spaces, e.g. to make the layout of shopping malls more convenient to customers or to optimise the space usage of an office.

Virtual Environments: Mathematical models of crowds can be employed in virtual environments to enhance the simulation of crowd phenomena and to enrich the human life experiences.

Visual Surveillance: Crowd analysis can be used for the automatic detection of anomalies and alarms. Furthermore, the ability to track individuals in a crowd could help the police to catch suspects.

Intelligent Environments: In some intelligent environments that involve large groups of people, crowd analysis is a prerequisite for assisting the crowd or an individual in the crowd. For example, in a museum, deciding how to divert the crowd is based on the patterns of the crowd.

Crowd management and public space design have been studied by sociologists, psychologists and civil engineers, virtual environments by computer graphic researchers, and visual surveillance and intelligent environments by computer vision researchers. The approach favoured by psychology, sociology, civil engineer

and computer graphic research is an approach based on human observation and analysis. Sociologists, for instance, study the characters of a crowd as a social phenomenon, exploring human factors. For example, the computational model, developed by Seed Projects at Stanford University (136), incorporated human behaviour in environments with emergency exits. The Crowd - MAGS Project, which is funded by GEOIDE and the Canadian Network of Centers of Excellence in Geomatics, aims to develop micro-simulations of crowd behaviours and the impact of police or military groups (35). The Police Academy of the Netherlands and the School of Psychology of the University of Liverpool are cooperating on a project funded by the UK Home Office: "A European study of the interaction between police and crowds of foreign nationals considered to pose a risk to public order" (1).

On the other hand, computational methods such as those employed in computer graphics and vision methods focus on extracting quantitative features and detecting events in crowds - synthesising the phenomenon with mathematical and statistical models. For example, early projects funded by the EPSRC in the UK were concerned with measuring crowd motion and density and, hence, potentially dangerous situations (38) (139) (146). The EU funded projects PRISMATICA (109) and ADVISOR (2), completed in 2003, were concerned with the management of public transport networks through CCTV cameras. The UK EPSRC funded project, BEHAVE, was concerned with pre-screening of video sequences for the detection of abnormal or crime-oriented behaviour (19). ISCAPS (70), started in 2005, is a consortium of 10 European ICT companies and academic organizations that aims to provide automated surveillance of crowded areas. SERKET, a recently started EU project, aims to develop methods to prevent terrorism (67).

Figure 2.1 illustrates the processes involved in crowd analysis. In a crowd scene, the attributes of importance are crowd density, location, speed, etc. This information can be extracted either manually or automatically using computer vision techniques. Crowd models can then be built based on the extracted information. Event discovery is achieved using pre-compiled knowledge of the scene or by using the computational model, although both approaches can be combined. In both cases, the model is updated with newly extracted information.

Sensor typology and topology	Moving/Static platform Number of cameras Type of video sequenc: colour/gray scale, etc.	
Environmental conditions	Indoor/outdoor Level of clutter Light condition, etc.	
Scene typology	Individual characters Appearance, etc.	location/velocity/etc.
	Collective Average speed, etc.	Crowd density

Table 2.1: Features in crowd analysis by computer vision methods.

This chapter is organised as follows. Section 2 introduces research in automatic crowd feature extraction. Section 3 discusses existing work on crowd modelling and crowd event inference. Sections 4 and 5 provide some examples of how the two complementary approaches can be bridged.

2.2 Crowd Information Extraction

The components of crowd analysis from a computer vision perspective are described in Table 2.1. Essentially, the typology and topology of the sensors influence the scene capture processes. Environmental conditions such as natural and artificial illumination changes often introduce noise, and the scene typology affects the type of process one requires to extract the most accurate information of a dynamics scene.

Visual surveillance methods have been developed to estimate the motion of objects and people in a scene, in isolation or in groups; a review can be found in (62). When video is analysed for very crowded scenes, conventional computer vision methods are not appropriate. In these cases, methods must be designed to cope with extreme clutter. Features from conventional image processing are still employed such as colour, shape and texture etc. However, sophisticated methods have been developed to retrieve crowd information. In the following sections, the existing state of the art will be reviewed.

2.2.1 Density Measurement

An important crowd feature is crowd density, and it is natural to think that crowds of different densities should receive a different level of attention. Polus et al. (108) provide a clear idea of the problem of the *level of services* for a pedestrian flow, defined as: free flow, restricted flow, dense flow, and jammed flow according to a density metric defined as the number of pedestrians per unit area. The research reviewed here either estimates the crowd density directly or counts the number of pedestrians that provide information for density estimation.

Research methods have been proposed for crowd analysis which employ background removal techniques. In (146), a reference image with only a background is used to classify image pixels as belonging to either pedestrians or the background. A functional relationship between the number of pedestrian-classified pixels and the number of people is then established manually for the measurement of crowd density. Another example is proposed by Ma et al. (92) using background removal. A mathematical relation for the geometric correction of the ground plane is derived. The authors prove that this can be directly applied to all foreground pixels. A linear relation between the number of pixels and people is derived by applying the geometric correction. These works have a typical assumption that the number of foreground pixels is proportional to the number of people, which is only true when there are not serious occlusions between people. (40) makes use of examples to directly map the global shape feature to configurations of humans. This training-based algorithm is quite a novel approach, but the problem of how to decide the size of the training dataset remains unclear.

Image processing and pattern recognition techniques are also used for the analysis of the scene to estimate crowd density. Marana et al. (96) assume that images of low-density crowds tend to present a coarse texture, while images of dense crowds tend to present fine textures. Self-organising neural maps (97), combined with Minkowski fractal dimensions (95), are employed to deduce the crowd density from the texture of the image. The work by Marana is compared in (111) with another method that uses Chebyshev moments. An optimisation of performance under different illumination conditions is discussed. Lin et al. (90) present a system that estimates the crowd size through the recognition of

the head contour by using Haar wavelet transform (HWT) and support vector machines (SVM).

The approach of information fusion has also be applied, e.g. Yang et al. (145) estimate the number of people directly from groups of image sensors. For each sensor, foreground objects are segmented from the background and the resulting silhouettes are aggregated over the sensor network. A geometric algorithm is then introduced to limit the number and possible locations of people using silhouettes extracted by each sensor. Alternative methods combine several techniques to achieve more accurate and reliable measurements. For example, in (139), an edge-based technique is integrated with background removal using a Kalman filter. Marana et al. (94) use different methods, including Fourier and Fractal analysis and classifiers, to estimate the crowd density level. Kong et al., in (82)(83), employ background subtraction and edge detection; the work defines the extracted edge orientation and blob size histograms as features. The relationship between the feature histograms and the number of pedestrians is determined from labelled training data. Obviously, more cues may indicate a more accurate solution.

2.2.2 Recognition

Conventional visual surveillance focuses on object detection and tracking. In essence, image processing techniques are employed to extract the chromatic and shape information of the moving objects and the background for detecting and tracking purposes.

For crowd dynamics modelling, detecting and tracking are also important as they provide the location and velocity features of the dynamics. Crowded scenes add a degree of complexity to the conventional detection and tracking problem of single individuals. In the following sections, the focus will be on methodologies for crowded situations.

2.2.2.1 Near Field

The face is the most discriminating feature of the human body, and many researchers try to detect pedestrians through face detection. The majority of the

existing research employs supervised learning methods. A few attempts to detect the faces in complex scenes are introduced in the following text.

Early works such as (132) use a technique where genetic algorithms are employed for face localisation in a complex scene. The system proceeds with a training phase to generate a simple object mean image using a single object image, and a test phase using arbitrary images.

However, this previous work highly depends on the training set and if the faces appear at different sizes and orientations, it may require a very large training set and long processing time. Hence, different techniques have been developed to address the problem of multi-view face detection. (89) proposes a pyramid structure that adopts a coarse-to-fine strategy to handle pose variance. Another approach by Jones et al. (71) illustrates how different detectors are used for different views of the face, and a decision tree is trained to determine the viewpoint class. (64) uses a Width-First search tree structure to improve the performance in both speed and accuracy. This kind of work is quite likely to be adopted into crowd analysis, especially from a single camera view, as the problem of human pose and the perspective are both compensated here.

Methodologies for stereo face detections in crowd have also been developed. For example, Huang et al. (65) propose a three-steps technique: first extracting the likelihood evidence of heads from the stereo image by scale-adaptive filtering and then spurious clues are suppressed from the extracted points according to the average human height. Finally, the human heads are located by applying a mean-shift algorithm to the likelihood map.

2.2.2.2 Medium to Far Field

Pedestrian detection and tracking is a well studied problem in computer vision. Many methods have been proposed such as using the afore mentioned background removal technique, or by combining the chromatic and shape information of the tracked pedestrians. The following sections discuss the methods that attempt to provide a solution for pedestrian detection in crowded scenes.

- **Occlusion handling.** Occlusion caused by the high clutter of pedestrians in a crowd scene is the major challenge for the crowd detection problem.

Some research addresses the problem by using human body parts. Wu et al. (142) propose a method to detect multiple, partially occluded humans in a single image. Edgelet features are introduced in their work. Part detectors, based on edgelet features, are learned by a boosting method. The responses of part detectors are combined to form a joint likelihood model that includes cases of multiple, possibly inter-occluded humans. The human detection problem is then formulated as one of a maximum a posteriori (MAP) estimation. The models of the group of people in (42) are initialised based on segmenting the body into regions by modelling their appearance and spatial distribution. A framework uses the maximum likelihood estimation to estimate the best arrangement of people in terms of a 2D translation that yields segmentation for the foreground region. Occlusion reasoning is then conducted to recover relative depth information.

Leibe et al. (88) present a different algorithm that integrates evidence in multiple iterations and from different sources. The local cue is based on a scale-invariant extension of an Implicit Shape Model (ISM), and global consistency is enforced by adding the information from global shape cues. Local and global cues are combined via a probabilistic top-down segmentation to detect the pedestrian.

- **Moving Views.** Special solutions are required for moving platforms for some of the applications, e.g. for an onboard vision system to assist a driver.

  Some of the implementations make assumptions of a human's appearance. In Broggi et al.'s work (26), a coarse detection of pedestrians is computed through the processing of a single image based on the assumption of the symmetry, size and ratio shape of a human body. Heisele et al. (52) apply spatio-temporal methodologies by recognising walking a pedestrian based on the characteristic motion of the legs of the pedestrian walking parallel to the image plane. Each image is segmented into region-like image parts by clustering pixels in a combined colour/position feature space. A classifier is then used to extract the clusters, which are most likely to be the pedestrian's legs.

  In contrast to the above, Shashua et al. (119) describe a functional and

architectural breakdown of the pedestrian detection system. Single classification is based on a scheme of breaking down the class variability by repeatedly training a set of relatively simple classification performance results. The path from single-frame to system level performance includes the integration of additional cues measured over time and situation specific features via building up additional object categories consisting of vehicles and stationary background structures.

- **Spatial-temporal methods.** Besides conventional cues of pedestrian appearance, space-temporal cues are used for detection. Brostow et al. (27) tackle the problem by tracking simple image features and probabilistically grouping them into clusters that represent independently moving entities. Space-time proximity and trajectory coherence through the image space are used as the only probabilistic criteria for clustering. Moreover, this motion-based detection could be easily extended to the tracking of individuals in dense crowds by merging the outcomes.

  In extremely cluttered scenes, individual pedestrians cannot be properly segmented in the image. However, sometimes the *crowd* within which the pedestrians share a similar purpose can be recognised. Reisman et al. (115) propose a scheme that uses slices in the spatial-temporal domain to detect inward motion, as well as intersections between multiple moving objects. The system calculates a probability distribution function for left and right inward motion and uses these probability distribution functions to infer a decision for crowd detection.

2.2.3 Tracking

Tracking has been proposed to localise the interested object in time-space. The velocity feature of tracked object also can be derived afterwards. As a natural extension of detection, though, tracking has its own problem in recognising and identifying pedestrians in the consecutive frames. Tracking could be regarded as the most popular topic in visual surveillance; however, currently for crowd analysis, most of the techniques are validated only for multiple (e.g. up to 10) people.

As discussed in the last subsection, occlusions could occur very frequently when there are many objects and people in the scene. Tracking techniques have to overcome the problem in order to continuously track before, during and after the occurrence of occlusions. A comprehensive review on occlusion handling can be found in (48). A formulation of the occlusion problem is provided, and the techniques are divided into two groups: the merge-split approach, which addresses the problem in re-establishing object identities following a split, and the straight-through approach, which maintains object identities at all times.

The following text covers three aspects: the techniques that are developed to track multiple people (objects) without any assumptions of the dependence of their motion, e.g. interactions, the techniques that try to explain the interactions between the pedestrians, and some practical analysis of handling the problem of occlusion in a crowd situation.

The following text covers three aspects: the techniques that are developed to track multiple people (objects) without any assumptions of the dependence of their motion, e.g. interactions, the techniques that try to explain the interactions between the pedestrians, and some practical analysis of handling the problem of occlusion in a crowd situation.

2.2.3.1 Tracking Methodologies

Crowd scenes increase the complexity of tracking because there are multiple moving objects in the scene. Quite a few techniques have been developed based on the colour, geometry and other features for tracking.

- **Likelihood.** Colour, edge etc. are the most popular features in tracking. In a crowd, salient traceable image features are of particular interest for tracking. As one of the better candidates, interest points (IPs) are employed in (48) and (99). In both works, the IPs are obtained by a popular colour Harris detector. Gabriel characterised IPs, by their position relative to the estimated centre of the object and Mathes, build a point distribution model between ASM and AAM. Both of the methods require a pre-defined region (or object) of interest. Their salient features benefit from their robustness under different light conditions. The tracking inference using these

features can work better under occlusions than when using the entire contour. Therefore, the usage of those features could be more applicable to a large amount of people in the scene.

- **Human body model.** Methods using models of human bodies or human body parts have been developed for tracking in complex crowded scenes, which are usually completed with probabilistic frameworks. Zhao et al. (147)(148) worked on the former approach, using explicit 3D human shape models. The problems of detection and tracking are formulated as a Bayesian inference to find the best interpretation given the image observations. The latest one is the work of Wu et al. (143), who extend the previous detection work in (142) (which has been discussed) using edgelet features to human body part detectors. Tracking is implemented by probabilistic data association, i.e. matching the object hypotheses with the detected response.

- **Tracking inference strategies.** Tracking inference strategies have been developed for the problem of tracking multiple objects. For non-linear and non-Gaussian dynamic models, the particle filter technique - also known as CONDENSATION(69) - is one of the most popular. Particle filters are sequential Monte Carlo methods based upon point mass (or 'particle') representations of probability densities (41). A large portion of multiple object tracking work has employed this technique. For example, Venegas et al.(140) use a particle filter to track the moving objects by generating hypotheses on the top-view reconstruction of the scene. Okuma et al. (105) combine mixture particle filters and an Adaboost algorithm. Sidenbladh et al. (120) extend the particle filter formulation according to finite set statistics (FISST) for tracking. Cai et al. (28) tackle the problem by embedding the meanshift algorithm into the particle filter framework. Koller-Meier et al. (81) introduce an extension of the CONDENSATION algorithm that relies on a single probability distribution to describe the likely states of multiple objects. Kang et al. (73) propose the discrete shape model and the competition rule to improve the performance of the condensation tracker for real time tracking.

The Multiple Hypotheses Tracker (MHT) and Joint Probabilistic Data Association Filter (JPDAF) address the data association problem. MHT tries to keep the track of all the possible hypotheses over time (114). A detailed summary and discussion of MHT for multiple target tracking is included in (20). MHT suffers from the storage of the redundant track, hence some of the work proposes extensions and modifications to the algorithm to obtain better performances, e.g. (50). JPDAF computes a Bayesian estimation of correspondence between the different features and objects, e.g. Rasmussen and Hager (113) apply this technique with colour region and a snake-based tracker. Another approach has been introduced by Karlsson (74), which uses the Monte Carlo method.

The fusion of the different cues from a number of detection and tracking algorithms is also used to produce a more robust tracker. Siebel et al. (121) propose a tracking system containing three cooperating parts: an Active Shape Tracker, a Region Tracker, and a Head Detector. (124) proposes an approach based on the principles of the self-organisation of the integration mechanism and self-adaptation of the cue models during tracking. Cues from different sensors and models can increase the dimension of information, which is preferable in the multiple objects situations. However, the goodness of integration scheme is very crucial in these algorithms.

2.2.3.2 Tracking Interacting People

In certain cases, interaction happens frequently in a crowded scene. Researchers have shown great interest in studying these interactions to derive new perspectives on tracking techniques.

Some of the work formulates the interaction to enhance the tracking scheme. For example, both Smith et al. (122) and Khan et al. (76) propose to use the Markov Chain Monte Carlo (MCMC) and the particle filter. Smith used a joint multi-object state-space formulation and a trans-dimensional MCMC particle filter to recursively estimate the multi-object configuration and to efficiently search the state-space. Khan developed a joint tracker that included a motion model to maintain the identity of targets throughout the interaction, thus reducing tracker

failure. Pre-defined motion models are used in this approach, with the trade-off between improving the tracking performance in a crowd with known interactions and the adoption of the motion model to an arbitrary crowd.

Some researchers interpret interactions as relationships between pedestrians and a group (pedestrian merging/splitting into groups). Marques et al. (98) propose a two-layer solution to overcome the problem. The first layer produces a set of spatial temporal strokes based on low level operations to track the active regions. The second layer performs a consistent labelling of detected segments using a statistical model based on Bayesian networks, which is recursively computed during the tracking operation. Mckenna et al. (101) perform tracking at three levels: regions, people and groups. Background subtraction is used to cope with shadows and unreliable colour cues. Colour information is used to disambiguate occlusions and to provide qualitative estimates of depth ordering and position. Pedestrian merging and group splitting are frequent phenomena in the crowded scene; however, the major challenge for these kind of methods is to recover the object label after splitting from the group.

Sullivan et al. (130) label tracking targets by exploring the trajectories. Trajectories of when a target is isolated are found, and it is claimed that these trajectories end when targets interact. A graph structure is formed by the interactions of these trajectories. This method could be very useful for offline crowd analysing but for online processing it may have a bottleneck in the storage of the trajectories.

2.2.3.3 Tracking from Multiple Views

For large public areas, the use of a multi-camera system is required to cover most of the monitored areas.

For the multi-camera system arrangement, Mittal et al. (102) present a system named "M2Tracker", using multiple synchronised cameras located far from each other that segment, detect and track multiple people in a cluttered scene. First, a region-based stereo algorithm is introduced for finding 3D points inside an object. Then, a scheme is developed that dynamically assigns priors for different objects at each pixel. Finally, the evidences gathered from different camera pairs are

combined by using occlusion analysis to obtain a globally optimum detection and tracking of the objects. A different arrangement of cameras is used in (30). This method uses both static and Pan-Tilt-Zoom (PTZ) cameras. The static cameras are used to locate people in the scene, while the PTZ cameras *lock-on* to the individuals and provide visual attention. The underlying visual processes rely on colour segmentation, movement tracking and shape information to locate target candidates, and colour indexing methods to register these candidates with the PTZ cameras.

Meanwhile, special techniques have been developed for the tracking from a multi-view; normally, a planar homography constraint would be included. For example, in (75), feet regions of the people are located by the constraint. The contiguous spatial-temporal region formed by the feet regions belonging to the same person are clustered as the track of the person. In (77), people's ground points are located and a multi-hypothesis framework using a particle filter is developed for tracking.

2.2.4 Motion Extraction

Crowd motion is a concept distinct from individual motion and it represents the overall motion of a crowd. In a very dense crowd individuals are restricted and constrained by the motion of the crowd. The extraction of crowd motion helps to understand these restrictions and constraints and their influence over the individuals in the crowd. Techniques for crowd motion extraction have been proposed quite recently in computer vision research.

In (4) Lagrangian particle dynamics has been employed to segment crowd moton. Lagrangian Coherent Structures are constructed and used to divide crowd flow into regions of qualitatively different dynamics. Mathematical complexity might be a potential problem for efficient and fast computation.

Motion patterns can be defined as groups of flow vectors that are part of the same physical process. From the same group of authors, two different techniques have been proposed for extracting motion patterns. The first method in (60) constructs super tracks, which are the collective representations of motion patterns, based on detecting the representative modes of motion vectors. The

second method in (61) formulates the problem as a clustering problem of the motion flow fields. A hierarchical agglomerative clustering algorithm is applied to group flow vectors into motion patterns. Both methods produce good visual results while the first one lacks quantitative results. Though extraction of crowd motion is very important in analysing very dense crowded scenes, the work on it is relatively rare, which is probably caused by the lack of reference and ground truthed video data.

2.3 Crowd Modelling and Events Inference

Dynamics in public spaces can indeed be recurrent. Crowd information can be better exploited to indicate the status of the crowd so its events can be inferred. Crowd models have been built to represent these statuses, either implicitly or explicitly. On the other hand, some research makes direct use of crowd information instead of building models. In such cases, the events are usually inferred based on some prior knowledge of the properties of the particular scene and the crowd. In this section, the inference of crowd models and events in computer vision will be presented, as well as some crowd models from non-vision areas.

2.3.1 Crowd models' and crowd events' inference in computer vision

In computer vision, crowd modelling is achieved based on the extracted information from visual data and can normally be employed in crowd events inference. Meanwhile, there are also some approaches that attempt to infer events without the construction of models. Here, examples are given for both of the cases.

- In the computer vision approach, crowd models are built as representations of recurrent behaviours by analysing video data of the crowd through vision methods.

  Andrade et al. (14)(13)(11) characterise crowd behaviour by observing the crowd's optical flow associated with the crowd and use unsupervised feature extraction to encode normal behaviour. The unsupervised feature

extraction applies spectral clustering to find the optimal number of models to represent normal motion patterns. The motion models are HMMs to cope with the variable number of motion samples that might be present in each observation window. The objective of this model is to detect abnormal events in crowd scenes.

- Apart from building models, in computer vision crowd monitoring systems, the extracted information is used to recognise the event; usually under some assumptions of the involved crowds. Early work on crowd monitoring using image processing is reviewed by Davies et al. (38).

  In more recent work such as that by Boghossian et al. (21), a system is presented using computer vision techniques to estimate the paths and directions of crowd flows in CCTV images and to improve the perception of scene dynamics by offering online illustrations.

  Maurin et al. (100) propose a system to detect, track and monitor both pedestrians (crowds) and vehicles. The system contains a detection scheme based on optical flow that can locate vehicles, individual pedestrians and a crowd. The detection phase is followed by the tracking phase that tracks all the detected entities. Traffic objects are tracked and a rich set of descriptors are computed for each object that include a wealth of information (position, velocity, acceleration/deceleration, bounding box, and shape).

  Cupillard et al. carry out event recognition by means of *behaviour*. In (37)(36), an approach using multiple cameras is presented. The algorithm relies on both low level motion detection and tracking, and a high level module that recognises predefined scenarios corresponding to specific behaviours.

  Michael et al. (29) present a method that jointly performs the recognition of complex events and links fragmented tracks. The recognition work is implemented by combing the appearance and kinematic constraints from tracking and constraints from a hypothesised event model.

  In these methods especially, assumptions of the crowd are usually involved - indicating that some prior knowledge is required for events inference. These

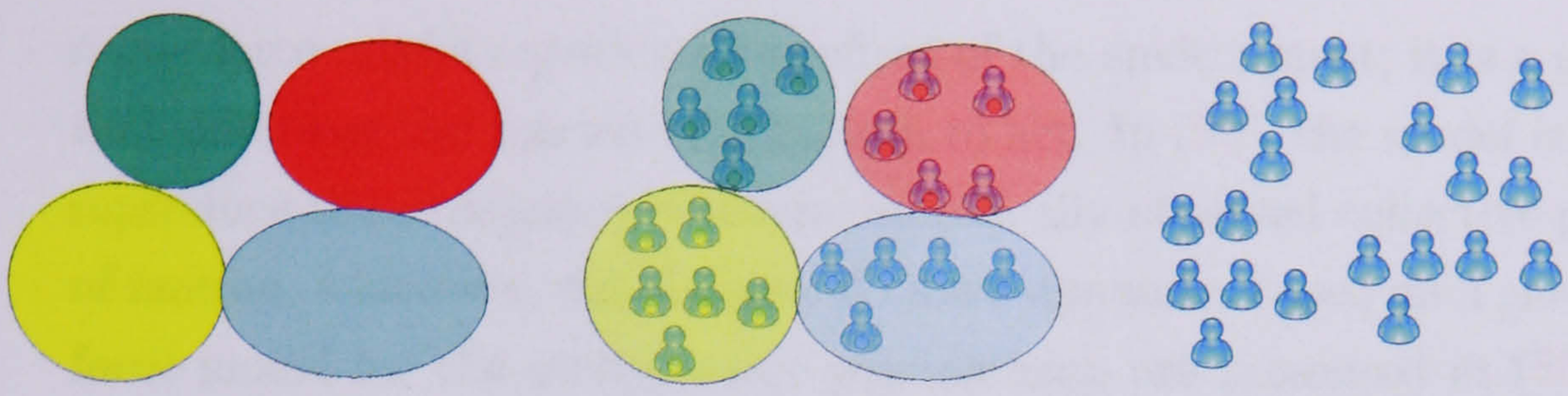


Figure 2.2: (left) Macroscopic, (centre) Mesoscopic, (right) Microscopic.

methods may be very efficient and computationally inexpensive for some particular systems where the interested events are simple and clear, although this is not always the case in general situations.

2.3.2 Crowd models from non vision approach

Computational models aim to describe and predict the collective effects of crowd behaviour by identifying the relationship between crowd features. There are three distinct philosophies for modelling a crowd. Traffic analysis (44) proposes a categorisation where crowd models can be defined as microscopic, mesoscopic and macroscopic. The microscopic model deals with pedestrians as discrete individuals, the macroscopic model deals with the crowd as a whole and the mesoscopic model combines the properties of the previous two, either keeping a crowd as a homogeneous mass but considering an internal *force*, or keeping the characters of the individuals while maintaining a general view of the entire crowd (Figure 2.2). In the following section, some typical techniques of crowd modelling will be introduced and some examples will be given.

- **Physics inspired models.** Several quantitative factors of crowds and pedestrians are measurable. This fact encourages researchers to look for the mathematical models of crowd dynamics.

  Helbing has a series of work upon this topic. His first experiment is in (55), with a stochastic formulation at microscopic level, a gas kinetic formulation at the mesoscopic level, and fluid dynamic equations at the macroscopic level for the crowd model. Later, he (54) proposes another more popular microscopic model: the social force model based on social field theory. The

social force model represents the effect of the environment; it is a quantity that describes the concrete motivation to act. In (56), the model is used to reproduce the emergence of several empirically observed collective patterns of motion. Moreover, simulations of crowd dynamics based on a generalised force model for the escape panic phenomenon are presented in (53). Furthermore, quite a few other works have been developed upon this work. For example, in (32) an additional pattern is introduced by considering the unequal information distribution in a crowd.

In contrast to the former works, macroscopic models often draw upon an analogy between the crowd and a continuum responding to local influence. Hughes (66) is more interested in modelling rational, goal-directed pedestrians. His theory does not govern the behaviours of any individual pedestrians, as it is a macroscopic model; instead, the crowd is divided into (approximate) pedestrian types where pedestrians in each type have the same walking habits.

Physics-inspired models are widely used to study crowds from different perspectives, e.g. to study the effects of introducing autonomous robots into crowds (79), or to model a historic scene (117). The interrelations of the factors and equations (e.g. employing the same factors in different levels of equations) imply the possibility of having a model encompassing all the levels. In addition, the quantitative analysis of crowd dynamics can be relatively easy to adapt into computer-based algorithms.

- **Agent based models.** These qualitative models include employing fuzzy methods to describe the relations of factors and crowd motion instead of pure mathematical methods. Agent-based models use agents to represent the pedestrian or the crowd. Many examples are from the former case, e.g. in (104), crowd individuals have their own emotional parameters to govern behaviour while they belong to a collection of goal-directed groups on a mesoscopic level.

  In (106), the agents are modelled following the concept of non-adaptive behaviours. Non-adaptive crowd behaviours refer to the destructive actions that a crowd may experience in emergency situations. The human and social

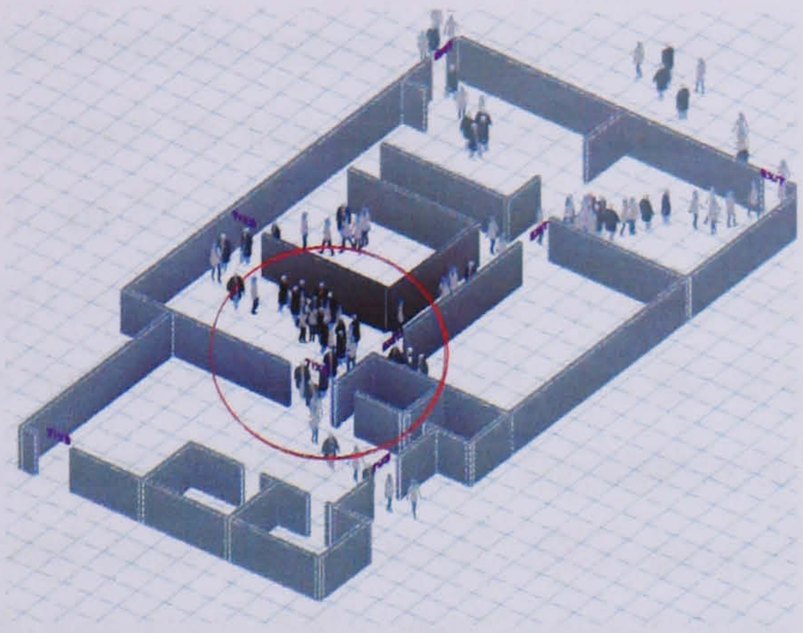


Figure 2.3: A screenshot of XiaoShan Pan's work: human agents try to self-organise into exiting lines.

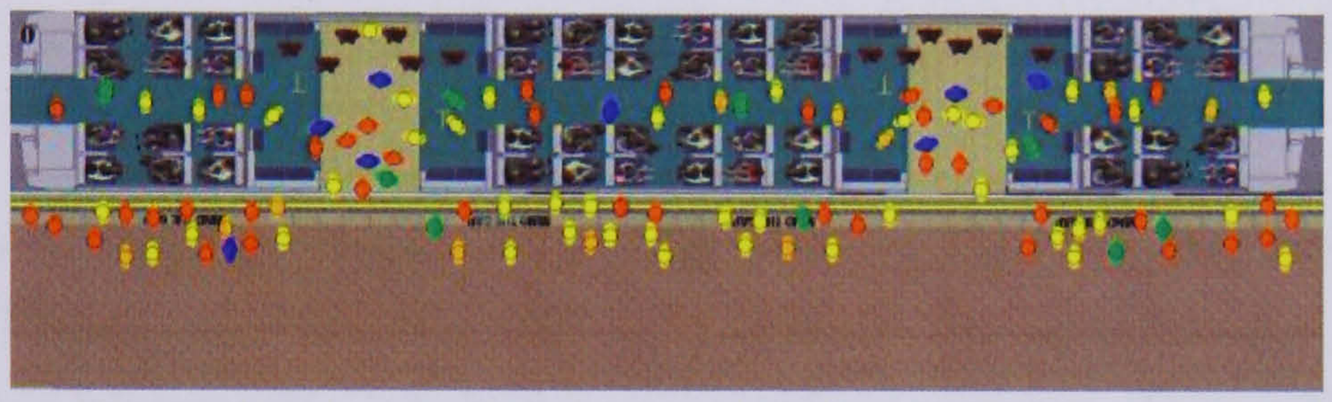


Figure 2.4: Dwell analysis by Crowd Dynamics Ltd, using agents to assess the throughput of specific geometric designs.

models are categorised into the individual, the interactions among individuals, and the group and the environment into three non-independent levels. (Figure 2.3). Brenner et al. (25) provide an example model by assuming that people at the same location experience the same psychological and environmental influences.

Some of the work on agent-based models has already been commercialised, such as the work of Keith Still at Crowd Dynamics Ltd (34) and LEGION international LTD (87) - both provide pedestrian simulations for space design and planning, based on agent technology. For example, the model developed by Crowd Dynamics Ltd aims to simulate how people react to their environment in a variety of conditions (Figure 2.4).

Usually, these examples employ agents to act as individual pedestrians and are only concerned with the microscopic level.

- **Cellular automation models.** Another research approach employs the

construction of local models, where an active area has been virtually divided into cells. An example is the commercialised tool EGRESS, from AEA Technology Plc (3). In EGRESS, the floor area of an environment is covered with cells equivalent to the minimum occupancy area of a person. The used cells represent the free floor area, a wall or a blockage, a cell with a person, or a region with some other attributes. Pedestrians move between the cells, following pre-defined rules. Krez et al. (84) present a model of pedestrian motion using both the floor field and agents. The model consists of three floor fields: *Static floor field* for each cell contains information about the distance to the exit; *Dynamic floor field* changes with the motion of the pedestrians and the third floor field saves the distance of a cell to the next wall.

- **Nature based models.** Some of the models take their inspiration from nature. The emotional ant model (18) extends the psychological information by using biologically inspired ant agents as a crowd. Four different cognitive behaviours of a crowd have been modelled, and transition behaviour is modelled using fuzzy logic.

  Kirchner et al. (78) apply a bionics approach to the cellular automation model by describing the interaction between the pedestrians, using ideas from chemotaxis. The simulation of the evacuation from a large room is also presented to show the ability of the model to represent different types of behaviours.

2.4 Examples of Bridging the Research

Computer simulation can be used to evaluate the developed system's performance. Considering that real visual evidences for abnormal scenarios are rare or unsafe to reproduce in a controllable way, Andrade et al. (12) have developed an approach that generates simulations to allow for training and validation of computer vision systems applied to crowd monitoring. The simulation is generated by a pedestrian path model and a pedestrian body model. Vu et al. (141) conceive a test framework that generates 3D animations corresponding to behaviours

recognised by an interpretation system. In other words, this is a test system for a given interpretation system by generating test animations. Non-vision models can be borrowed for computer vision analysis. Antonini et al. (15) (16) propose a framework using a discrete choice model, which is widely used in traffic simulations, for pedestrian dynamics modelling. The framework models the short-term behaviours of individuals as a response to the presence of other pedestrians. The model is calibrated using data from actual pedestrian movements, manually taken from video sequences. The work is applied to the problem of target detection in the particular case of pedestrian tracking. More recent work by Ali et al. (5) employs the idea of floor field. In order to track a specific individual in the crowd, the probabilities of the directions of instantaneous movement of that person are modelled by taking into the consideration the floor fields. The introduction of the concept of floor field is to incorporate the factors of crowd flow and the structure of the scene. According to the reported results the method works well with very dense crowd. However the test data only contains crowds with unique collective motion, for example three of four testing sequences presented in the paper are videos of marathon running while the forth is of a scene where people are moving out of the train at a station.

2.5 Discussion

This chapter provides a review on current crowd analysis work in computer vision. Perspectives from sociology, psychology and computer graphics are presented, as these research fields have also contributed to an in-depth study on crowd analysis and modelling. Sociological and psychological studies on the crowd phenomenon make use of human observations. Their studies indicate various ways to represent and model people's relationships in isolation and as part of a more or less large group. The microscopic, mesoscopic and macroscopic levels are defined to characterise people as individuals that are part of crowd. The computer vision approach tackles the problem of automatically extracting sufficient information to characterise some special crowd events.

Antonini and Ali give good examples of employing a non-vision model; however, the work only uses very limited information and only acts as a *clever* tracker

at the moment. The works of non-vision analysis presented in this chapter show that all of the factors or information extracted from the real world using computer vision techniques are inter-related. Moreover, they have proposed the probable relationships in their works, which represent the human understanding of crowd dynamics. On the other hand, computer vision techniques have the ability of exploiting special environmental constraints, which could be applied to calibrate the proposed models. The conclusion is that it is possible to develop intelligent systems by combining these works with computer vision approaches. The system would be capable of automatically understanding and modelling the crowd behaviours, which works at both the instantaneous and recurrent levels.

Chapter 3

Group behaviour analysis

This chapter describes the work of behaviour analysis in a group environment. Dynamics estimators are introduced not only at individual level for behaviour classification but also at a global level for estimating the level of cluttering in the scene. Following this, two people counting algorithms are developed based on the detection and tracking of colour patches. This chapter introduces methods that learn scene semantics from group environments, which can be used in a wide range of applications.

3.1 Introduction

Starting from colour modelling and colour tracking, the objective of the work in this chapter is mining semantic meaning from the group environment. The group environment is defined as an environment with up to tens of people. Under these circumstances, the individuals are free to walk all around the scene rather than following certain path constraints by the crowd flow.

The scene semantic and/or automatic event understanding by computer vision has been proposed for different applications; for example (37), where the scene understanding is achieved through human crafted event models, (144) in which behaviour profiles are built that aim for 'anomaly' detection, and for applications like semantic-based retrieving and browsing of a video database - for example, (43)(63) - in which semantic information is employed to cluster and index the

video data. In this chapter, the techniques developed for Ambient Intelligence application and the experiment results are presented.

Ambient Intelligence *(AmI)* is a term used to identify a paradigm to equip environments with advanced technology and computing so that they can respond to the presence of people (116). The AmI paradigm will be able to aid people's daily life and support everyday life activities. For example, intelligent dorms have computer-controlled heating and lighting systems (68), and in-vehicle Ambient Intelligence systems provide assistance to drivers (112). A number of technologies involving modern computing hardware and software are required for the paradigm. Distributed sensors and actuators are employed to observe and interpret users' behaviour and the paradigm aims to learn these users' preferences and adapt the system parameters to improve the quality of life and work of the occupants. Ambient Intelligence has been a popular research topic since the late 1990s; research groups have been founded in both academic and industrial sectors. To mention a few: Kingston University (134),University of Essex (68), Autonomous University of Madrid (8), MIT (7), Mitsubishi Electric Research Laboratories (MERL) (6), Philips Research (10), and NTT Research (9).

The methods presented in this chapter have been developed from two experiments. The first experiment was run in a University laboratory. A number of video sequences were recorded with individuals performing the same action repeatedly. Mixtures of actions were then recorded with more people in the scene, performing either the same or different actions. The trajectories of people were built by skin colour tracking. The second experiment was for an inter-faculty nurse training project where student nurses are being trained in a simulated clinic environment. Computer vision techniques are being introduced to assist the instructors to better understand the behaviours of their students. To identify the different professions in the environment, different colour patches were applied in the scene. The behaviour analysis was based on the tracking of different colour patches. The underlying mechanisms for colour modelling and tracking were the same for both experiments. For both of the experiments, sample colour patches were learned by an expectation-maximisation algorithm and a mixture of Gaussian colour models were built. When processing the video sequence, the colour Probability Density Function (PDFs) of each frame was built according to the

colour models; colour blobs in the frame were picked up by running a connected component algorithm upon pixels with a high probability of being the target colour. The CAMSHIFT algorithm was then used to track the detected blobs over time.

Based on the colour PDFs and the tracking of the colour blobs, the methods discussed in this chapter can provide semantic interpretation of the group environment and act as a prerequisite of ambient intelligence. Trajectories generated from colour tracking are used for individual level dynamics estimators; curvature and corresponding speed are analysed by applying a time window to classify an individual's behaviour. Scene entropy is measured by using the colour PDF and is used as an estimator of the global dynamics. A simple people counting algorithm is implemented by accumulating the colour PDF along the horizontal axis of the scene. The number of peaks of the resulting histogram is counted as the number of individuals. This is a very simple and low-cost algorithm; however, it can only work for limited situations. A more sophisticated method is proposed by establishing spatial relationships between the colour blobs; the blobs that are close to each other for most of their lifetimes are counted as a single colour patch, and the total number of the counted colour patches is regarded as the actual number of the people in the scene.

This chapter is organised as follows: Section 3.2 introduces the nurse training project, Section 3.3 discusses the techniques for colour modelling and tracking, and Section 3.4 presents two dynamics estimators based on colour tracking. Section 3.5.1 proposes methods that aim to count the number of people in the group scene and, finally, Section 3.6 summarises the whole chapter.

3.2 Nurse Training Project

The nurse training project is an interdisciplinary project to aid professional skills' practitioners at Kingston University[1]. The project has engaged the computer vision team in the Faculty of Computing, Information Systems and Mathematics and the School of Nursing at Kingston University.

[1]The research was partially funded by the European Office of Aerospace Research and Development (EOARD) project FA8655-06-1-3013.

Figure 3.1: Pictures illustrating two individual skills, and two instances of a typical simulation.

The School of Nursing at Kingston Hill campus trains student nurses, paramedic and medical students (in a joint degree with St. George's Medical School, London). The training consists of individual and group practical exercises based on taught techniques (Figure 3.1 illustrates examples of individual and team skills), entailing both medical and managerial skills. Group skills are tested in large simulations. During term time, skills training practice is organised in a series of morning and afternoon sessions. Simulations involve a preliminary preparatory round table discussion to introduce the practical exercises, the actual simulation where skills are tested at individual, and team level and a final round table discussion where the strengths and weaknesses of the assessed students as individuals and groups are discussed.

Intelligent algorithms have been studied to enhance and automate the professional training of nurses. The inter-faculty project is the first attempt at Kingston University to design an ambient Intelligence system for use in the training of professionals. In the context of the nurse training project, the paradigm is interpreted as a set of guidelines to develop algorithms capable of interpreting behaviour in a very complex environment, monitored by an array of cameras.

Conventional training of nurses and medical students is very time consuming

and when large numbers of students are involved, it is very hard for an instructor to assess correctly the performance of a student or a group of students. The School of Nursing runs a state-of-the-art training methodology, engaging students in individual and team work. Assessment is usually carried out during practice with on-the-fly verbal feedback and by recording video footage of students' performance. This is discussed in classes to illustrate best practice, encourage less capable students, and praise the best practice of better students. The skills laboratory, situated at Kingston Hill campus at Kingston University, can host up to 30 students at a time with instructors and role players engaged in large simulations. The lab is currently endowed with a variety of medical equipment as well as mobile and fixed cameras. The images in Figure 3.2 illustrate the experimental setup, the large skills laboratory (medically equipped), and a round table example.

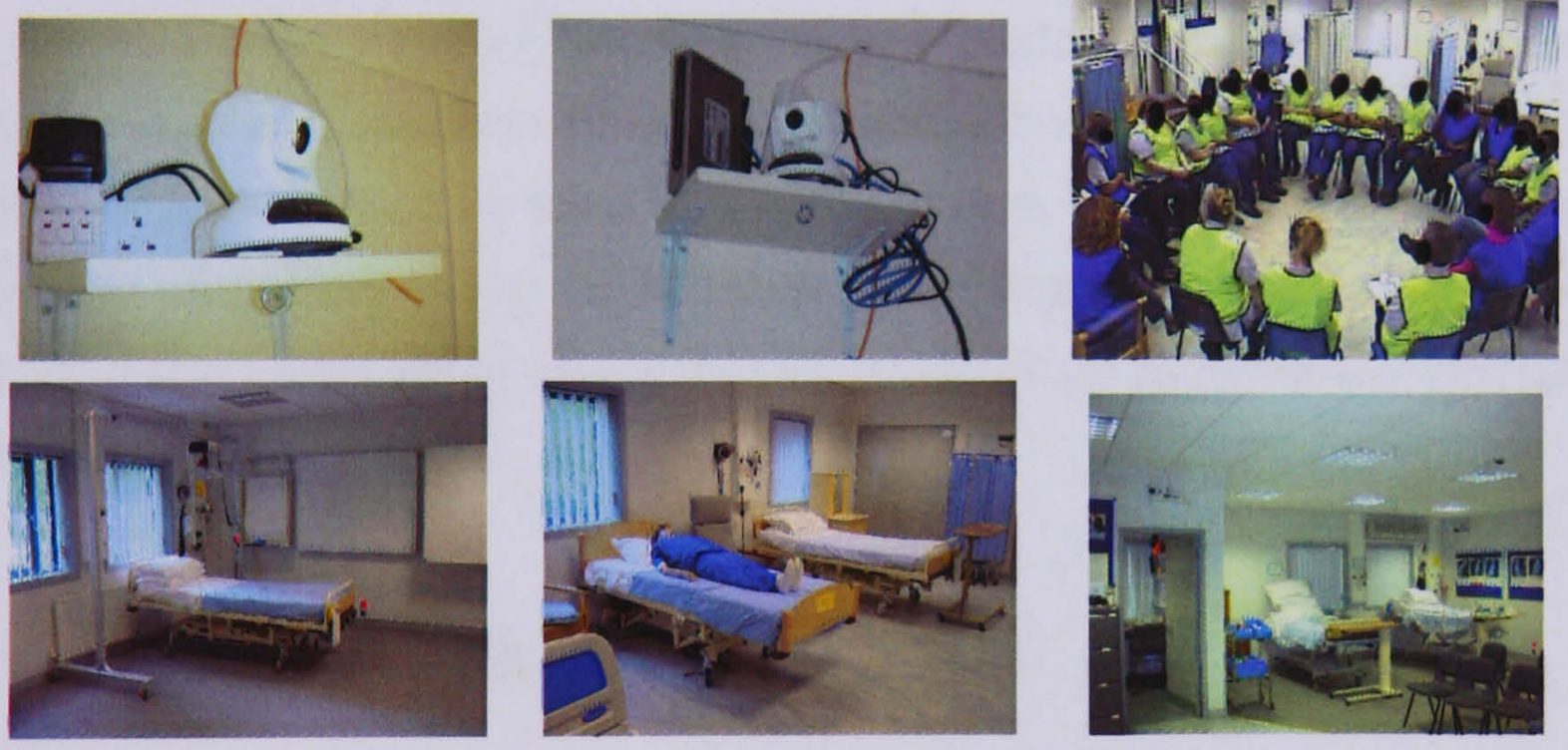


Figure 3.2: Pictures of the experimental setup, including two pan-tilt-zoom (PTZ) cameras, the used router, some views of the skills laboratory and an example of a roundtable meeting.

The inter-faculty collaboration was established in 2001 thanks to a common research interest on human behaviour in complex scenes. Both partners were driven by complementary research interests: the nursing practitioners were interested in an innovative educational methodology using video recordings and the computer vision team were interested in studying algorithms to automatically describe a scene in terms of human dynamics.

The computer vision techniques used in monitoring applications lend themselves well to the automatic understanding of semantics (identification, classification and dynamics explanation of a simulation) in a professional training environment. Automatic understanding of scenes has been studied in (37), where the scene understanding is achieved through the creation of event models, and in (144), where behaviour profiles are built to identify anomalous behaviour. In (43) and (63), semantic information is employed to cluster and index the video data. This application bears a resemblance to monitoring applications, as all scenes are extremely complex and the main goal is to model the nominal behaviour (best practice) and deviations (bad practice). The objectives of this project, described in this chapter, include the identification and classification of role players and algorithms to describe the dynamics in the environment.

The algorithms described in this chapter are tested on video data where all role players in the scene wear a coloured tabard. Four colours are used to distinguish among the instructors (blue), student nurses (yellow), medical and paramedic students (green) and patients (red). The colour coding was introduced to simplify the computer vision processes. Four cameras (pan-tilt-zoom used as fixed cameras) are employed in the experiment. A preliminary study was carried out by analysing the four views independently, attempting to generate the automatic understanding of an evolving scene.

3.3 Colour tracking of people

A colour model can be estimated by acquiring video data of a given colour by using template patches, and via the training of a colour model using the expectation maximisation algorithm (33). In order to optimise the model, colour image data can be studied and an optimal initialisation defined in terms of the number of clusters and initial positions and approximating the functions.

A colour model is fairly robust to changes in illumination but it has the weakness of being specific to the camera used to acquire the training data. In all the tests, a new video camera that is used to acquire video footage has its own colour model. As the training can be performed offline, the limitation is not prohibitive. Colour models were trained for the four different colours used

to recognise the categories of people. For the initial experiment, it is the skin colour. For the nurse training project, these include the student nurse (yellow), the instructor (blue), the patient (red) and the medical student (green).

3.3.1 Colour Modelling

Although background models can work in the group situation, problems arise when people or objects, after a period of time spent moving, stop and become stationary. In such cases, background models might not work well, as they are based on learning adaptation to changes. The learning variable regulates the rate by which the model adapts to changes in the scene: a faster adapting learning rate results in foreground objects and people not being detected if they do not move for short periods of time; a slow adapting learning rate has the opposite effect.

The use of colour information makes possible the identification of people, even if they are not moving. Colour segmentation helps the process of identification of people in the scene, even if they stop and do not move for variable periods of time. A number of colour spaces invariant to illumination were tested, including HSV and YUV (118). A model for each of the selected colours was learned from sample colour images. These models were learned as a mixture of Gaussians, using the expectation-maximisation algorithm. Once the training was complete, a probability density function could be estimated for each trained colour within each frame. Each pixel in a test image could then be probabilistically classified to belong to a specific trained colour. The modes of each density function could be interpreted as an image location with a very high probability of being a person. A connected component algorithm was then employed to build regions with a high probability of representing the target colour patches. Figure 3.3 shows some example colour PDFs from the nurse training project.

3.3.2 Modified CAMSHIFT

In order to track colour patches - identifying people's locations - the CAMSHIFT algorithm has been adapted. The CAMSHIFT algorithm was originally proposed in (24) as an evolution of the MEANSHIFT algorithm (46)(31). CAMSHIFT

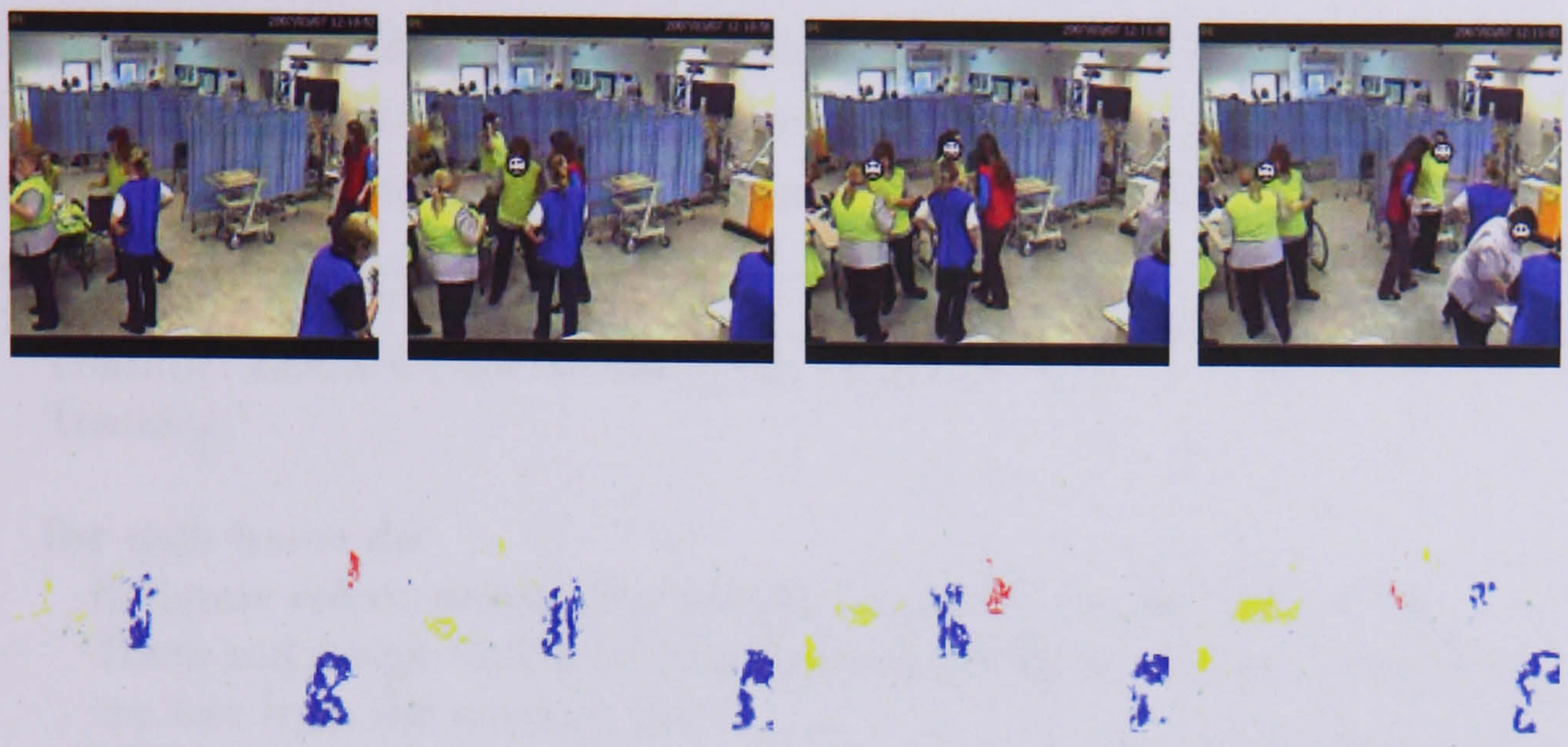


Figure 3.3: Colour PDFs (Example frames from the nurse training project): Top row - the raw frames from the nurse training project; bottom row - the cumulative distribution for the colours.

adapts to evolving a probability density function (PDF) by alternating cycles of the MEANSHIFT algorithm with a resizing of the search window. The window size is a function of the centre of mass of the probability density map (0^{th} moment).

Tracking colour patches entails running the CAMSHIFT algorithm for each patch. However, this is not sufficient to maintain hypotheses in a rapidly evolving scene. That is why the method here keeps track of a list of living patches by tracking them throughout the scene with the CAMSHIFT algorithm, removing those that have too low a probability associated for a number of frames, and introducing new patches whenever sufficiently large new patches appear in the scene with a sufficiently high probability (Algorithm 1). Instead of using a pre-selection of colour patches, the initialisation of the locations of the colour patches is achieved automatically by picking up the high probability patches of the colour PDF generated by the colour model. In addition,in the original CAMSHIFT algorithm the method to generate the colour PDF was to back-project the colour distribution of the initial colour patch to the new frame. In the approach presented here, the colour PDF is generated directly by the learned colour model. This modification to the CAMSHIFT algorithm can improve the tracking under changing light conditions, as the colour models are built from the training dataset from the different

light conditions. Furthermore, this can also reduce the risk of introducing more noise during the tracking process, especially when the mis-tracking happens in several frames. Here, the tracker is instantiated each time a new region of inter-

Algorithm 1 Colour Tracking

Training: Learn Colour Model $\sum_i w_i \cdot \{\mu_i, \Sigma_i\}$
Tracking:

for each frame **do**
  Estimate colour probability density images by the learned model.
  Track and assign likelihood to colour patches in the L_{track}; remove those that are lost from the tracking list L_{track}.
  Identify new colour patches (connected components) and add them to the tracking list L_{track}.
end for

est is recognised by the colour model, while regions with very low probability are removed from the list of active regions. The trajectories of the tracked objects and people can then be parametrised and annotated. The current implementation goes as far as a spline fitting of the trajectory. Automatic annotation, not present in the current implementation, can be thought of as an additional process capable of providing the user with a natural language description of the scene in terms of people and objects present in the scene, their location, trajectory trends and their interactions.

Figure 3.4: Tracking of colour patches, example frames from the nurse training project. First row: tracking of three student nurses (yellow patches) in a relatively simple scene; second row: tracking of a patient (red patch) in a complex scene.

Figure 3.4: Tracking of colour patches, example frames from the nurse training project. Tracking from another camera view.

3.4 Estimating Dynamics

The main goal of this research is to automatically estimate the dynamics in a complex scene, frequented by an unspecified number of people. Dynamics is estimated in two levels of detail: the individual level and global level. For an individual level scene, dynamics is represented by different classes of people's behaviour, i.e. in this case, the degree of interest of people to the environment. At global level, scene dynamics is judged by the distribution of people (represented by the detected colour patches) in the scene. This is important in applications where situational assessment is crucial to better inform people inhabiting a specific environment. For instance, in a shopping mall or in a museum, individuals and more or less large groups of people might pass or stop by to window shop or observe an exhibit. In such cases, information about the merchandise or exhibit could be delivered in a more efficient manner, for instance, with the aid of a robot. Automatic estimation dynamics is also important in crucial situations where people must be informed of exits and escape routes or where people's behaviour can be suspicious or dangerous.

3.4.1 Trajectory

This method offers a means to classify people's behaviour as "interested" or "uninterested" in the scene. One can then imagine the degree of interest in a scene being used to inform a robotic platform to deliver a specific message to the user. The video data was captured in a simulated exhibition environment. Actors played different types of behaviours as they would act in a real exhibition. They stopped at the exhibits and looked around at the exhibits to show an interest. With uninterested behaviour, the actors just moved across the scene without stopping. Actors also pretended to be panicking in an emergency and walked in the scene swiftly and desperately looking for the exit(s). The last behaviour is referred to as "animated" behaviour in this section.

Skin colour is used to extract exposed patches of the human body and these patches can be robustly tracked throughout a scene. The tracks are then employed to annotate the dynamics of individual patches and draw some qualitative and quantitative descriptions of the global evolution of the scene.

Figure 3.5: The two above frames show the low curvature trends of uninterested behaviour: when people pass by an exhibit.

Trajectories of skin patches identify people's trajectories and can be seen as signatures of their behaviour. For instance, people interested in exhibits in a museum or merchandise in a shopping mall have a more irregular signature, distinguished by a curvature that becomes higher and changes more frequently as the patches represent people looking around at an object.

The amount of time spent in the scene also plays an important role: the shorter the time, the smaller the interest shown in the exhibit/object. Frames in Figure 3.5 illustrate two examples of uninterested behaviour, well correlated with a smoother (low curvature) trajectory, while frames in Figure 3.6 clearly illustrate how the interest in an object is correlated with a change in curvature. Dynamics can therefore be estimated by studying the trajectories of the tracked skin colour patches and making use of their trends. An in-depth study of the trajectories led us to the following conclusions, all based on the assumption that the extracted skin patches belong indeed to people in the scene:

- Fast patch movements indicate that people in the scene are moving rapidly:

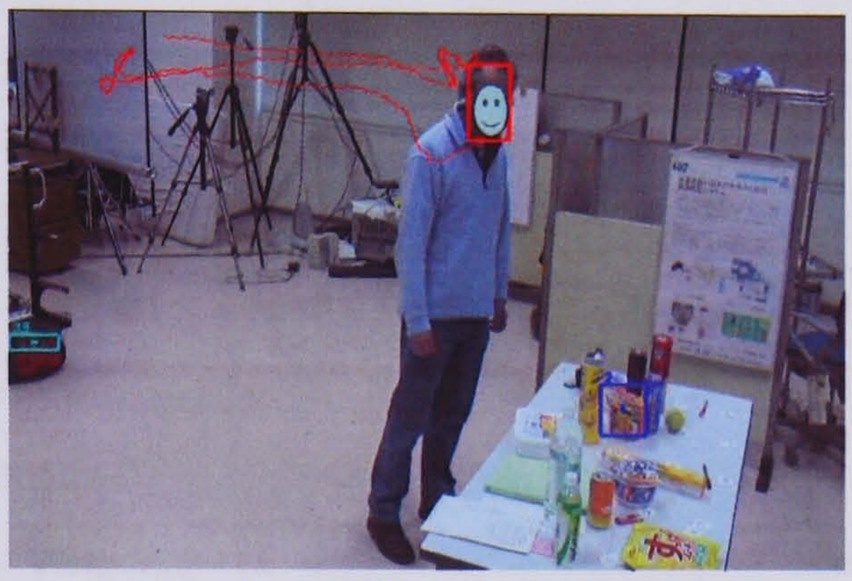


Figure 3.6: The above frames show when people are interested in the shown exhibits and they stop by the exhibit. Trends of such trajectories have a higher curvature.

the speed of each patch is estimated by the distance in pixels of a patch between frames.

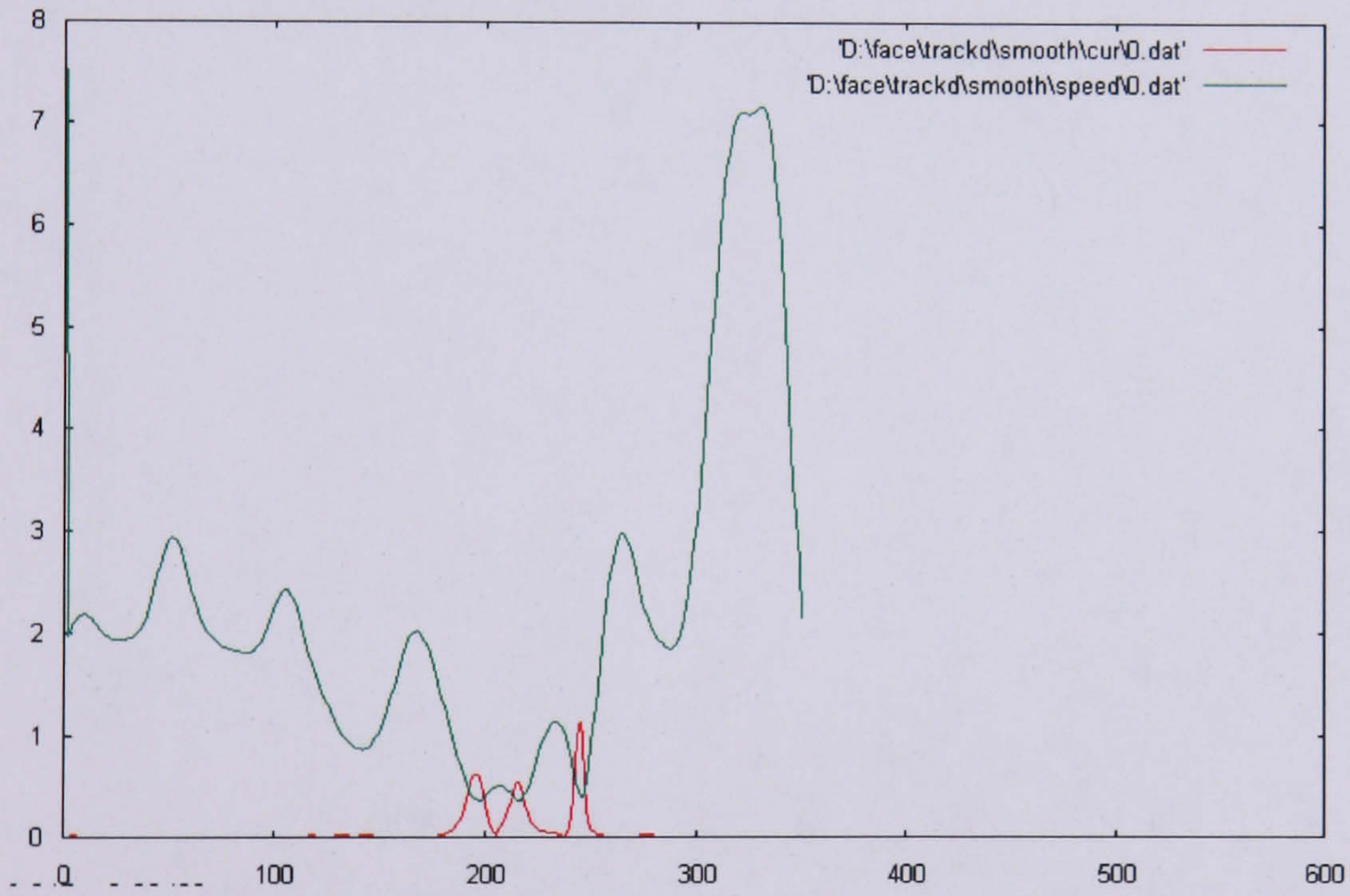
- The curvature of a trajectory is a good indicator of how many twists and turns the trajectory trend has. Changes in curvature might occur more frequently in some moments than in others: the frequency of change and the magnitude of curvature is an indicator of the person's interest in some parts of the scene.

- A density signature of curvature peaks can therefore be estimated to describe people's interest: the higher the density, the higher the attention a person has for an object. This study demonstrates that highly interested people will stop and move about in front of the object, whilst uninterested or vaguely interested people will move a lot in the scene and rarely stop - their curvature signature shows trends with a small number of high peaks. A suitable time window is defined to estimate the density: typically, a number of seconds are usually spent by a person observing an object in the scene. This parameter depends on the application and can be learned.

Figure 3.7 illustrates the speed and curvature trends of a patch used to train the model of uninterested people. The speed becomes fairly high; however, the curvature remains lower than a low threshold, which is typically around 1. Figure 3.8 illustrates the signature of a patch related to a person who is interested in the scene. The speed is lower, indicating the person pays more attention to the scene. The frame in Figure 3.8 clearly shows the close occurrence of curvature peaks in two points of the scene, indicating that the person stopped and then looked around for a while before moving to the next area of interest to stop again and observe, before leaving the scene. The yellow trajectory - associated with a hand was picked up too late to illustrate the curvature phenomenon typical of an interested behaviour.

Figure 3.9 illustrates what are known as uninterested and animated behaviours, characterised by patches of people uninterested in the scene objects, but where those people stay for longer in the scene and move about without really focusing on any object and do not stand still in any particular position of the scene.

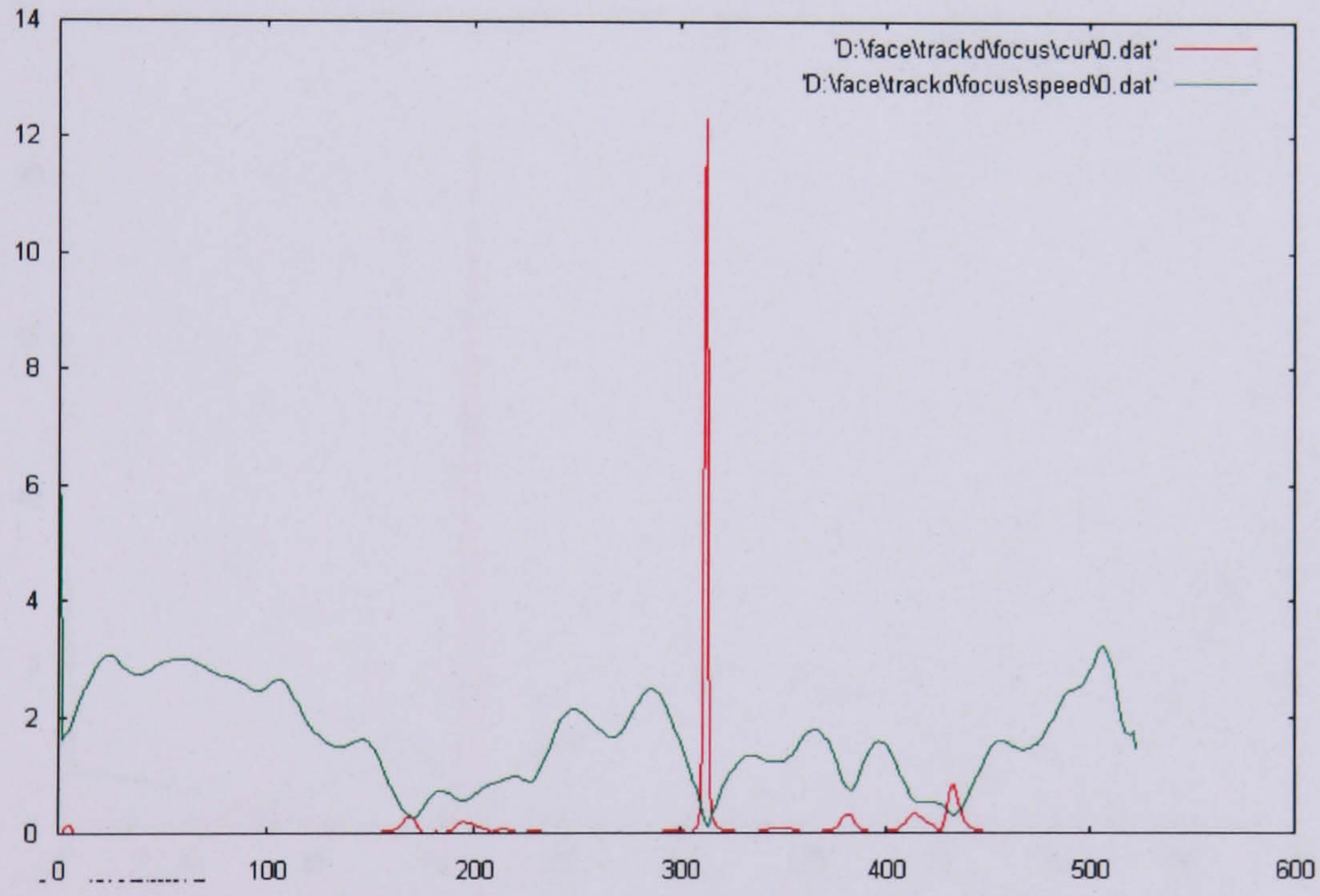
(a) Trajectories of uninterested people



(b) Speed (green) and curvature (red) of the above trajectory

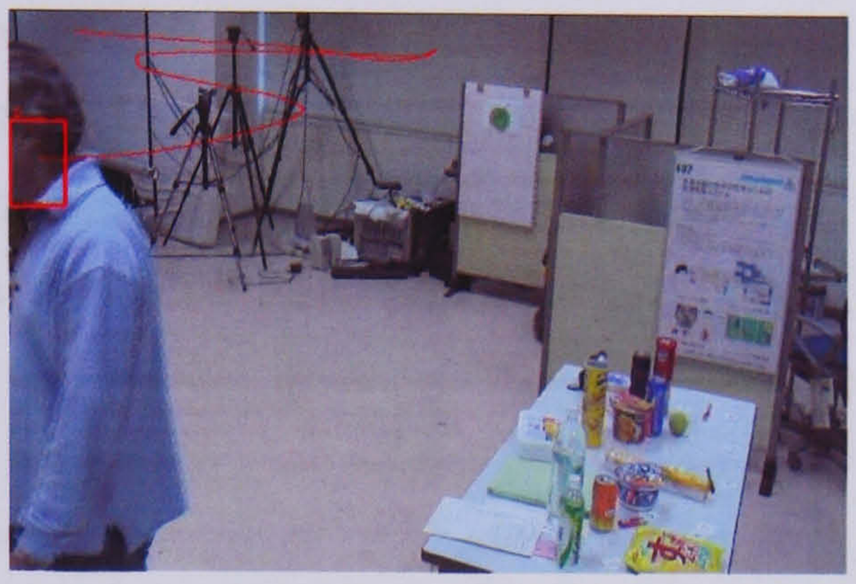
Figure 3.7: An uninterested behaviour: people who are not interested in the exhibit, passing by without stopping.

(a) Trajectories of interested people

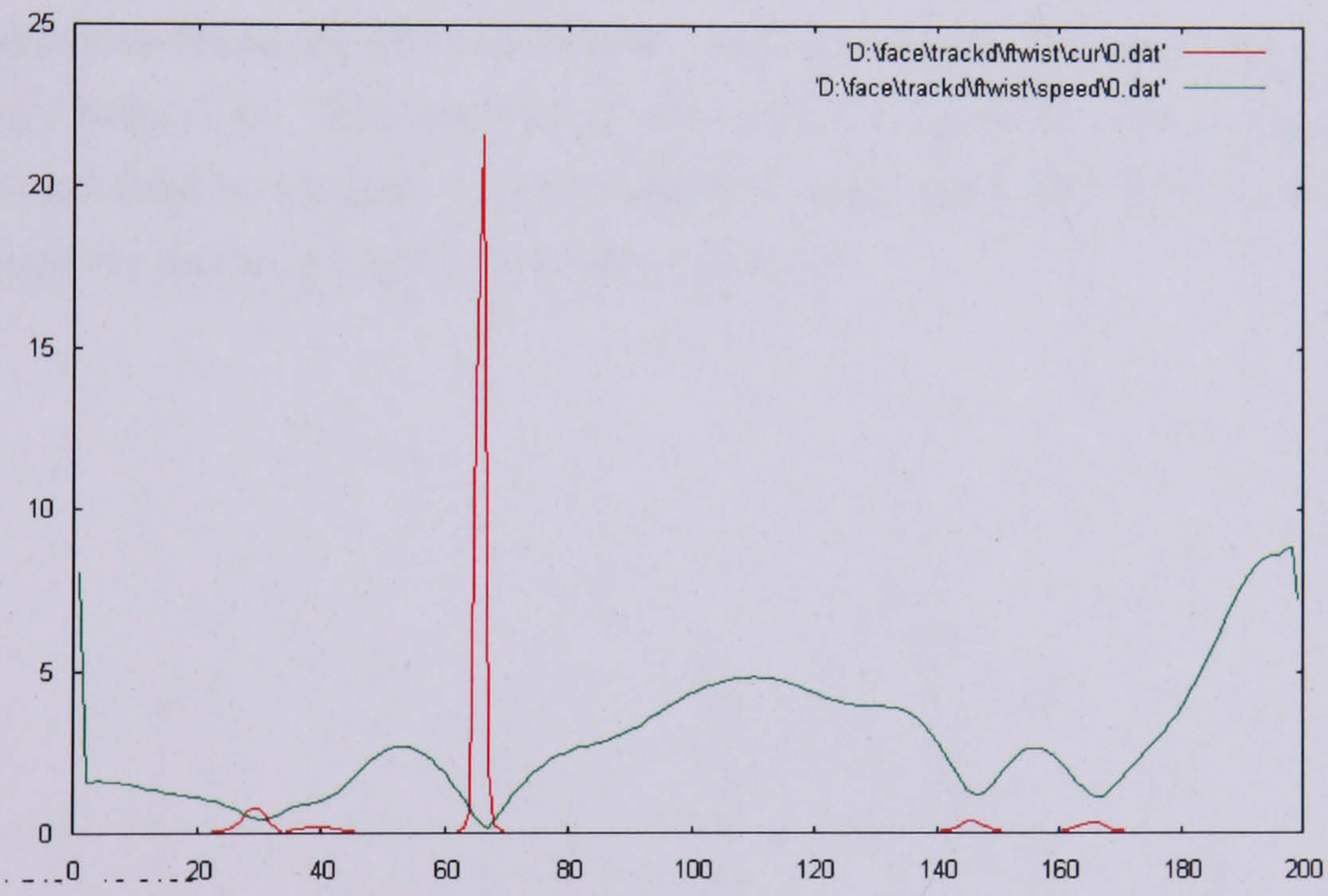


(b) Speed (green) and curvature (red) of the above trajectory

Figure 3.8: An interested behaviour: people who are interested in the exhibit, stopping and looking at the exhibit.



(a) Trajectories of a person with animated behaviour and uninterested in the scene



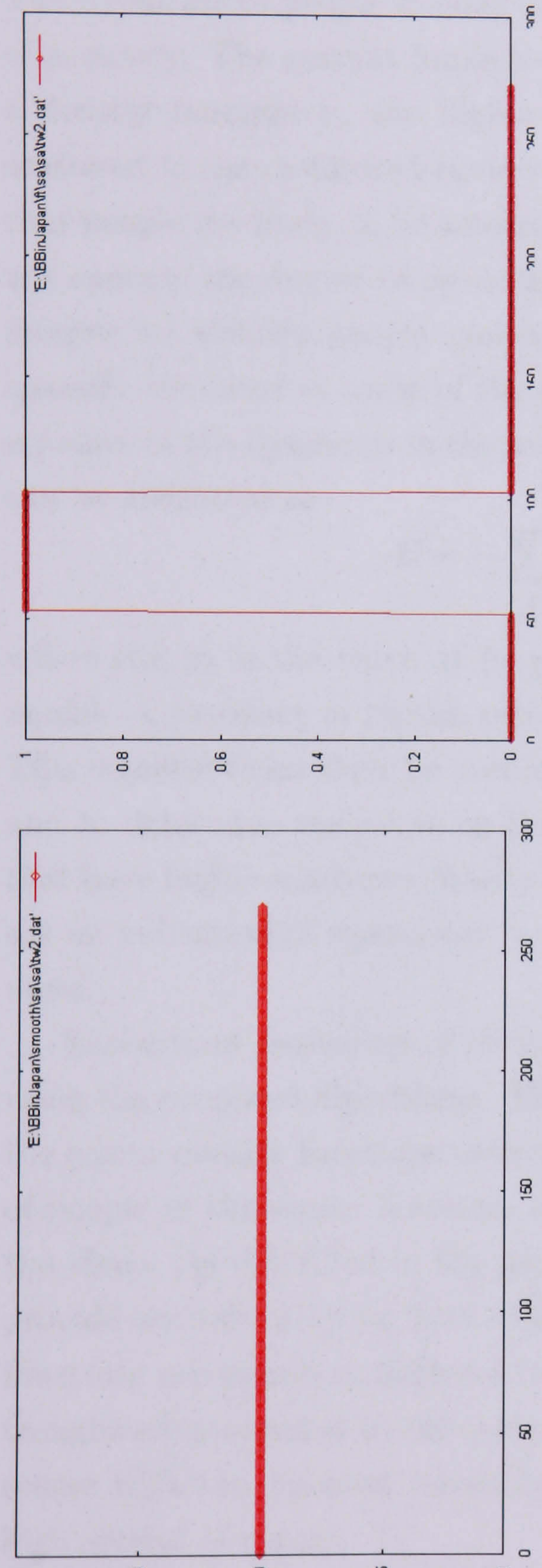
(b) Speed (green) and curvature (red) of the above trajectory

Figure 3.9: An animated behaviour behaviour: people who stay for longer in the scene and move about without really focussing on any object and not standing still in any particular position of the scene.

As can be seen in the above graph, such behaviour shows a large number of sparse high curvature peaks and also correlates with a higher speed, indicating that the person did not stop for longer than the short period of time required to change direction in the scene a few times and then leave the scene.

The graphs in Figure 3.10 illustrate how the density of curvature maxima can be employed to disambiguate between interested, uninterested and animated behaviour. The graphs clearly indicate that whenever a person is interested in the scene objects, then they stop and spend time looking. While they do so, they move about, building up a density of curvature maxima. Completely uninterested behaviour shows no density at all for maxima above a threshold estimated by measuring the mean curvature of patches of uninterested people. Finally, animated behaviour builds some density which, however, is not comparable with the density built for a focused behaviour.

Experiments illustrate that curvature can be employed to analyse trajectories and classify behaviour. The amount of skin colour patches in the scene and their life spans can shed some light on the clutter in the scene, and their dynamics can be employed to assess a highly changing situation.



(a) Uninterested behaviour.

(b) Animated behaviour.

(c) Interested behaviour.

Figure 3.10: Examples of curvature density

3.4.2 Entropy

The dynamics of people moving in a scene can be estimated at different degrees of accuracy. The current implementation makes use of the idea that the sparser a density function is, the higher the probability is that the scene has people scattered in the monitored environment. The degree of scatter does only identify that people are likely to be attending to their tasks around the scene, but it does not capture the degree of dynamics in the scene. In fact, it might be of greater interest to identify people moving frantically in the scene rather than people sparsely allocated at areas of the practice skills laboratory. Entropy is used as a measure of the dynamics in the scene (135)(17). At each frame, an entropy value can be measured as

$$E = -\sum_{x}\sum_{y} p(x,y)\log p(x,y) \tag{3.1}$$

where $p(x,y)$ is the value at (x,y) of the colour PDF generated by the colour model. A sequence of frames provides a signature for a given view of the scene. This signature can then be studied to establish an entropic metric (a baseline) and to determine variations in the signature. Peaks in entropy describe frames that have highly scattered density functions, and rapidly changing entropy values are an indication of movement in the scene from scattered to compact, and vice versa.

Excerpts of sequences of interesting behaviour were extracted and analysed using the proposed algorithms. The testing was organised as follows. First of all, the colour density functions cannot be sufficient to identify correctly the number of people in the scene; however, they are sufficient to estimate the dynamics in the scene. As described in the previous section, entropy is used as a measure to provide an indication on how crowded a scene is. Figure 3.11 and Figure 3.12 illustrate two scenes at different time stamps and the related entropy. Peaks and troughs are associated to the corresponding scene frame. Troughs tend to identify scenes with the compact assembly of people, while peaks refer to scenes with a high spread of people.

(a) Frame 0

(b) Frame 57

(c) Frame 195

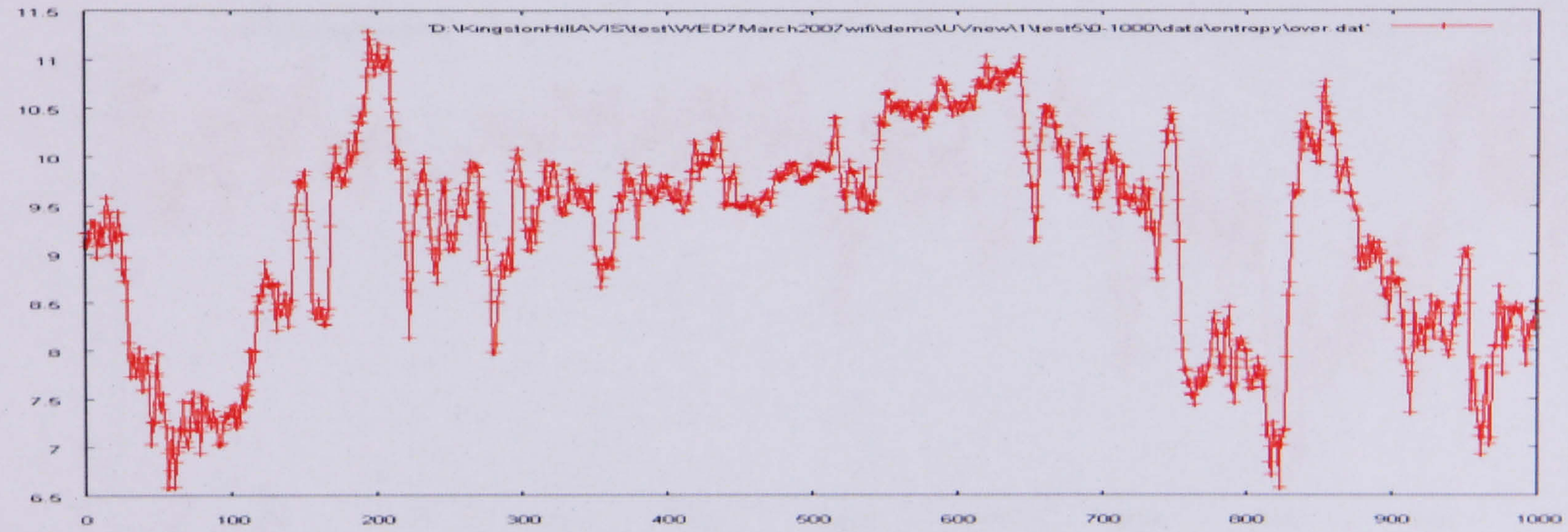
(d) Frame 500

(e) Frame 645

(f) Frame 764

(g) Frame 907

(h) Frame 1000



(i) Entropy

Figure 3.11: Dynamics signature of scene A

(a) Frame 0

(b) Frame 130

(c) Frame 350

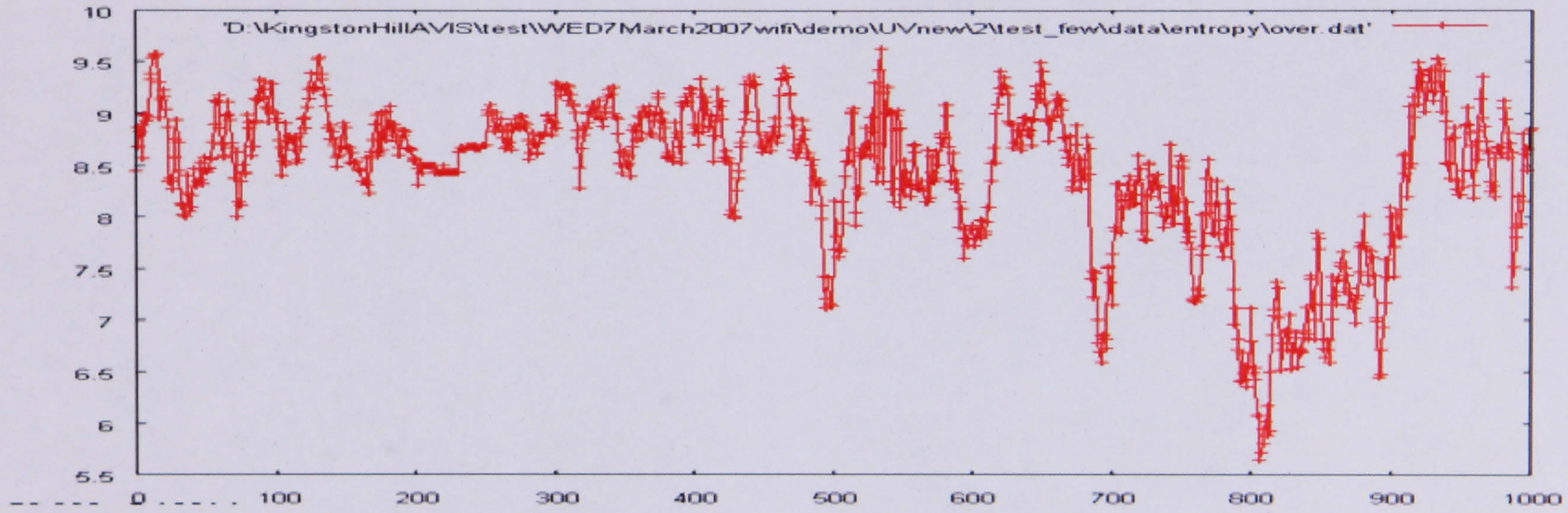
(d) Frame 535

(e) Frame 653

(f) Frame 807

(g) Frame 930

(h) Frame 1000



(i) Entropy

Figure 3.12: Dynamics signature of scene B

3.5 Counting People

Colour segmentation generates fragmentation by identifying one person with more blobs. Segmentation could also cause false groupings by clustering together more people in close proximity. Both problems are due to occlusions (between people and objects) and self-occlusions (between people body parts), as well as by the reflections of artificial illumination on the monitored person.

In this section, first, a simple algorithm is described that can provide a qualitative counting of the people acting in the monitored scene. Then, a more elaborate algorithm is introduced, whose performance is also quantified using conventional performance measures. In the second counting algorithm, spatial relationships group the blobs split from a single person. At first, a graph is created for each frame with links between all identified blobs. Each link is then evaluated to judge whether the linked blobs should be merged into a cluster to recover an individual or whether they should be kept separate - making the assumption that both blobs are disjointed and likely to be parts of different people in the scene.

3.5.1 A Simple Algorithm

As mentioned previously in this chapter, one of the main problems caused by the colour segmentation is the fragmentation or over-segmentation of people in the scene. When a person is close to the camera, he/she is usually represented by a number of blobs bearing the same colour.

One way of solving the problem is by grouping the blobs by using a proximity constraint. A first attempt at providing the user with a rough count of people in the scene can be done by employing an accumulator along the horizontal axis of the scene. Such an accumulator will accrue information of the existing blobs of a given colour. The counting simply vertically accumulates the contributions of each blob and adds such contributions to the accumulator. This is illustrated in Figure 3.13. The rationale is that more blobs in close neighbourhood contribute to peaks in the 1D signature, and that the likelihood of blobs belonging to people next to one another is lower than the likelihood of belonging to the same person. The algorithm simply accumulates over time the blobs identified in the video sequence and it normalises the signature to a given maximum height. The

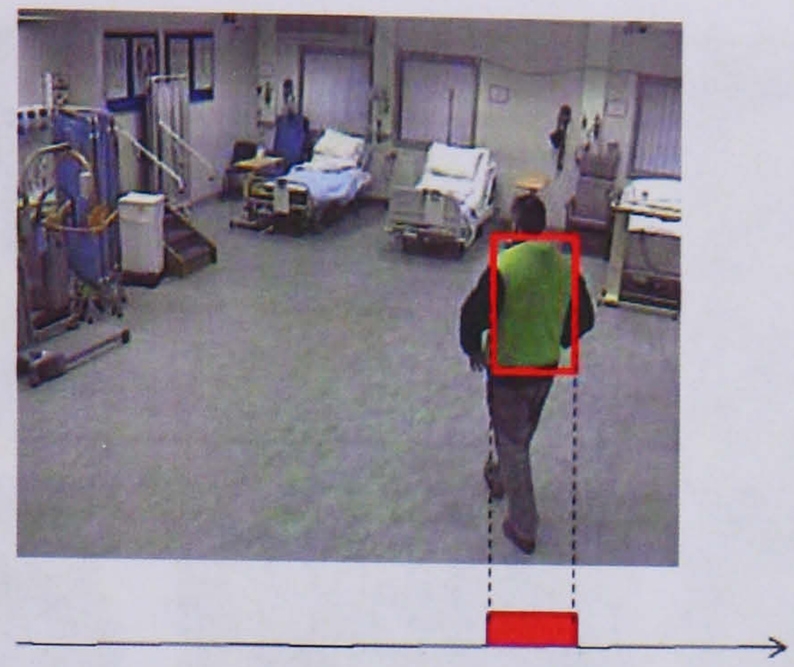


Figure 3.13: The bounding box of a blob representing a person or part of a person is collapsed onto the horizontal axis. This will contribute to the profile of the scene for that specific category of people.

signature is then smoothed a few times with a Gaussian filter, and the modes are identified on this signature as the highest peaks. The signature works effectively as a probability density function of the presence of blobs in the scene. Peaks that are suboptimal as they are closer to higher peaks are eliminated, removing false alarms, and peaks that are sufficiently close are merged together by the Gaussian smoothing, effectively integrating information.

By no means can this be claimed to be a perfect method. In fact, it clearly suffers from the loss of vertical dimension, collapsing each blob vertically and therefore losing the information of *how far* a person is in the scene. The algorithm also underestimates the people count by suppressing peaks that may be small, but still identify the presence of a person in the scene. The sparseness of blobs when segmenting a person could also introduce noise and identify more people than there are in the scene.

Figure 3.14 illustrates the pros and cons of the developed algorithm. In the following, the frames in Figure 3.14 are referred to using an incremental numbering, starting from the top left with frame 1. In frames 1, 3 and 15, people are isolated and, thus, the algorithm is successful. In frame 11, for instance, the colour segmentation fails and introduces false alarms, which are in turn identified as peaks in the related PDFs. Frames in which people are at different distances from the camera - but not aligned - can be correctly interpreted as shown in frames 13 and 14. In other cases, the algorithm fails to perfectly disambiguate

Figure 3.14: From top left to bottom right, frames are numbered frame 1 to frame 15. The above figure illustrates fifteen frames. The frames include the bounding rectangles, detected by the colour tracker, and the profiles representing the probability density functions of the defined categories of role players. The white vertical lines illustrate the detected peaks, corresponding to an estimate of the modes. Each mode represents a person in the monitored scene.

aligned people, as shown in frames 2 and 9. The algorithm might fail to detect people in the scene due to illumination problems or because people are too far from the camera, as shown in frame 1.

3.5.2 Graphs of Blobs

Graphs are generated from the previously detected blobs. The nodes in the graph represent blobs while the links in the graph joining the pairs of blobs represent the spatial relationship between the two blobs. The creation, deletion and updating of the links are required to be automatic according to the change of the situation. The algorithm links are enforced between a blob, say A, with all the other blobs in the scene during its life cycle. During the life cycle of A, another blob, say B, could appear in the scene and then leave the scene. Under such circumstances, A should then be linked to B once B has entered, and the link should be eliminated right after B has left the scene. The complexity of the problem increases when the number of people involved increases. The creation of the links is triggered by the appearance of blobs, deletions are triggered by the disappearance of blobs, while updating is carried out at regular intervals every Δt, taking into consideration all the blobs at that moment in time.

Following in line with the above example, a link is created between A and B when B enters the scene. The link should be kept updated while B is in the scene. The link should then be removed when B is no longer in the scene. For algorithmic simplicity, a link is bi-directional, so each link between blob A, for instance, and any other blob also implies that all linked blobs keep track of the existence of A. When a blob leaves the scene, it sends a signal to all the links connected to it to release and delete them. At each frame sampled at a given Δt, the system checks the blobs to create, delete or update the existing links. Algorithm 2 illustrates this process.

3.5.3 Estimation of Distance Between Blobs

Spatial relationships between blobs are represented as distance information contained within the links connecting the nodes. The distance between blobs is calculated as the Euclidean distance between the blobs' centres. Because of the

Algorithm 2 The creation, deletion and update of the links

if objects: $O^-_{0...m}$ are leaving the scene **then**

  for $i = 0$ to m **do**
    Object O^-_i send signals to all the links connected with it
    Delete O^-_i
  end for
end if
Delete links with signals
if objects: $O^+_{0...n}$ are entering the scene **then**

  for $j = 0$ to n **do**
    Build links between object O^+_j and all of the existing objects in the scene
  end for
end if
Update all of the existing links

perspective distortion, the absolute value of the Euclidean distance cannot be used to estimate the spatial relation between the blobs. For instance, two blobs at an absolute distance of 50 pixels could be close to each other when they are in front of the camera, while they could be far from each other when they are distant from the camera. Hence, a method for calculating relative distance by comparing the absolute distance with the size of connected blob has been proposed here, i.e., the ratio of the absolute distance and the blob size was used. In this method, the variation of dimensions of blobs at different locations is considered. The Euclidean distance used as the absolute distance between blob i and j is as below:

$$D_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{3.2}$$

where (x_i, y_i) and (x_j, y_j) are the coordinates of centre points of blob i, j, respectively, and the temporal relative distance of blob i and j is calculated as:

$$d_{ij} = \frac{D_{ij}}{\sqrt{w_k^2 + h_k^2}},\ k = \begin{cases} j,\ if\ y_i - 0.5h_i < y_j - 0.5h_j \\ i,\ otherwise. \end{cases} \tag{3.3}$$

where w_k and h_k are the width and height of the blob. The denominator is a measure of the size (its diagonal) of the blob and is used as a weight, as a compensating factor for the link.

The above calculations are carried out in a single frame. A temporal average operator has been applied every Δt frames for each distance calculation. This operation can reduce the instability caused by the tracking algorithm, thus the video sequence has been sampled at fixed regular time intervals, i.e., each time segment contains distance information for Δt frames. Equation (3.4) describes the calculation of this distance,

$$\overline{d}_{ij}(T) = \frac{1}{\Delta t} \sum_{\Delta t} d_{ij}(T - \Delta t) \tag{3.4}$$

so the distance between blobs i and j at time T is the average of the distances over the previous Δt frames. The main reason for this temporal smoothing operation is to stabilise the distance. Δt is a short time interval. For example, in this case an 8-frame Δt is used, which is equivalent to 0.5 seconds.

3.5.4 Temporal Pyramid for Distance Estimation

Short-term spatial relations are not sufficient for clustering blobs. The temporal pyramid of a distance scheme has been introduced to maintain longer term distance information. In this algorithm, two blobs belong to the same cluster if they are close to each other during their life span. A coarse pyramid was used, where the current time frame is represented by the top of the pyramid, while the whole lifetime of the blob and half of its lifetime represent the other two layers. For each pair of blobs, the algorithm takes into account the distance information from each level of the pyramid and calculates the overall probability that they belong to the same cluster. This scheme is based on an assumption that two persons are not likely to stay next to one another for a very long time period. This is clearly not true in general, but it suits the application of nurse training where nurses, instructors and medical students are continuously moving about.

The temporal pyramid consists of three levels: the bottom layer holds the overall distance information between two blobs from their appearance in the scene

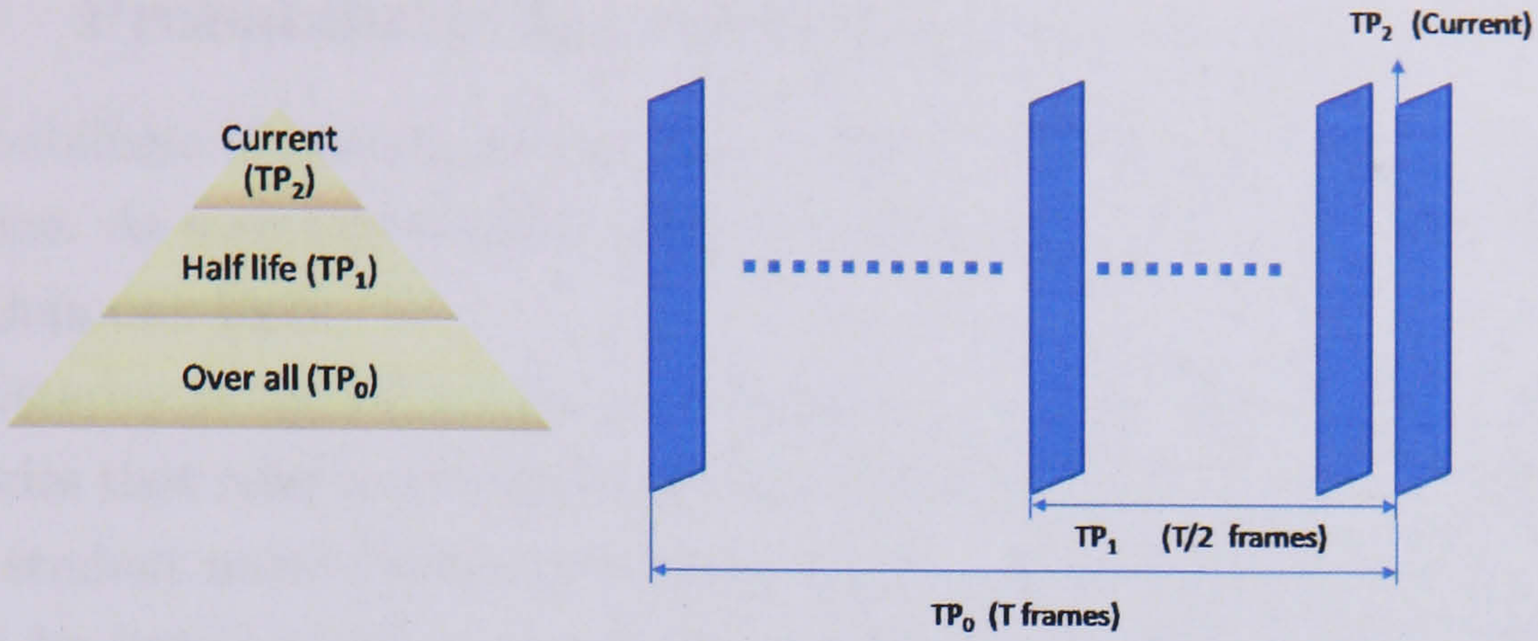


Figure 3.15: Temporal Distance Pyramid: The bottom layer represents the overall distance information from time 0 to time T, the middle layer represents the distance information from time $\frac{T}{2}$ to T and the top layer holds the distance information for the current time slice T.

to the present time, the top layer holds the present distance information and the middle layer holds the information from the half time to the present. This is illustrated in Figure 3.15. The generation of the temporal distance pyramid is:

$$TP_0(T) = \bar{d}(0 \rightarrow T) = \frac{1}{T}\sum_{t=1}^{T}\bar{d}(t) \tag{3.5}$$

$$TP_1(T) = \bar{d}(T/2 \rightarrow T) = \frac{1}{\frac{T}{2}}\sum_{t=\frac{T}{2}}^{T}\bar{d}(t) \tag{3.6}$$

$$TP_2(T) = \bar{d}(T) = \bar{d}(T) \tag{3.7}$$

where $TP_0(T)$ to $TP_2(T)$ represents the distance information held from the bottom layer to the top layer at time T. In practice, to reduce the redundant calculations of the top layer ($TP_0(T)$) and middle layer ($TP_1(T)$), a recursive method has been employed and the equations are modified as follows:

$$TP_0(T) = \frac{1}{T}(TP_0(T-1) \times (T-1) + \bar{d}(T)) \tag{3.8}$$

$$TP_1(T) = \frac{1}{\frac{T}{2}}(TP_1(T-1) \times \frac{T-1}{2} - \bar{d}(\frac{T}{2}-1) + \bar{d}(T)) \tag{3.9}$$

$$TP_2(T) = \bar{d}(T) \tag{3.10}$$

3.5.5 Probabilistic Estimation of Groupings

A probabilistic clustering scheme was devised to eliminate over-segmentation in the scene. As mentioned earlier in the chapter, one person may be identified with more than one blob.

Clustering is carried out for each category, so, if two blobs belong to colours / categories that refer to two different role players, for instance an instructor (blue) and a student nurse (yellow), then their link has a probability of zero and they cannot be linked to the same graph. In all other cases, spatial relation is the main criterion used for clustering. This means that the probability associated with the link between blobs is inversely proportional to their Euclidean distance. This rule is represented by a function $\varphi(\overline{d})$:

$$P(\overline{d}) = \varphi(\overline{d}) = \begin{cases} 1, & when \ \ \overline{d} = 0 \\ 1 - \frac{1}{\theta_d} \times \overline{d}, & when \ \ 0 \leq \overline{d} \leq \theta_d \\ 0, & when \ \ \overline{d} > \theta_d \end{cases} \tag{3.11}$$

where θ_d is the threshold of distance. When the distance falls below this value, the probability of clustering is equal to 0. When the distance is equal to 0, the probability is equal to 1. The probability of clustering two blobs with a distance that falls between 0 and θ_d is interpolated with a linear function. Each layer of the temporal distance pyramid provides a probability of clustering and the outcome of the three layers has been averaged as follows:

$$P_{dis} = \frac{1}{3}(P(TP_0) + P(TP_1) + P(TP_2)) \tag{3.12}$$

The overall size of the blobs is also used to bias the probability of clustering blobs. A linear approximation of the blob size at different locations of the scene has been used as a reference. The size of the overall bounding box between blobs is compared against the estimated reference, according to their locations. This comparison is represented by the ratio:

$$\overline{s} = \frac{S_o}{S_r} \tag{3.13}$$

where S_o is the size of the blobs and S_r is the reference size from the linear approximation. The probability of clustering by area is calculated by: $\varphi(\overline{s})$:

$$P_{size} = P(\overline{s}) = \varphi(\overline{s}) = \begin{cases} 1, & when \ \ \overline{s} = 0 \\ 1 - \frac{1}{\theta_s} \times \overline{s}, & when \ \ 0 \leq \overline{s} \leq \theta_s \\ 0, & when \ \ \overline{s} > \theta_s \end{cases} \tag{3.14}$$

where θ_s is the threshold of the ratio of the size $(\overline{s})$. $\varphi(\overline{s})$ is employed for the reason that smaller fragments should increase the probability to cluster. The overall probability of clustering is:

$$P = P_{dis} \times P_{size} = P(\overline{d}) \times P(\overline{s}) \tag{3.15}$$

3.5.6 Grouping Blobs

For each frame, the clustering takes place in two steps, which are named as *pair clustering* and *sub-clustering*. Pair clustering checks all pairs of blobs, clustering together all the pairs with high probability. This rule ensures that all the blobs that potentially belong to the same person are clustered together. If two blobs are selected to be clustered and they already belong to two clusters, then the clusters can be merged, as shown in Figure 3.16(a). Pair clustering may generate

(a) A frame in which multiple blobs (illustrated with a black oval) should be clustered together.

(b) A frame in which blobs belonging to different people could be clustered together (illustrated with a black oval).

Figure 3.16: Two frames of problems in clustering.

bad clustering. In fact, blobs that belong to different people could be clustered

together as shown in Figure 3.16(b). The second step - sub-clustering - is used to obtain the scores of different numbers n $(1 \leq n \leq N)$ of sub-clusters of a cluster C which contains N blobs. In a cluster generated in the pair clustering step, each pair of blobs is associated with the probability of clustering, which is generated by the method described in Section 3.5.5. The strength Γ of a cluster is defined as:

$$\Gamma = \frac{1}{C_N^2} \sum_{i=0}^{C_N^2} P_i \tag{3.16}$$

where N is the total number of blobs, so there are C_N^2 pairs of blobs. *Connected* and *Unconnected* are defined as the pairs of blobs with a probability of clustering respectively higher and lower than a given threshold. Creating sub-clusters requires that every time the weakest *Connected* link is removed, the blobs are re-clustered by the remaining *Connected* list. The score of the operation is equal to the energy cost E of removing the *Connected* list and the related *Unconnected* list.

$$\Lambda = \frac{1}{n} \sum E + \frac{1}{m} \sum \Gamma \tag{3.17}$$

where the energy cost of removing a *Connected* list with the probability of clustering P is:

$$E = 1 - P \tag{3.18}$$

This operation is continues until all the *Connected* are removed; meanwhile, all the blobs are isolated. Figure3.17 shows an example of the sub-clustering process of a cluster containing four blobs.

During the operation, the scores are accumulated for different numbers of sub-clusters. In this case, the number of sub-clusters with highest score is selected to be added to the person-count and the sub-clusters are regarded as individuals. The total number of people is the sum of the selected numbers of sub-clusters of all the clusters in the frame.

3.5.7 Experimental Results

The counting algorithm has been tested with tens of video sequences consisting of at least 300 frames. The sequences are a selected sample from a large database

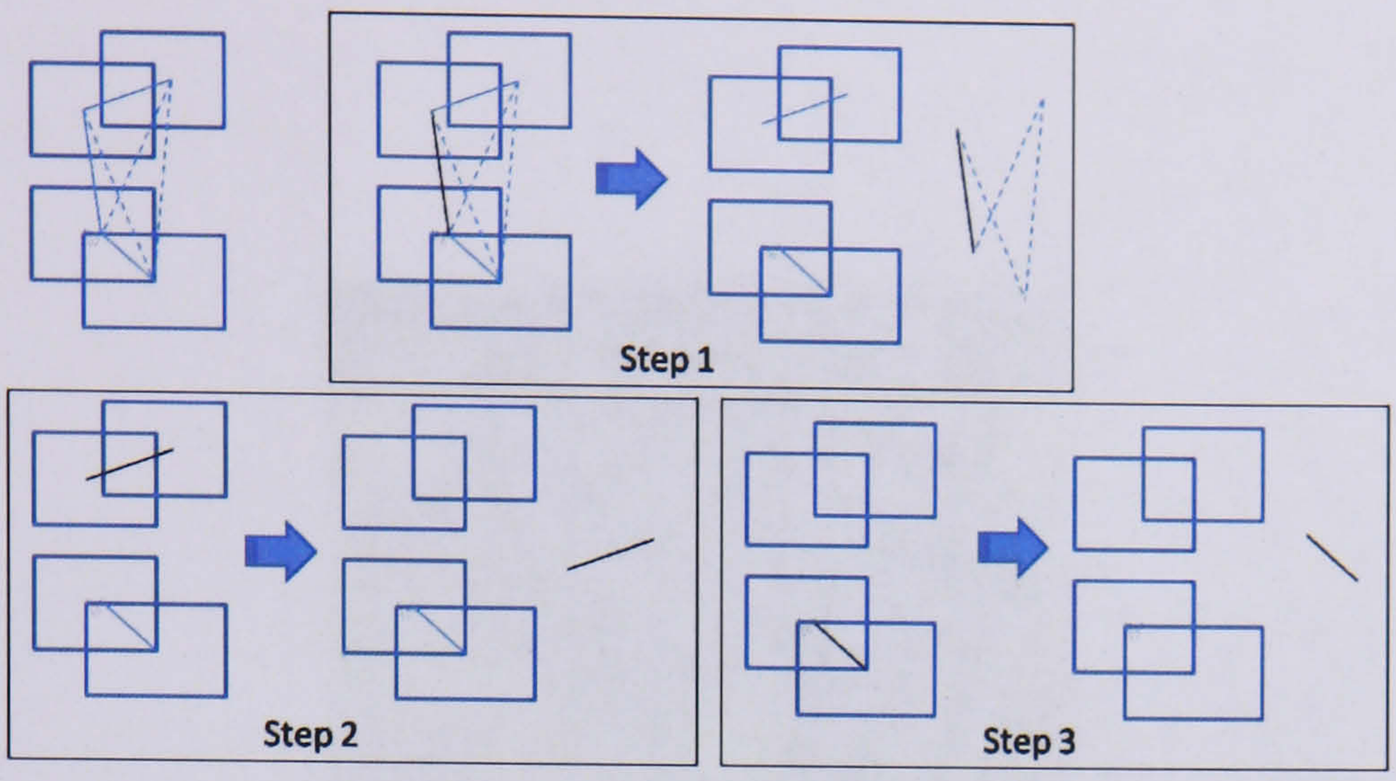


Figure 3.17: An example of sub-clustering. Solid lines between blobs show the *Connected* and the dashed lines are the *Unconnected*. In each step, the black *Connected* is removed, and the related *Unconnected* are removed. This operation is updated until all the *Connected* are moved and all the blobs are isolated.

of video data acquired at Kingston Hill during a number of simulation sessions. The selected excerpts of video sequences were ground-truthed.

For a video sequence, the number of people as well as their locations is retrieved for each frame. To access the system performance, ground truth is manually marked up by the ViPER Ground Truth Authoring Tool(ViPER-GT tool), which is a part of The Video Performance Evaluation Resource (ViPER) developed by the Language and Media Processing Laboratory, University of Maryland [1]. The *ground truthing* process is carried out every frame, and each person is selected by a bounding box (Figure 3.18). In this counting work, performance was evaluated using measures borrowed from the information retrieval literature. Recall and Precision, which have been used in evaluating search strategies, are used here to test the results of the counting algorithm against ground truth information. Recall is the ratio between the number of relevant records retrieved and the total number of relevant records in the database. Precision is the ratio between the number of relevant records retrieved and the total number of irrelevant and relevant records retrieved. The Precision-Recall curve is employed to provide a quantitative assessment of the performance of the algorithm (39). The

[1]The details of ViPER and ViPER Ground Truth Authoring Tool are available online at http://viper-toolkit.sourceforge.net/.

Figure 3.18: A ground truth example from ViPER-GT.

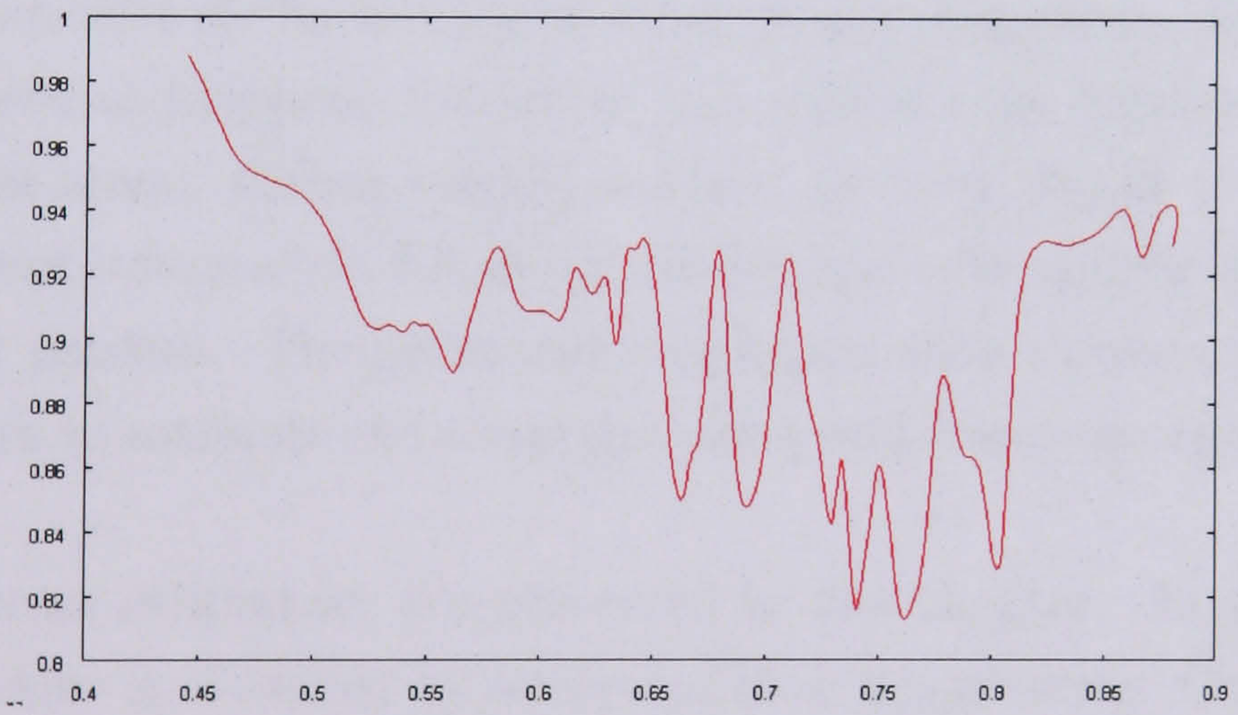


Figure 3.19: Precision-Recall curve.

bounding boxes of the ground truth GT and the bounding boxes generated by the presented algorithm RE are used to estimate the following measures:

$$Recall = \frac{GT \bigcap RE}{GT} \tag{3.19}$$

$$Precision = \frac{RE \bigcap GT}{RE} \tag{3.20}$$

Category information is also considered, i.e. the intersections of GTs and REs with different colours are not taken into account. The Recall and Precision estimates have been recorded along with time scale in all the video sequences, and each pair of measures contributes as a point on the Precision-Recall curve. Figure 3.19 shows the Precision-Recall curve for a video sequence. The graphs in Figure 3.20 illustrate the counting results from different situations with different numbers of people and different numbers of professions. These results show that the system has a quite stable performance under the tested different circumstances.

3.6 Summary

The contribution of this chapter is in the design of an algorithm for the interpretation of a group scene. This chapter has described methods for a system that follows the guidelines of the Intelligent Environment paradigm. At present, only cameras are used to recognise behaviour and estimate the category and number of people in the scene. Colour models are used to track people in the scene and provide sufficient information for the system to generate graphs of detected and tracked colour patches. The generated graphs are then automatically analysed by an algorithm to estimate the scene dynamics and count the number of people in the scene.

Two dynamics estimators are presented in the chapter. In the first work, people's behaviour is classified by analysing their trajectories. Curvature of the trajectory is accumulated in a time window to generate curvature density. Experimental results show that the curvature density can be used as a signature of people's behaviour. In the second work, entropy is calculated upon colour PDFs; the value of the entropy in a specific time frame indicates the level of cluttering

Figure 3.20: Counting people: ellipses represent the original blobs. Thick outlines of shapes show the existences of individuals. If a single colour blob is counted as an individual, the blob is displayed as an ellipse with thick outline; otherwise it is displayed with thin outline. For example, in the first graph of second row, the big yellow rectangle on the left with thick outlines is to show the existence of a nurse while the ellipses with thin outlines inside it are to show the original detected colour blobs.

at that time.

For counting the algorithms, the major challenge is to identify individuals from the colour segments in a complex dynamics environment. An algorithm that recognizes individuals as peaks of a colour PDF histogram has been proposed. The implementation of this algorithm is simple and it can work fairly well with not too complicated scenes, i.e. the scenes without occlusions or dramatic light changing. For more general situations, another algorithm is developed by analysing the spatial relationships between blobs. The basic assumption is that two individuals will not always remain in close proximity over a long period of time. Spatial relationships between blobs are judged to determine if two blobs should be merged or not. The current system provides a good estimation of the number of people. High precision values in quantitative results suggest that the system has a low false alarm rate. This is also confirmed by observing the qualitative results. However, as the analysis is limited to 2D information, the system would fail to count a person when seriously occluded and most of their patch is not visible from the view. As a result, Recall sometime drops to relatively low values. To tackle the problem of mis-counting, the next step for performance improvement is either to introduce an occlusion handling scheme or to fuse information from different views.

In terms of algorithm development, future work will focus on the description of the level of cluttering of the scene, and dynamics descriptions of the scene such as descriptions of people interactions. Furthermore, evidence from all the cameras can be combined to provide 3D information. In terms of technology, radio frequency technology will be introduced to help with the recognition of positional information of scene actors.

Chapter 4

Measuring Crowd Dynamics

In this chapter, computer vision techniques are used to automatically observe and measure crowd dynamics. The problem is studied in order to offer methods to measure the complex movements of a crowd. The refined matching of local descriptors was used to measure crowd motion. This chapter presents two novel algorithms to measure crowd dynamics; furthermore, a performance comparison of these two algorithms is also provided.

4.1 Introduction

The objective of this part of the work is to devise methods to automatically measure the crowd phenomenon. Crowded public places are increasingly monitored by security and safety operators. There are companies (for example LEGION (87)) that employed large resources to study the phenomenon and generate realistic simulations: for instance to optimize the flow of people of a public space.

Computer Vision research offers a large number of techniques to extract and combine information of a video sequence acquired to observe a complex scene. The life cycle of a computer vision system includes the acquisition of the monitored scene with one or more homogeneous or heterogeneous cameras, the extraction of features of interest and then the classification of objects, people and their dynamics. In simple scenes the background is extracted with statistical methods and then foreground data and related information are inferred to describe

and model the scene. The background is usually defined as stationary data, for instance a man-made structure such as buildings, in a typical video surveillance application, or the indoor structure of a building in a safety application, for instance deployed to monitor and safeguard elderly people in a home.

Unfortunately, background modelling becomes rapidly less effective in complex scenes and its usefulness seems to be inversely proportional to the clutter measured in the scene. Figure 4.1 shows a small experiment testing the effectiveness of background modelling with different types of scenes. The adaptive mixture of the Gaussian background models proposed in (126) is employed in the experiment. Three frames per chosen sequence and the resulting background image built with roughly 1000 frames are illustrated. The background modelling works well with the first scene but fails to recover the background of some regions in the second scene because of the frequent occupancy over these regions. In the third scene, due to the continuous clutter, the background model can be barely recovered. The failure of background modelling in extremely crowded situations is foreseeable, as the core of current background estimation is that the frequency of background is significantly higher than that of the foreground. When the monitored scene becomes very cluttered, then one could think of measuring dynamics with optical flow methods, designed to extract information about the dynamics of the scene and typically using gradient information. Unfortunately, popular and conventional optical flow techniques such as Horn and Schunck (59)and Lucas and Kanade (91) also work poorly with heavily crowded scenes. On the other hand, feature-based optical flow techniques using multi-resolution work quite well with relatively high frame rate (typically around 25fps) video sequences (23). Algorithms exist to analyse simple scenes, where a few people enter and exit the field of view of the deployed cameras. In such scenes, people and objects are identified and tracked throughout the network of cameras. People and objects such as vehicles are tracked between frames [1]and their trajectories are also predicted using conventional Kalman filters, or more sophisticated particle filter techniques. The problem with tracking in very cluttered and complex scenes is that matching is not always possible and tracks are frequently lost, creating fragmentation in the tracking process. In highly crowded scenes, tracking is not a viable option and

[1]Tracking refers to matching and predicting the position of identical objects over time.

Figure 4.1: The example frames and the built background images from three different scenes. Left to right: three different scenes; top to bottom, three example frames and the built background images, respectively.

it is more interesting and valuable to retrieve the global crowd motion instead of individual motion. The proposed method is to track for short periods of time, and two algorithms are proposed to provide matching between the frames for use in short-time tracking. The extracted and matched dynamics features can then be directly used in the process of crowd understanding and dynamics modelling.

This chapter presents two methods that can automatically measure crowd dynamics. The methods are feature-based and employ constraints to refine the matching. Both methods have been assessed with video sequences capturing different types of crowded situations. A comparison of the two methods is carried out and also described in this chapter. The performances of both methods pro-

duce satisfactory results, even with low frame rate video sequences (typically 4 to 8 fps).

This chapter is organised as follows: Section 4.2 describes the algorithm that uses Harris corner points and topological constraints and Section 4.3 describes the algorithm that uses maximum curvature as local descriptors and edgelet constraints. Section 4.4 illustrates the quantitative comparison of the two algorithms and Section 4.5 gives some comments on both of the algorithms.

4.2 Method *I*: Pyramid-based Interest Points Topological Matching

In order to devise algorithms to automatically derive complex crowd dynamics, local descriptors, classified as interest points, have been extracted using colour gradient information at the scale space. Furthermore, besides the use of the extracted descriptors, an advanced matching improved by incorporating topological constraints has been developed.

4.2.1 Extraction of the local descriptor: Harris detector

The first method employs a modified version of the Harris interest point detector (49)). The Harris interest point detector provides a repeatable and distinctive descriptor of the image features and is view-point and illumination invariant. This detector extracts feature points by making use of the three chromatic channels defined as the M matrix:

$$M = G(\sigma) \otimes \begin{pmatrix} C_x \cdot C_x & C_x \cdot C_y \\ C_y \cdot C_x & C_y \cdot C_y \end{pmatrix} \tag{4.1}$$

In the operation, the image is firstly smoothed using a standard Gaussian operator (of deviation σ). C_x and C_y are the gradients in x and y directions of the pixel chromatic triplet, respectively. They are estimated by applying the Gaussian derivative operator $G(\sigma)$ of (deviation σ) to the smoothed image, which is efficiently implemented by using the method from (138). The interest points are

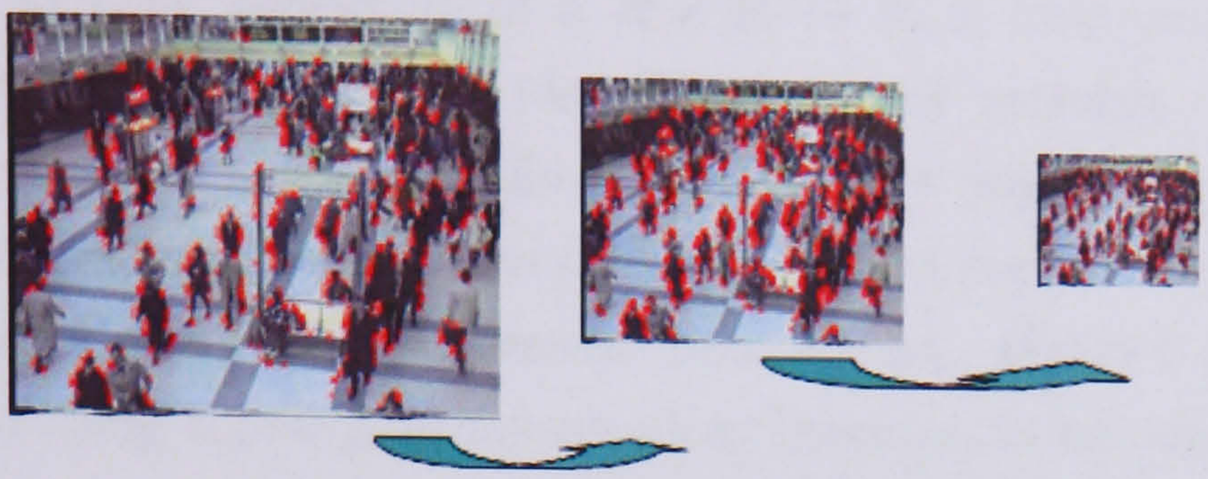


Figure 4.2: Interest Point Generation, from bottom layer to top layer

then extracted using term R, which is calculated as a combination of the Eigen values of the M matrix:

$$R = det(M) + \kappa trace^2(M) \tag{4.2}$$

where κ is a constant where $00.4 \leq \kappa \leq 0.06$. The points with a local maximum are selected as interest points. A multi-scale approach is used, generating the interest points at the lowest (finest scale) layer and then projecting them up to the top (coarsest scale) layer of the generated pyramid (Algorithm 3).

Algorithm 3 Creation of Interest Points

for for N images in (for temporal smoothing in Section 4.2.3) **do**
  Generate pyramid image gradient
  Detect interest points at bottom layer
  Project interest point to top layer
end for

4.2.2 Point Matching

The matching is carried out in two steps: searching for the candidate matching points by similarity and then applying the topological constraints described later. The similarity is given by the formula (given a, b as two interest points):

$$sim(a, b) = \frac{min(R_a, R_b)}{max(R_a, R_b)} \tag{4.3}$$

as introduced in (72), consisting of a $R \times R \rightarrow [0, 1]$ mapping. Frequent occlusions reduce the probability of identifying correct matches and as a result, without local support, similar gradient local regions might be found as plausible matches generating false positives. A topological constraint is implemented to make the search for correspondences more robust. Gabriel (47) proposed a similar method using topological information; however, in his algorithm the area (object) of interest was pre-defined and the topological information was evaluated by the already known centre of the object. Gabrie's work is for the purpose of tracking individuals, which is focused on the motion of a particular object over a long period. In the approach described here, the motion of all objects between two consecutive frames is more desirable. Therefore, the necessary local support is derived from local windows centred at the interest points and the relative locations of the interest points in such windows are used. Support is estimated for the matched interest point pair P_{t0} (the point at time $t0$) and P_t(the point at time t). Support vectors are calculated as:

$$V_{t0}^i = P_{t0}^i - P_{t0},$$
$$V_t^i = P_t^i - P_t. \tag{4.4}$$

where P_{t0}^i and P_t^i are the interest points inside the support window at times $t0$ and t, respectively. The matched support is then quantified in terms of the error by measuring the standard deviation of the ensemble of found correspondence.

$$\epsilon = f(\sigma_\theta, \sigma_\rho) \tag{4.5}$$

where

$$\theta^i = N(V_{t0}^i \cdot V_t^i),$$
$$\rho^i = \left\| V_{t0}^i \right\| - \left\| V_t^i \right\|. \tag{4.6}$$

are, respectively, the orientation difference (doc product) and the length difference between the support vectors.

4.2.3 Temporal pyramidal analysis

Temporal smoothing and matching is also carried out by comparing a number of N spatial pyramids, corresponding to a specific time window. Thus, a spatial-temporal pyramidal analysis of the sequence is generated for a number of frames. Temporal smoothing is employed to enforce time consistency on matches, reducing the false alarms generated by unstable interest points.

So matching is carried out in both space and time, starting at the highest level (coarsest level) of each pyramid, searching interest point correspondences between the initial frame of the N frames and each other's frames within the given time period (corresponding to $N-1$ matches). Spatial matching works from the top (finest scale) of a pyramid to the bottom (coarsest level). Then temporal integration of pyramidal matches of the interest point j in 0^{th} frame can then be applied by combining the N matches.

4.2.4 Evaluation

A series of experiments were run on different video sequences. The assessment of results is not a trivial task, given that it is virtually impossible to generate ground truth data. However, a quantitative evaluation of results can also be provided. For a window of interest in an image of a given sequence, all interest points are retrieved and then their displacements are estimated against the image at the next frame. All displacements are then combined into a resulting vector that indicates the position of the window of interest in the next frame. Comparing structures cannot work, because background structures would generate a large amount of noise. Therefore, Receiver Operating Characteristic (ROC) curves (133) are generated for performance evaluation. A series of points is estimated to produce the ROC curves, using the following two formulae:

$$P_{fp} = \frac{FP}{TN + FP},$$
$$P_{tp} = \frac{TP}{TP + FN}. \tag{4.7}$$

Table 4.1: The definitions of parameters for the ROC curve

ROC		Predicted	
		Positive	Negative
Signal	True	TP	TN
	False	FP	FN

where the definitions of the parameters are shown in Table 4.1. Hence, P_{tp} represents the fraction of positives correctly predicted and P_{fp} represents the fraction of negatives incorrectly predicted.

Figure 4.3 contains ROC curves for two image sequences. The ROC curves are generated by comparing the predicted positions of all interest points against the actual interest points found. If an interest point has been found in the location where it is predicted, it counts as a true positive (TP), otherwise it counts as a false positive (FP).

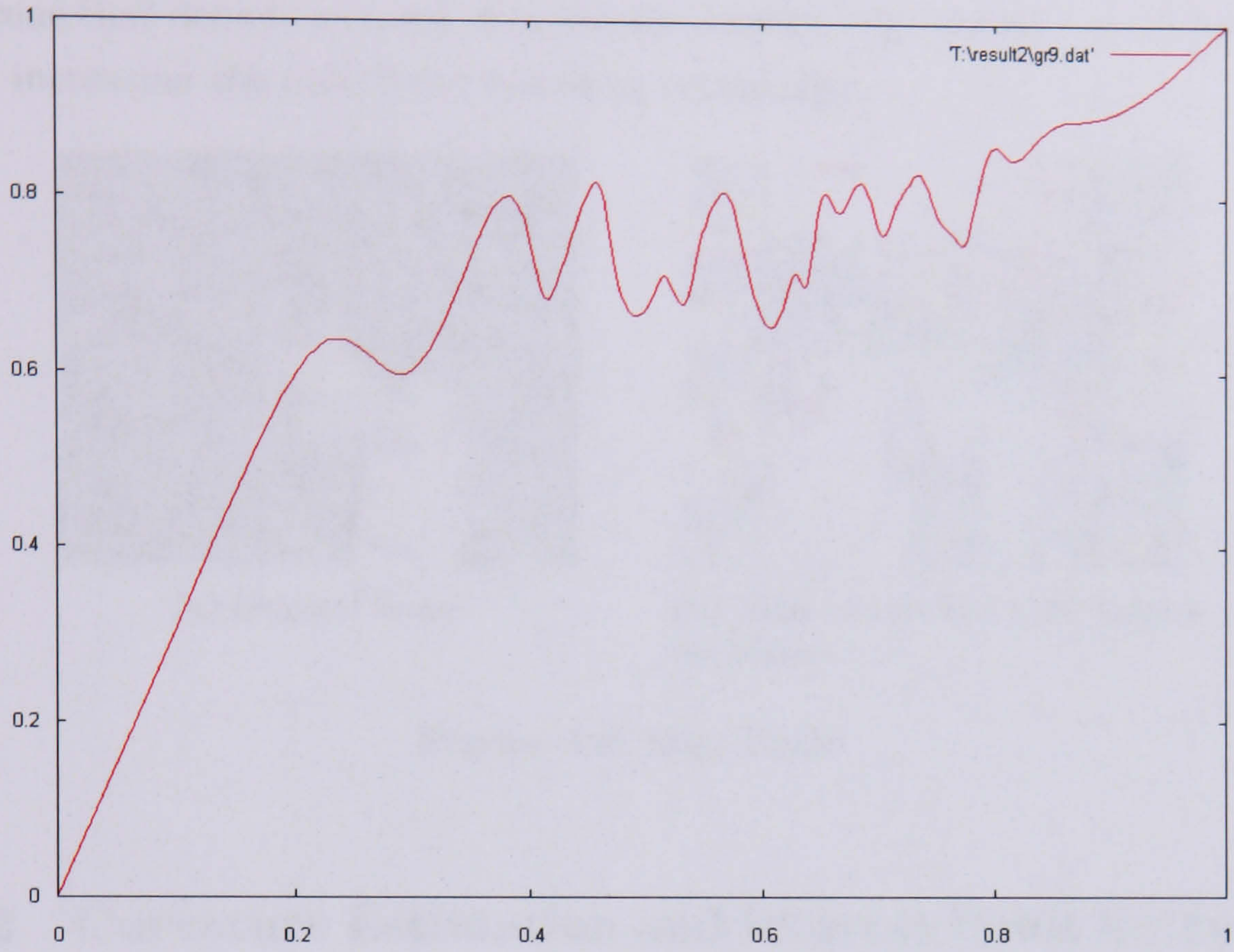
4.3 Method *II*: using Edge Continuity Constrains of Interest Points

The second method is developed using local descriptors, as well as incorporating shape information. Inspired by the methodology used in deformable object tracking, edge information is extracted and descriptor points are extracted as points along an edge with local maximum curvature. The information about an edge is maintained and used to impose the *edgelet constraint* and refine the estimate. Here, *edgelet* refers to equal length segments of the edge in the image. Thus, the advantages of using point features that are flexible to track and using edge features that maintain structural information are combined here.

4.3.1 Edge Retrieval

The Canny edge detector is employed to extract the edge information of a given frame. Each Canny edge is a chain of points, and all the edges are stored in an edge list. Figure 4.4 shows an example image frame and the extracted edge chains with associated bounding boxes, respectively. It can be observed that even

4.3 Method *II*: using Edge Continuity Constrains of Interest Points



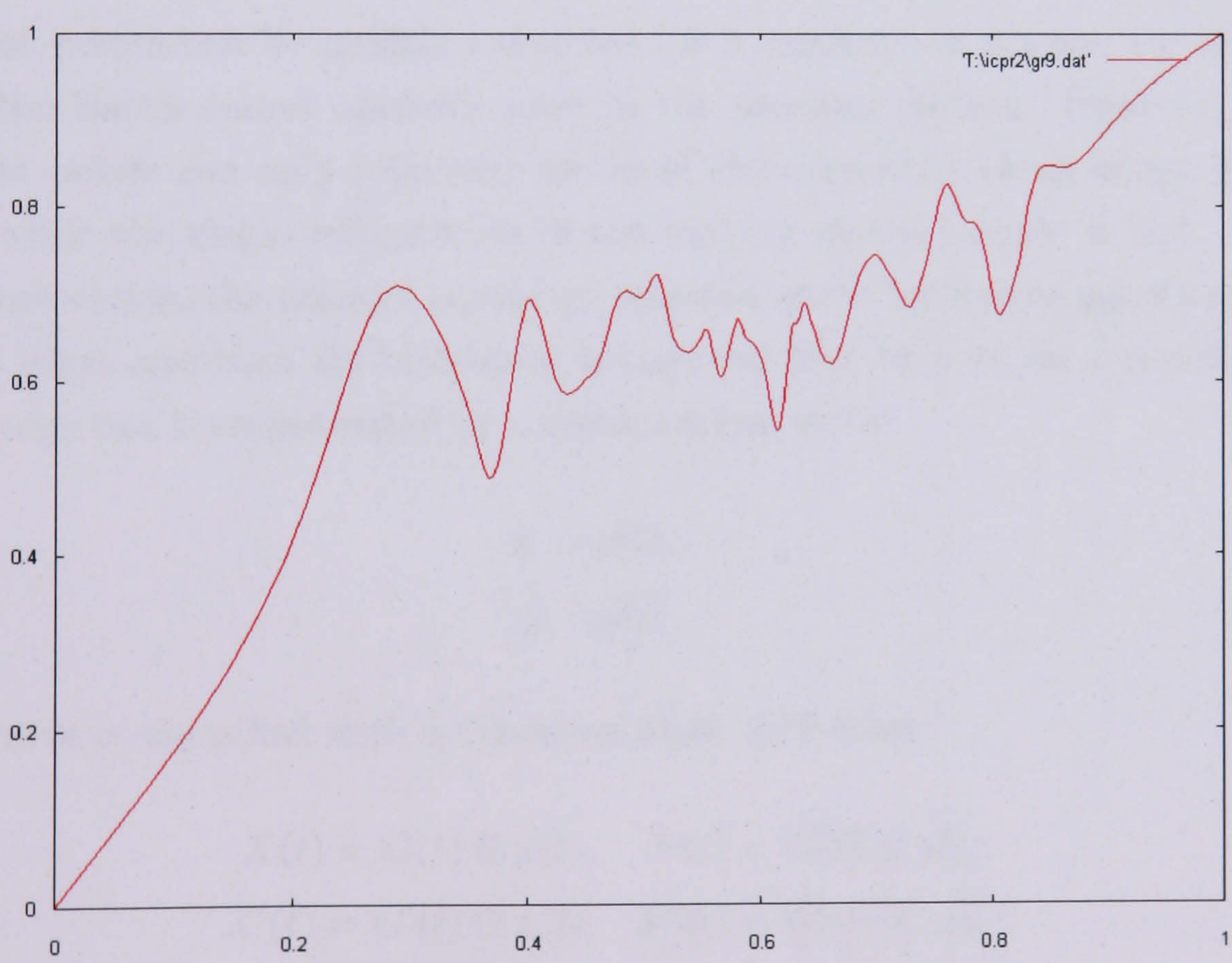
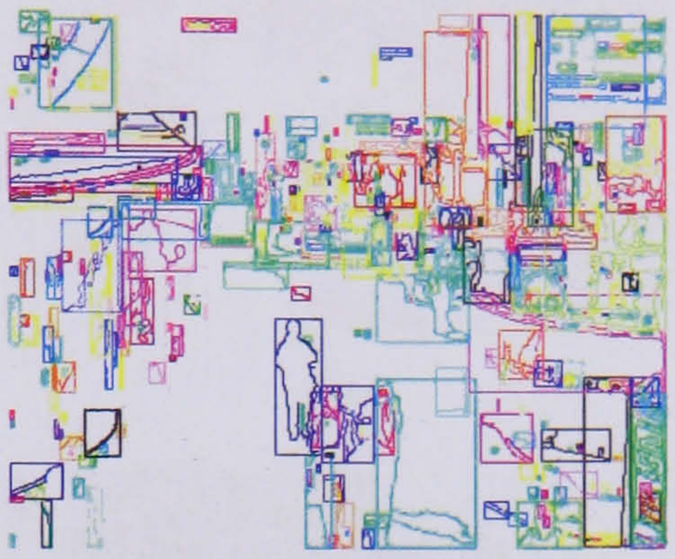


Figure 4.3: ROC curves of two image sequences (with the vertical axis as P_{tp}, the horizontal axis as P_{fp})

in a scene that depicts a crowd of moderate density, edge chains can occlude each other, increasing the descriptor matching complexity.

(a) Original frame.



(b) Edge chains and their bounding boxes).

Figure 4.4: Edge Chain

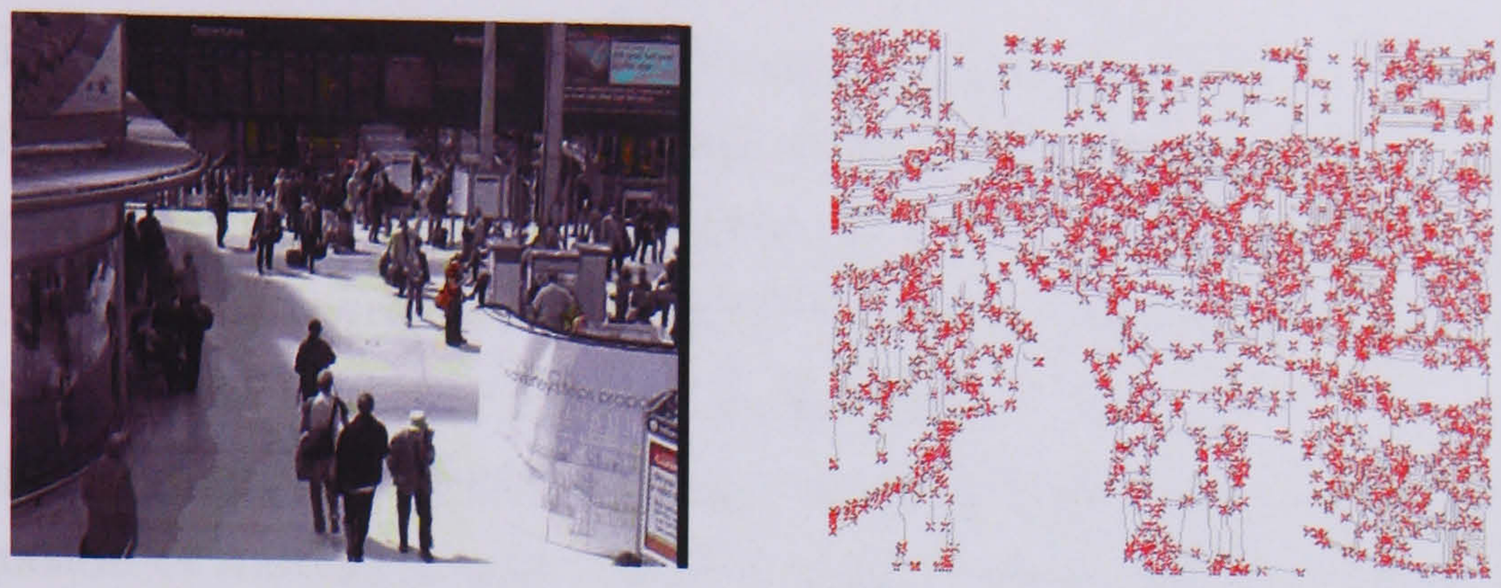
4.3.2 Curvature Estimation and Interest Point Extraction

Interest points can be quickly extracted for a sequence of frames, for instance, with the Harris corner operator used in the previous section. However, Harris interest points can only represent the local characteristics of an image in isolation, while the shape information of the moving person/people is lost. In this implementation, the interest points are selected as the local maxima of curvature of the edges and then the constraint is imposed that they lie on a specific edge. Each edge can be represented by a parametrised curve:

$$\begin{aligned} x &= x(t), \\ y &= y(t). \end{aligned} \tag{4.8}$$

The curve is smoothed with a Gaussian filter, as follows:

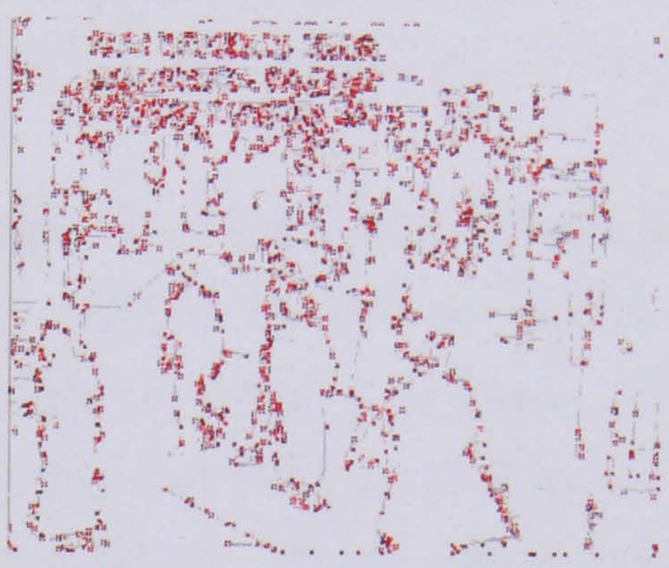
$$\begin{aligned} X(t) &= G(t) \otimes x(t), \quad Y(t) = G(t) \otimes y(t); \\ X'(t) &= G'(t) \otimes x(t), \quad Y'(t) = G'(t) \otimes y(t); \\ X''(t) &= G''(t) \otimes x(t), \quad Y''(t) = G''(t) \otimes y(t). \end{aligned} \tag{4.9}$$



(a)

(b)

(c)



(d)

Figure 4.5: Two scenes of different complexity levels are illustrated. The original frames (left) and the extracted corner points (right) that are marked with red crosses on grey edges.

The curvature of each edgelet at corner point C can then be given by (103):

$$\kappa(C) = \frac{X'Y'' - Y'X''}{(X'^2 + Y'^2)^{\frac{3}{2}}} \tag{4.10}$$

Corner points are defined and extracted as the local maxima of the absolute value of curvature on each edge. Thus, the edge representation is changed from a point sequence to a corner point sequence, resulting in a list of corner point sequences for all the edges of the image.

4.3.3 Point Matching and the Edgelet Constraint

Given two consecutive frames I_t and I_{t+1}, the motion is estimated for each extracted point of interest. For each corner point with the coordinate (x, y) in I_t, a

rectangular search window is defined centring at (x, y) in I_{t+1}. A look-up table (LUT) containing corner point and edge information is generated to enhance the matching. The correspondence is matched by using the curvature information of corner points in the search window in LUT against a reference point. The error is calculated by the curvature defined in Equation 4.10.

Complex dynamics and frequent occlusions generated in crowd scenes make the estimation of motion a very complex task. Point matching in isolation is too fragile and prone to errors to provide a good motion estimator. If the interest points are extracted on edge chains, then the edge constraint can be imposed and used.

For an image frame I_t, every edge is split to uniform length edgelets, represented by sub-sequences (so called edgelet). There are two reasons for this: to avoid a very long edge that could be generated by several different objects, and to enhance the matching of the edge fragments generated by occlusions. For each corner point there are n candidate matching points. Each candidate point belongs to an edgelet, thus there are $m(m <= n)$ candidate matching edgelets. To find the best match, three parameters are used: energy cost, variation of displacements and the match length for each candidate, and these are combined into a single matching score. The length of the edgelet is assumed to be small enough so that it will not split again to two or more matches. This is so that their candidate points correspond to the same candidate sequence. The parameters are defined below:

- **Energy cost** $\mathcal{E}$ This refers to the deformable object match that is calculated by accumulating the errors ϵ (again, calculated by the difference of the curvatures along the matching point pairs of the reference sequence and all the candidate match points that belong to the same candidate sequence).

$$\mathcal{E} = \sum \epsilon(i) = \sum |\kappa(C_{t_0}(i)) - \kappa(C_t(i'))| \tag{4.11}$$

  where $\epsilon(i)$ denotes the error at reference corner point $C_{t_0}(i)$ and $C_t(i')$ denotes the matching corner point on the candidate sequence.

- **Variation of displacements** $\mathcal{V}$ For each matching point pair there is a

displacement pair dx_i and dy_i. The combination of the variation between the two displacement vectors is as follows:

$$\mathcal{V} = \frac{1}{L_M} \times \sqrt{\sum \frac{1}{N}(dx_i - dx)^2 + \frac{1}{N}(dy_i - dy)^2} \tag{4.12}$$

where dx and dy are the average displacements between the matched point pairs, N is the size of the match window, and L_M is the number of total matched points of from the reference sequence to candidate sequence. Hence, $\mathcal{V}$ lies in the range between 0 and 1.

- **Match length parameter** $\mathcal{M}$:

$$\mathcal{M} = \frac{L_M}{L_E} \tag{4.13}$$

Where L_E is the total number of points on the reference sequence, M is between 0 to 1.

So, the overall matching score is given by:

$$\mathcal{S} = \mathcal{E} + \mathcal{V} + (1 - \mathcal{M}) \tag{4.14}$$

The candidate sequence of the minimum matching score will be selected. However, if the match length parameter $M < 0.5$, the result will be discarded. Figure 4.6 illustrates the process of applying edgelet constraints.

The matching is carried out over every point of the interested edgelet and an overall matching will be examined to determine the matched edgelet.

4.3.4 Evaluation

The video data used for this part of the evaluation is from the European project CAVIAR (110); the advantage of the CAVIAR dataset is that the ground truth information for this data is provided in XML format. To test the result, the foreground object position for each frame is estimated (by means of a bounding box) and translates every corner point in the bounding box to the consecutive frame by its estimated motion. All the translated points still in the box are

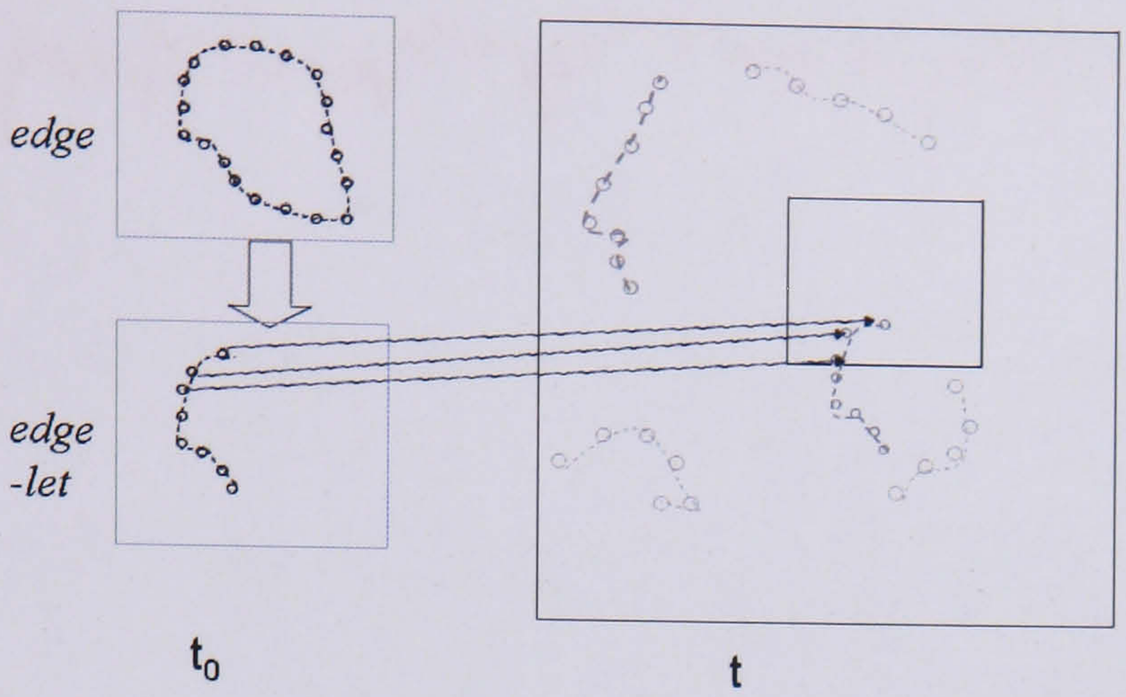


Figure 4.6: Applying edgelet constraints

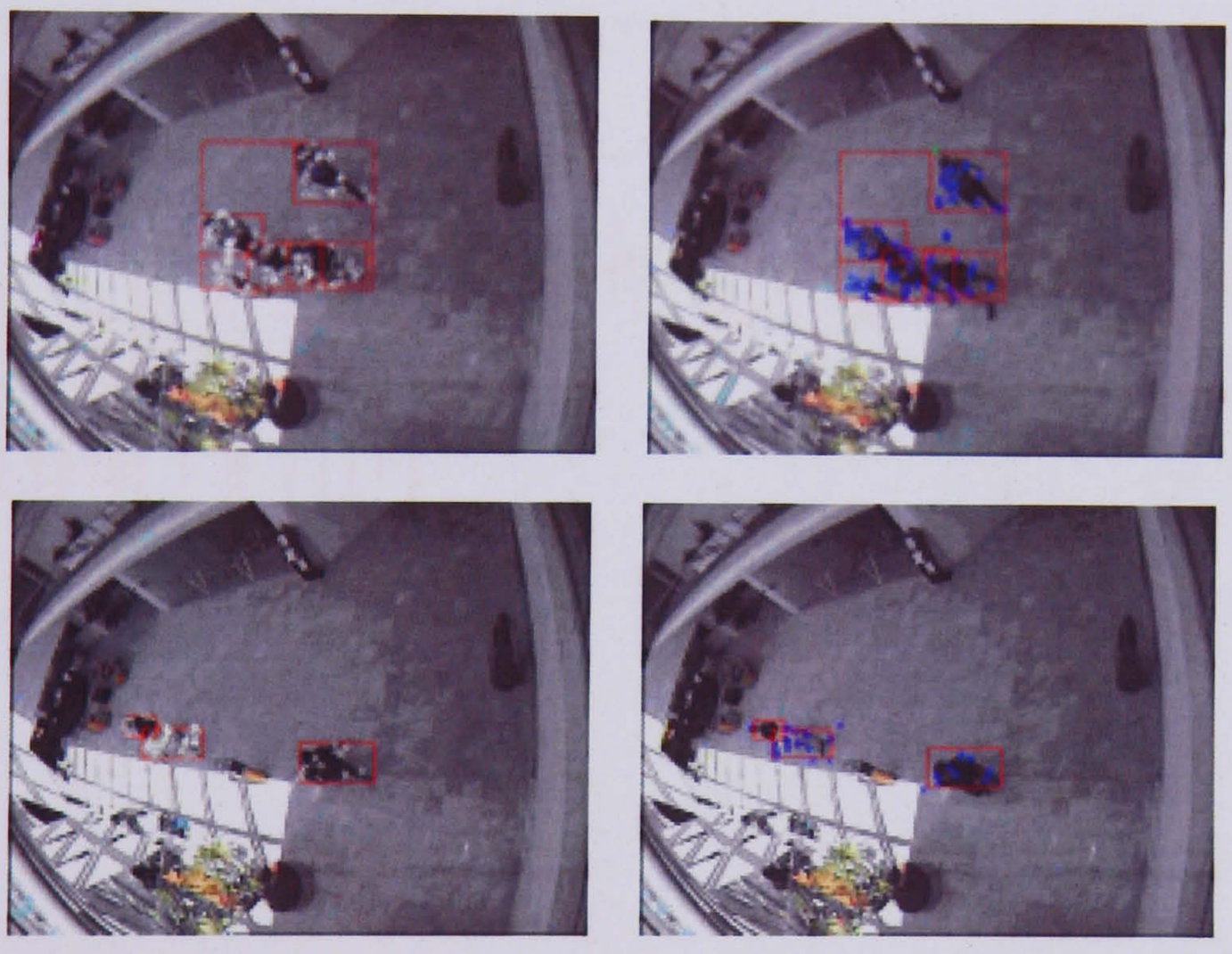
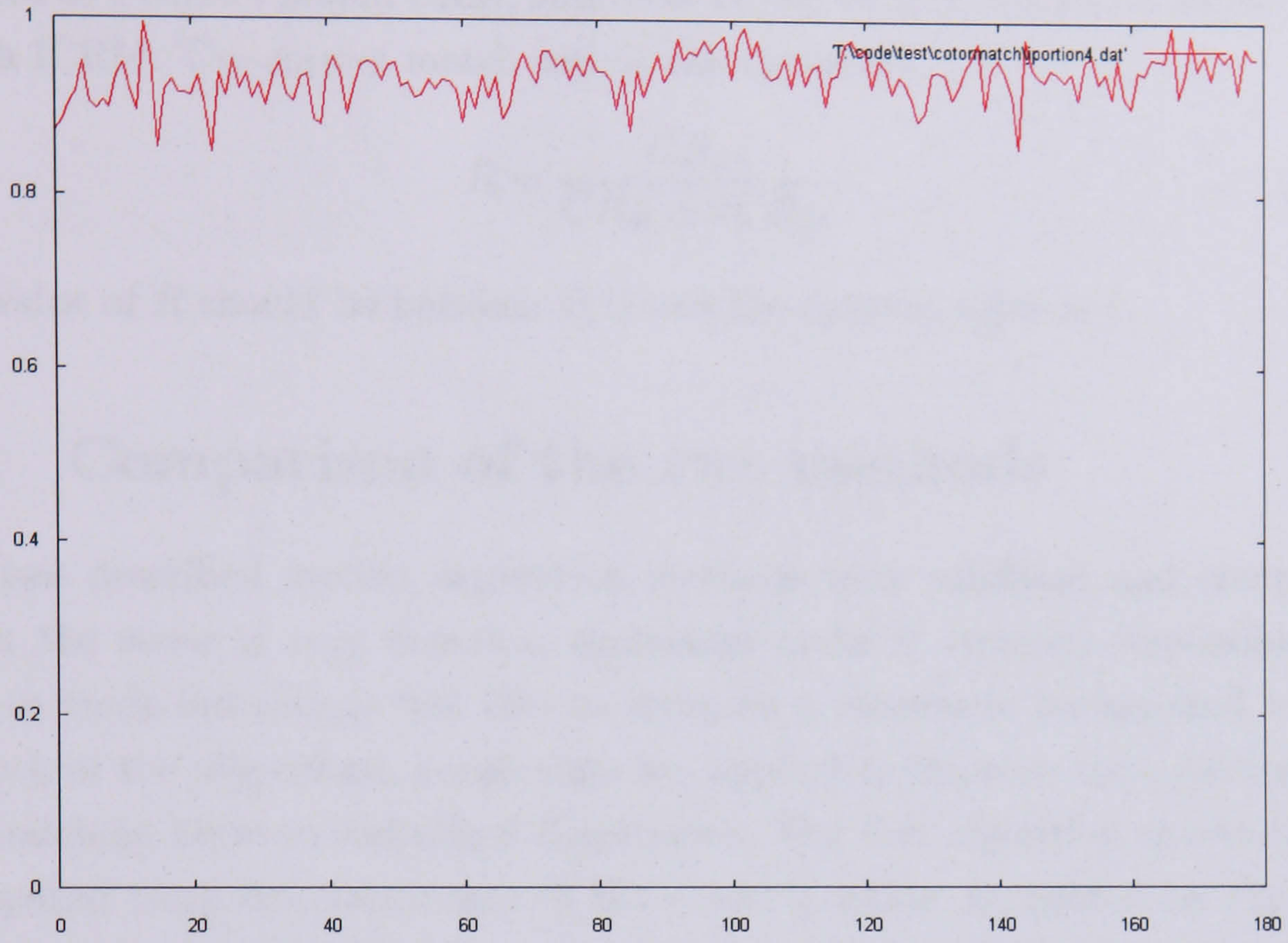


Figure 4.7: Two test data sets. The first column samples initial frames from both data sets, with the corner points indicated by white dots inside the ground truth box; the second column is the matched frame, with correct matched points CRM marked by a blue circle and incorrect matched points ICRM marked by a cross

4.3 Method *II*: using Edge Continuity Constrains of Interest Points



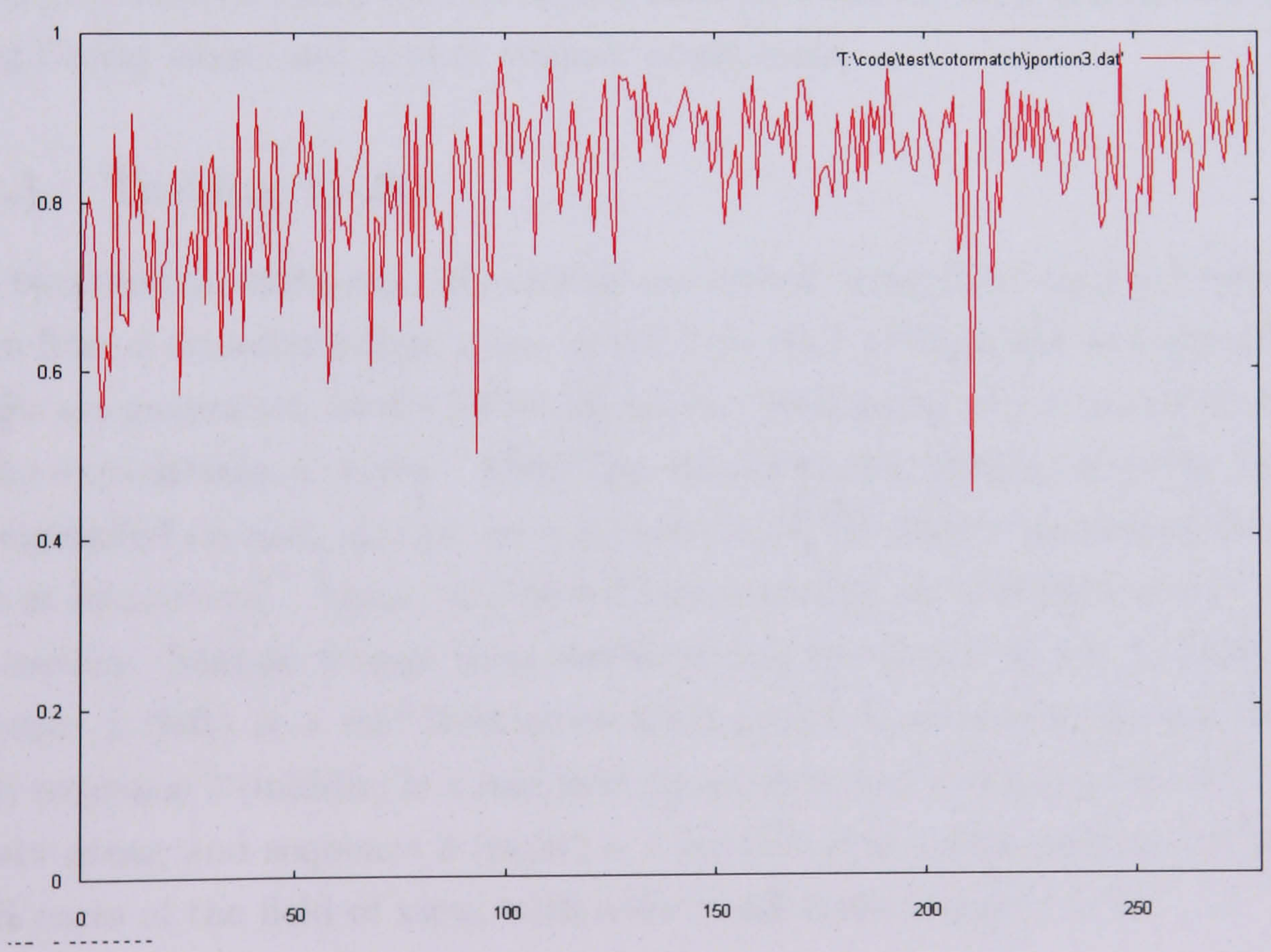


Figure 4.8: Correct match rate R along the frames of the sequences shown in 4.7.

counted as a correct match CRM, and those falling outside the box as an incorrect match ICRM. The correct match rate is calculated as:

$$R = \frac{CR_M}{CR_M + ICR_M} \tag{4.15}$$

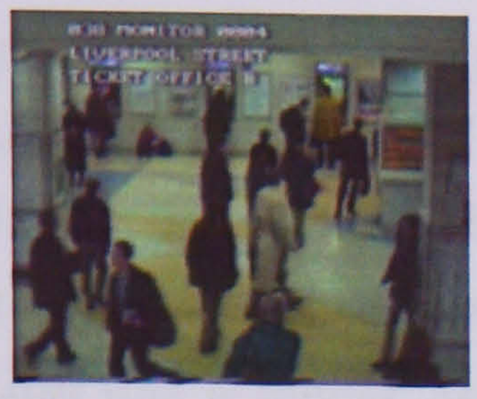
The value of R should be between [0,1] and the optimal value is 1.

4.4 Comparison of the two methods

The two described motion estimation methods were validated and compared. When the scene is very complex, occlusions make it virtually impossible not only to track individuals but also to estimate a stochastic background model. In both of the algorithms, constraints are applied to improve the robustness of the matching between individual descriptors. The first algorithm checks locally the spatial temporal consistency of the colour gradient supported by the local topology constraints and the second one uses the points of local extreme curvature along Canny edges and applies contour constraints.

4.4.1 Testing Data

The two motion estimation algorithms are tested using three types of sequences taken from a crowded public space on the London Underground and quantitative results are generated. In the following, a brief description of the test dataset used in the experiments is given. Then the details of the testing methods adopted are expanded on and, finally, an explanation of the results generated from the tests is introduced. Again, additional visual results are included at the end of the section. Sample frames from the three sequences are shown in Figure 4.9: sequence 1 (left) is a mid field scene with people scattered across the field of view; sequence 2 (middle) is a mid field scene with major motions taking place in certain areas; and sequence 3 (right) is a far field scene with pedestrians present in all parts of the field of view, with some predominant trajectories.



(a)

(b)

(c)

Figure 4.9: Sample frames from 3 testing sequences

4.4.2 Testing based on local descriptors

In this test, only the quality of the matching of individual local descriptors is considered. For each pair of consecutive frames, local descriptors in the initial frame are compared with their corresponding local descriptors, found by the two presented algorithms, in the target/other frame, respectively. Two measures - Mean Similarity (MS) and Mean Absolute Error (MAE) - are used to compare the performance of the two algorithms.

The images in Figures 4.10 and 4.11 represent the plots of MS and MAE for the two algorithms tested against the three sequences. MS and MAE are calculated for every frame along the sequence. MS is designed to assess the relative similarity of the matched local descriptors.

$$MS = \frac{1}{n}\sum_{i=0}^{n}\frac{min(X_i^{t_0}, X_i^t)}{max(X_i^{t_0}, X_i^t)} \tag{4.16}$$

where n is the total number of the local descriptors in the initial frame. MS is defined as the average of the similarity, and the similarity is calculated by the minimum of the two matched local descriptors' pixel value divided by the maximum. The result is a value that falls in the $(0, 1)$ range. Another measure, MAE, is commonly used for the testing of motion estimation algorithms (129) as it returns an error measure. MAE is defined as follows:

$$MAE = \frac{1}{n}\sum_{i=0}^{n}\|X_i^{t_0} - X_i^t\| \tag{4.17}$$

where $X_i^{t_0}$ is the pixel value at the i^{th} corner in the first frame, and X_i^t is the

corresponding local descriptor in the next frame. In each plot, the x axis represents time (the number of the frame) and the y axis represents the values of MS and MAE, respectively. Hence, for the two algorithms the MS and MAE for the three testing sequences are both good, although in most of the cases the second algorithm has a higher MS and a lower MAE. Meanwhile, along the time scale the MS and the MAE produced from the first algorithm fluctuate a lot while the second one produces more stable results. It can be concluded that the second algorithm has a more desirable performance than the first one.

4.4.3 Testing based on Motion Connect Component

The testing here makes use of the connected components algorithm based on motion vectors (the so called MCC - Motion Connected Component). The algorithm groups together motion vectors that are in close proximity and have common motion properties. The result of the MCC algorithm segments the motion field into clusters of a uniform motion group (e.g. a (part of a) pedestrian or a group of pedestrians), and the testing is based on each MCC to assess the two algorithms. In order to assess the two algorithms with MCC, Recall and Precision - used in Chapter 3 - are adapted again. In the proposed implementation, the bounding box of each MCC is taken and the average motion of MCC is calculated. Thus, the bounding box is mapped to the next frame. The number of "relevant records in the data base" should be the number of local descriptors of the MCC in the initial frame (N_{t_0}), while the number of "retrieved records" is the number of local descriptors in the mapped bounding box in the second frame (N_t). The definitions of the two measures are given by:

$$Recall = \frac{N_{t_0} \cap N_t}{N_{t_0}} \tag{4.18}$$

$$Precision = \frac{N_{t_0} \cap N_t}{N_t} \tag{4.19}$$

Both of the values fall in the range $[0, 1]$. For every frame, an average Recall value and an average Precision value are calculated. Figure 4.12 gives the plots of Recall and Figure 4.13 gives the plots of Precision; the layouts of these plots

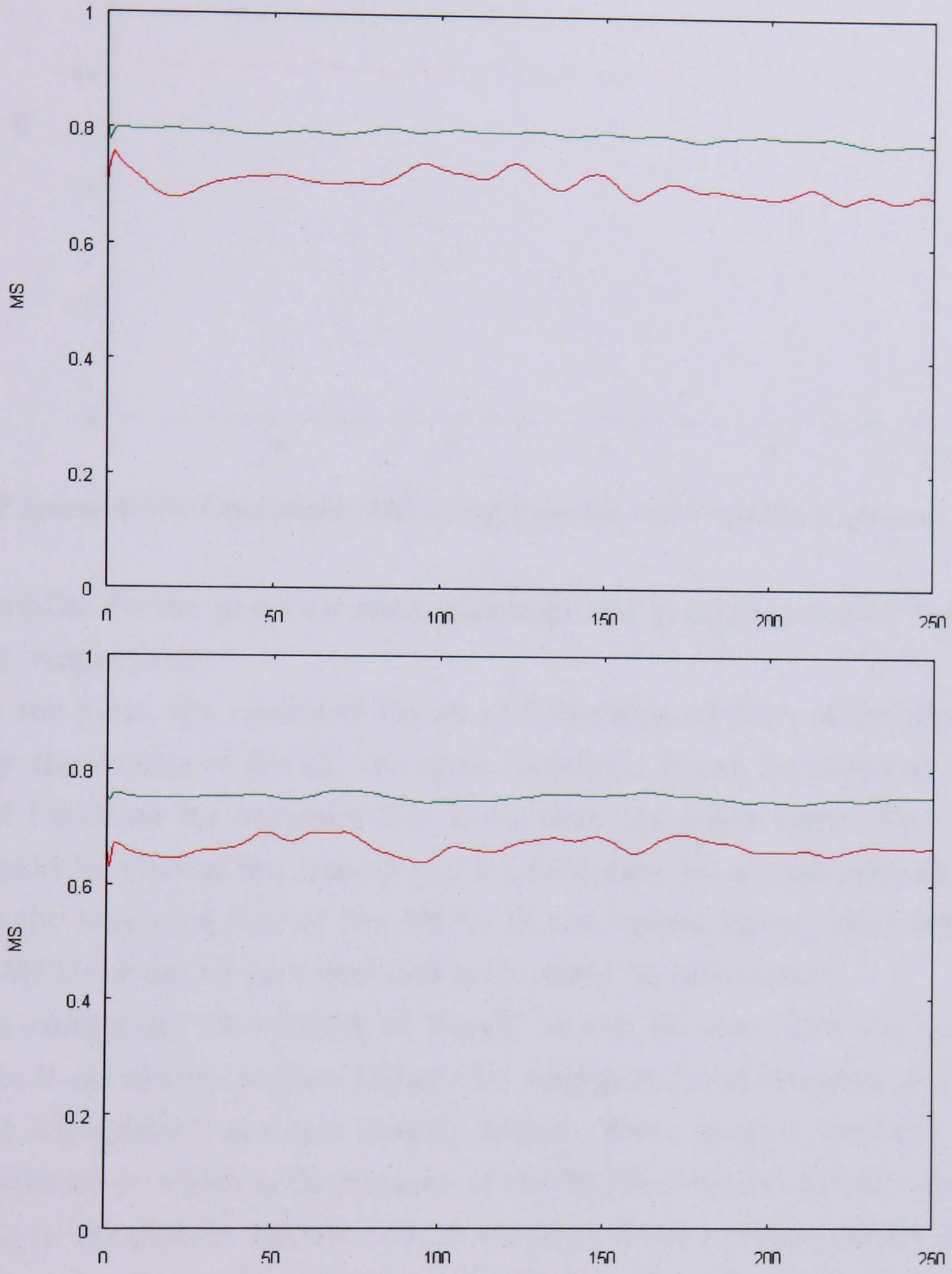


Figure 4.10: MS along time for the 3 testing sequences, red lines for Algorithm 1; green lines for Algorithm 2. Algorithm 2 keeps higher in MS.

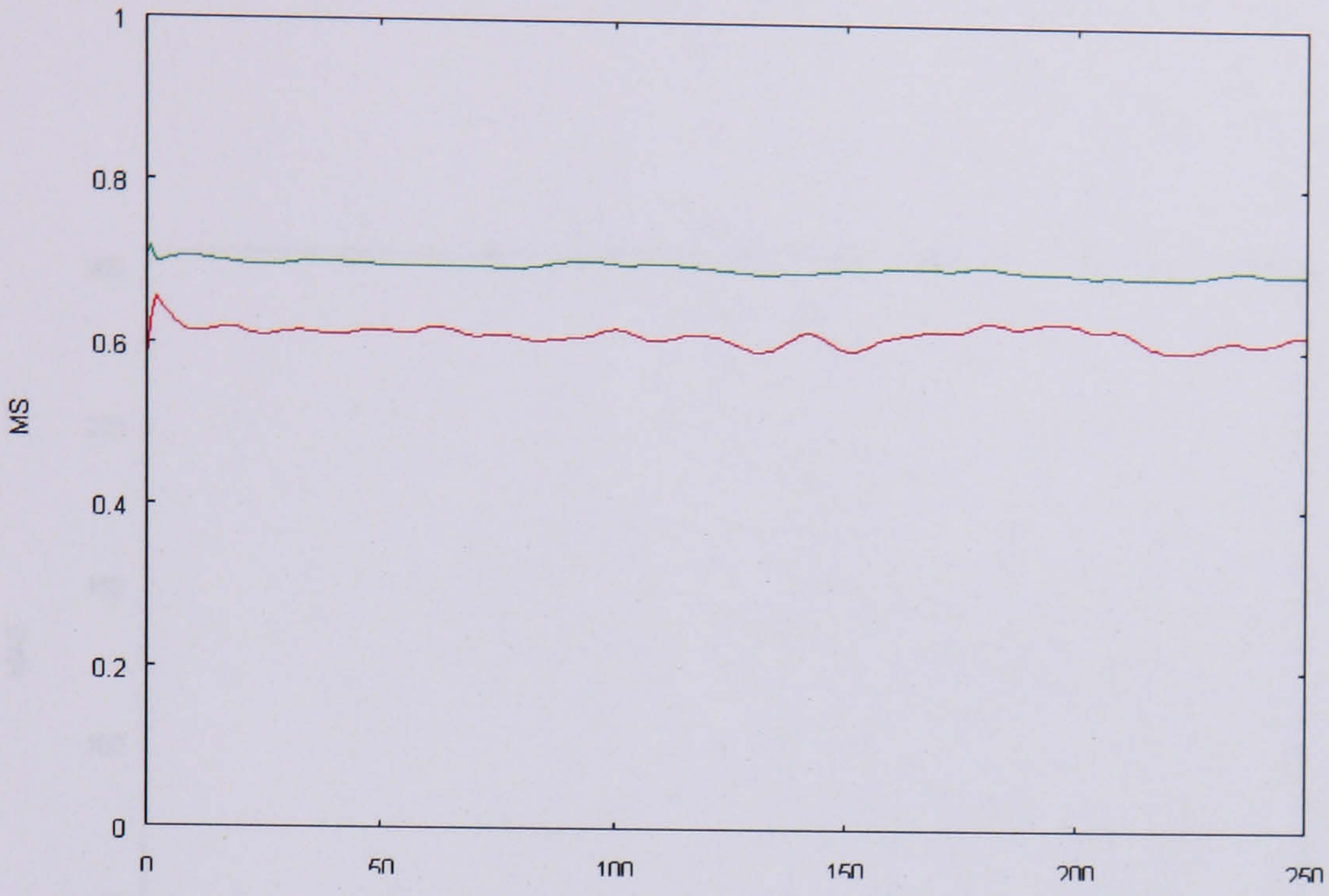


Figure 4.10: Continued: MS along time for the 3 testing sequences.

remain similar to the previous ones, although the y axis represents Recall and Precision, respectively.

From the plots, the results of Recall and Precision of both of the algorithms, especially the results of Recall, are again satisfied. It can be observed that the results of Precision for sequence 3 is lower than the other three. One possible reason could be that as sequence 3 is a far field view for a crowded scene, when mapping the bounding box of the MCCs to the second frame, local descriptors of other MCCs could be included and noise could be introduced.

When comparing the results of Recall, it can be seen that the values for Algorithm 2 are always higher, though for sequence 2 and sequence 3 Precision values for Algorithm 1 are only slightly higher. Here, another measure is taken into consideration, which is the number of the MCCs detected by each algorithm. According to the plots in Figure 4.14, in sequence 1 the average number of MCCs detected by Algorithm 1 is around 20, while by Algorithm 2 the number is around 100. In sequence 2, the numbers are around 20 and 200, respectively and in sequence 3 the numbers are around 40 and 280, respectively. Algorithm 2 detects much more MCC, especially for sequence 2 and 3. Due to the above fact and the fact that Algorithm 2 produces higher Recall, it can be deduced that the slight

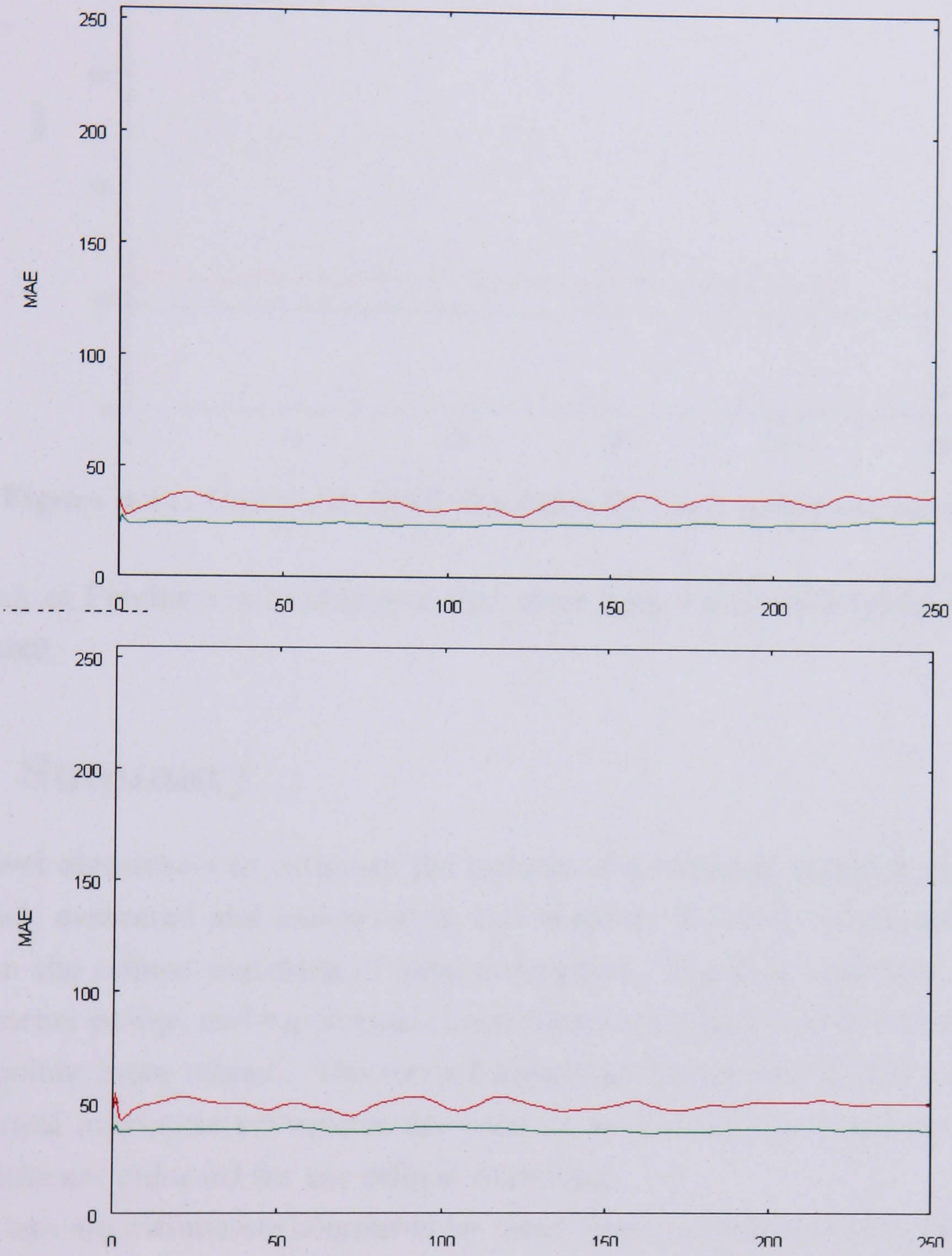


Figure 4.11: MAE along time for the 3 testing sequences, red lines for Algorithm 1; green lines for Algorithm 2. Algorithm 2 keeps lower in MAE.

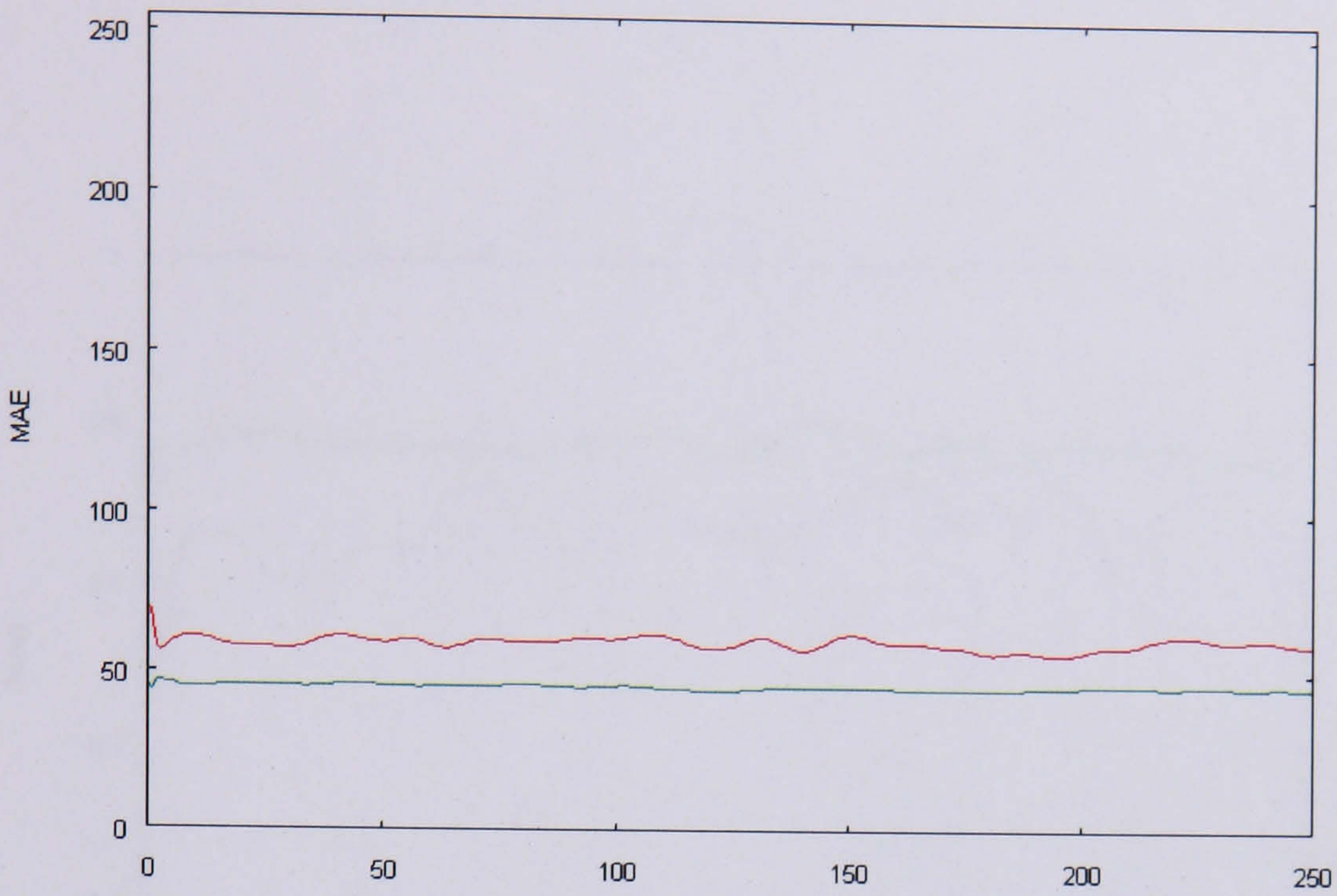


Figure 4.11: Continued: MAE along time for the 3 testing sequences

drawback of Precision only indicates that more noise has been introduced to the assessment.

4.5 Summary

Two novel algorithms to estimate the motion of a crowd in complex scenes are presented, evaluated and compared in this chapter. Both of the algorithms are based on the refined matching of local descriptors. The first algorithm employs Harris corner points, and topological constraints are applied to make the matching of the points more robust. The second algorithm makes use of shape information. Local maximum curvatures are used as local descriptors and the edgelet constraints are enforced for the refined matching.

The two algorithms are compared by using three surveillance video sequences and quantitative results are generated based on an individual local descriptor and MCC (Motion Connected Component). MS and MAE are used as criteria for the local descriptor-based assessment. The values of MS generated by the two algorithms are all above 0.6 and for Algorithm 2, the values are all above 0.7. For the values of MAE, those generated by Algorithm 2 are always below those

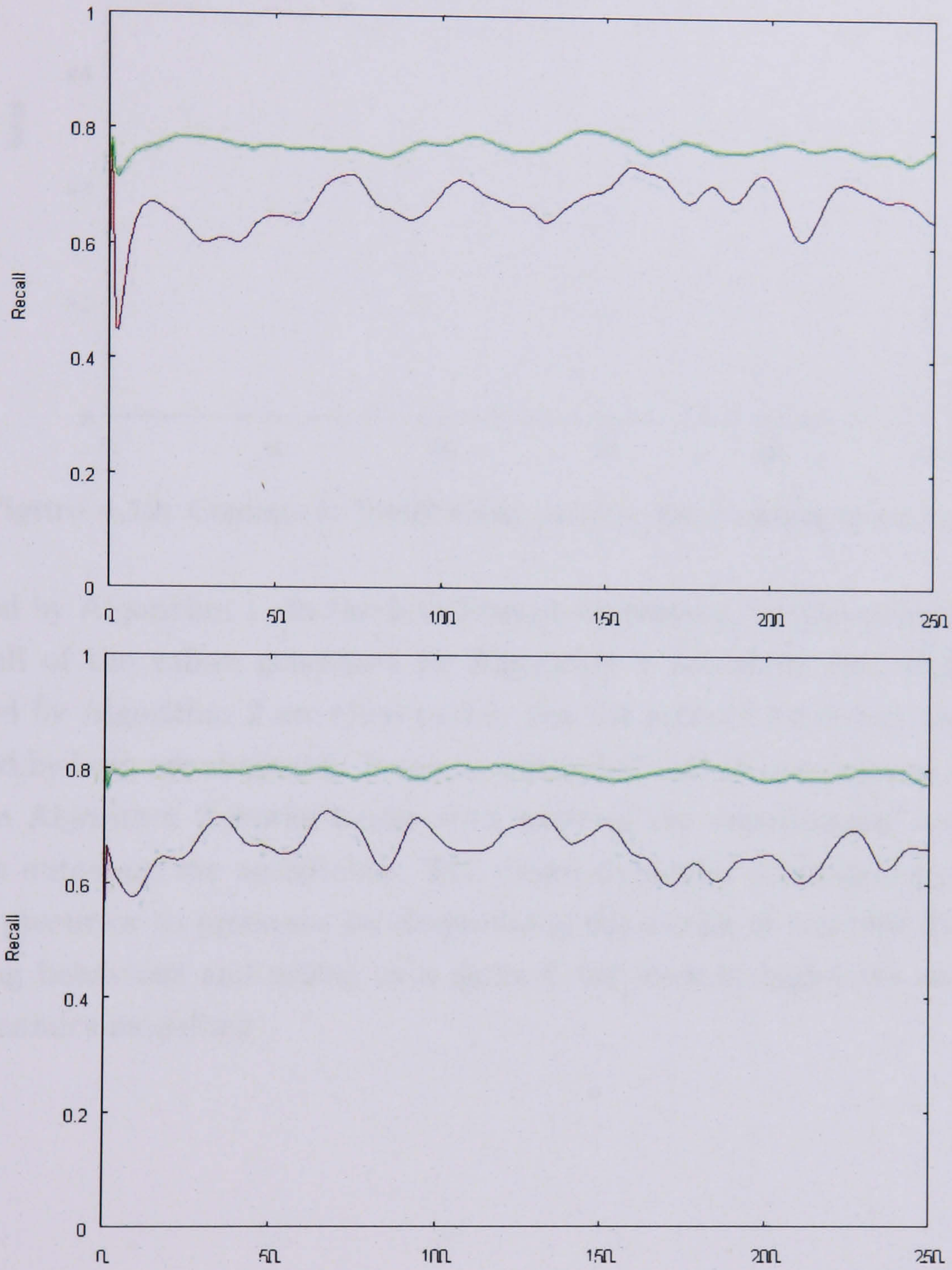


Figure 4.12: Recall along time for the 3 testing sequences. Algorithm 2 has higher values of Recall, red lines for Algorithm 1; green lines for Algorithm 2.

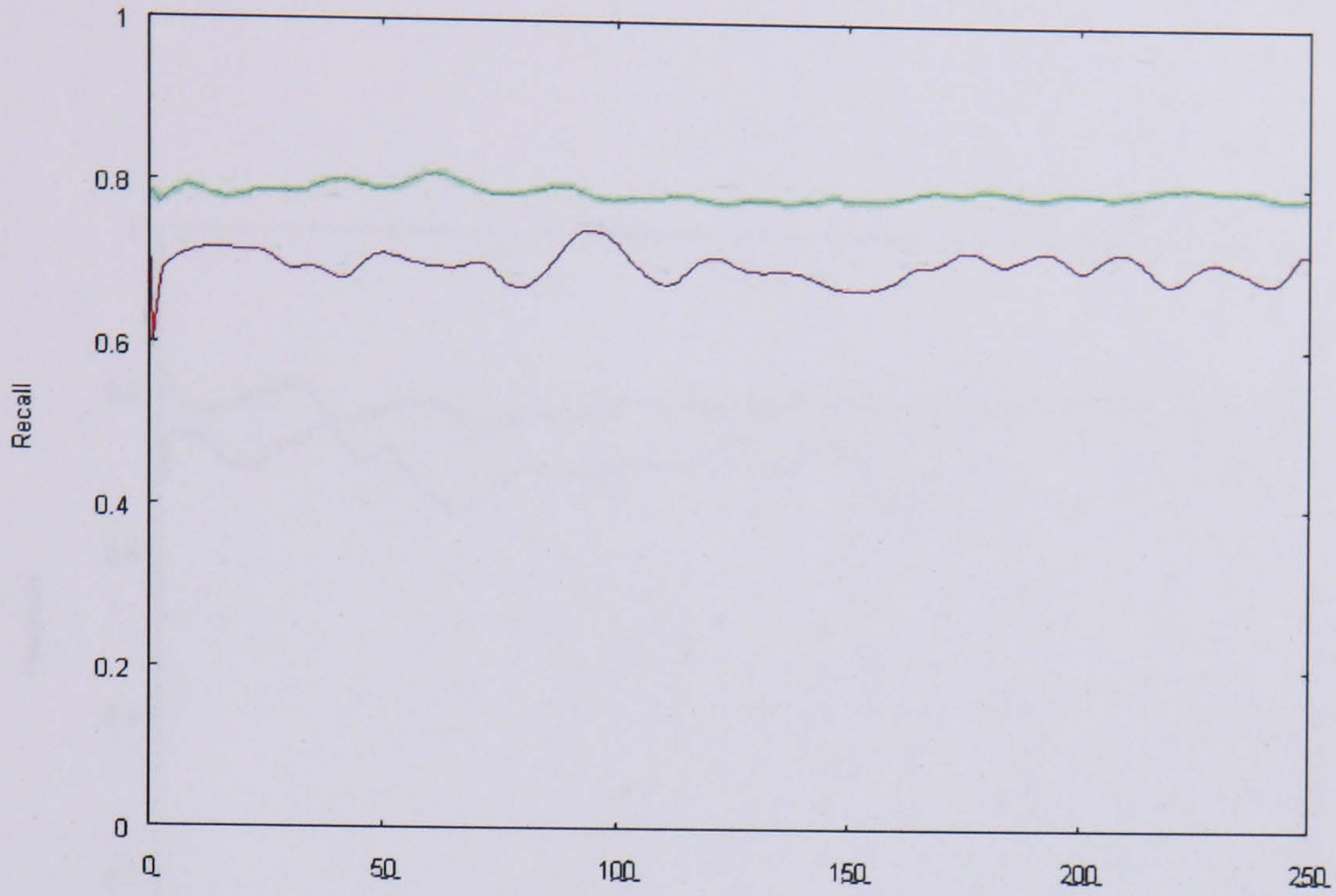


Figure 4.12: Continued: Recall along time for the 3 testing sequence.

generated by Algorithm 1. In the MCC-based assessment, for the ratio of Recall almost all of the values generated by Algorithm 1 are above 0.6, while those generated by Algorithm 2 are close to 0.8. For the ratio of Precision, the values generated by both are above 0.6. It can be concluded that the experimental results show the Algorithm 2 works better with most of the experimental sequences, but both outcomes are acceptable. The crowd dynamics estimation provides a suitable precursor to processes for determining the modes of complex dynamics, describing behaviour and acting as a support for work in high-level vision and socio-dynamics modelling.

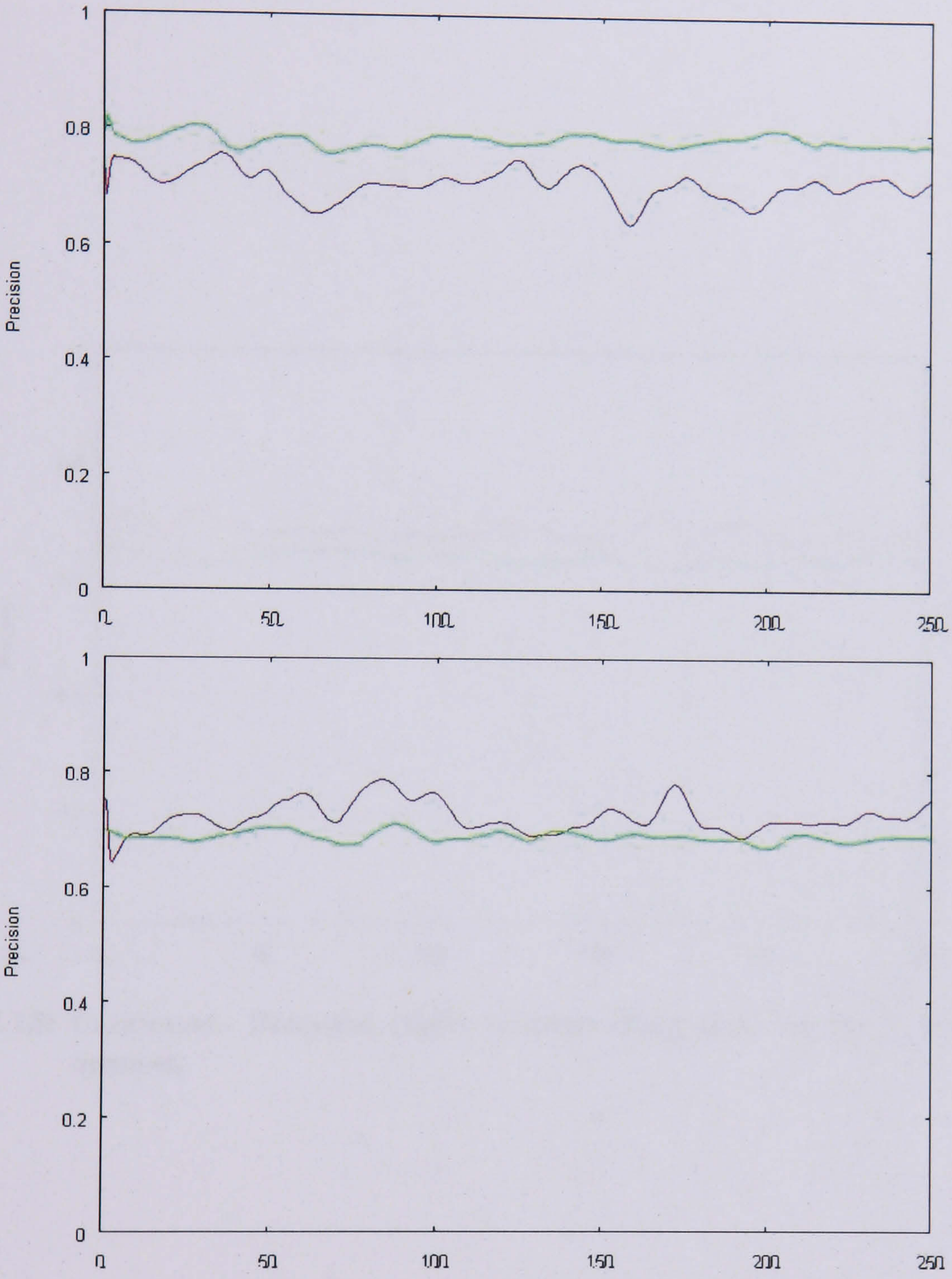


Figure 4.13: Precision (right column) along time for the 3 testing sequences, red lines for Algorithm 1; green lines for Algorithm 2. Algorithm 2 has higher values of Precision for two sequences.

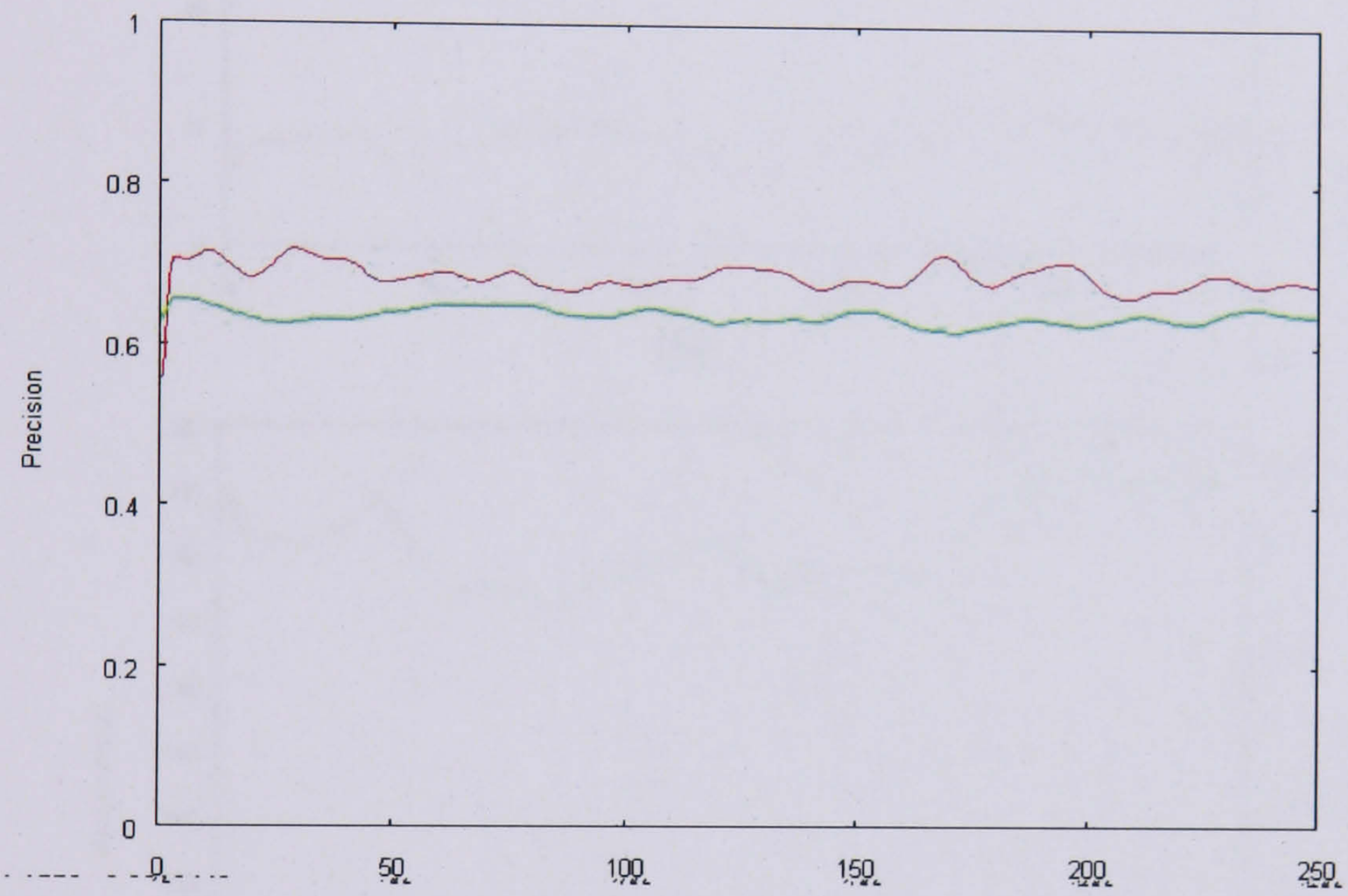
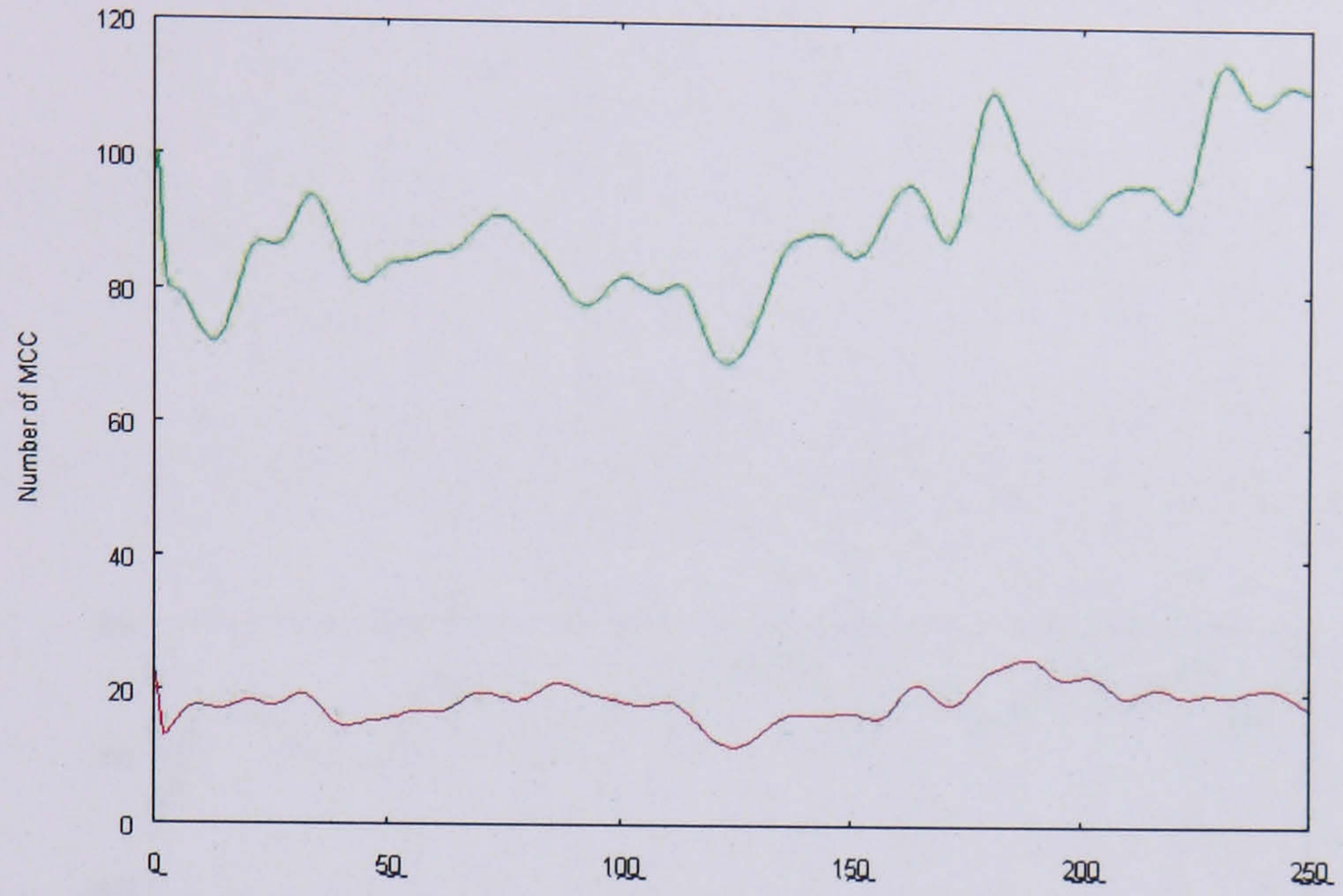
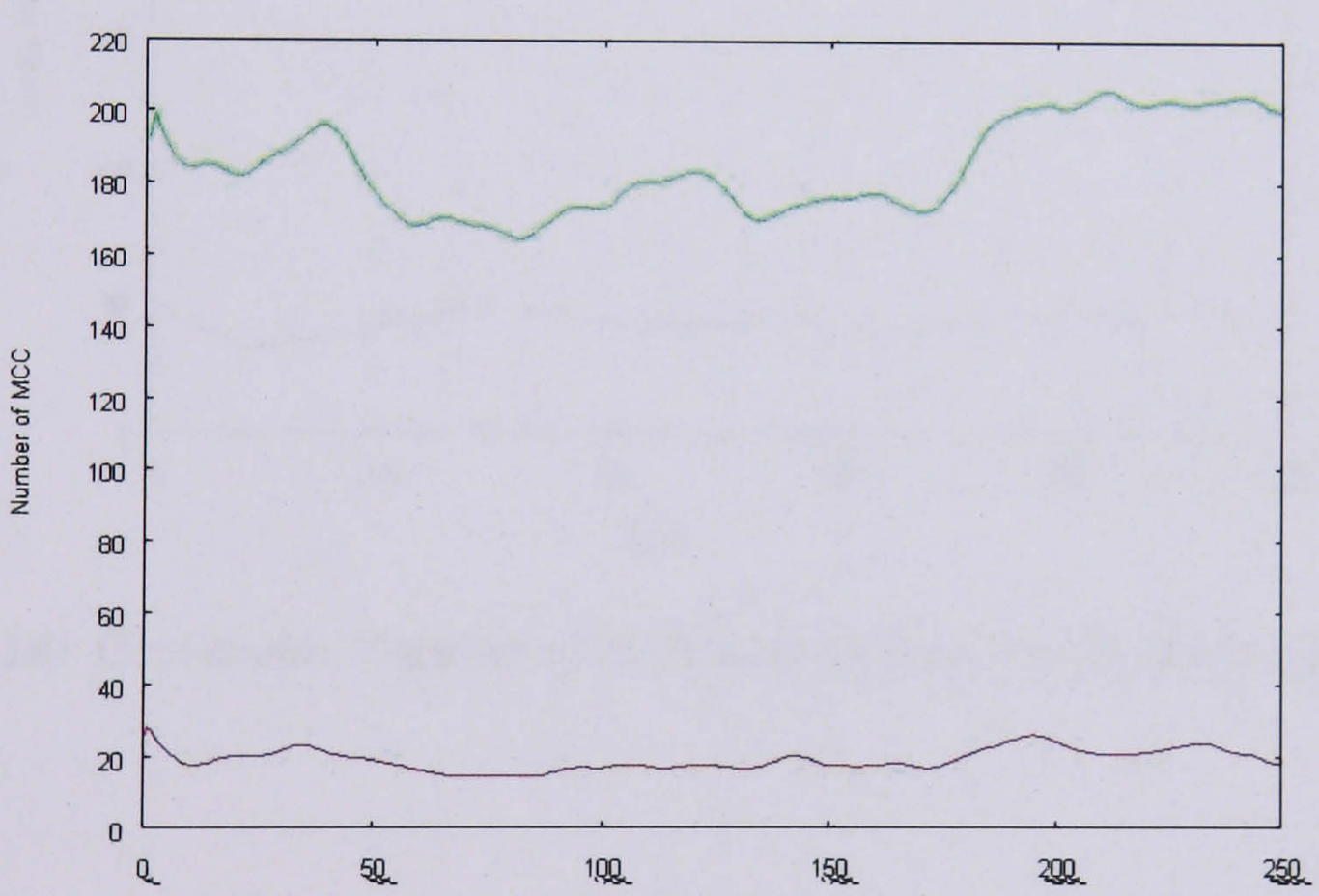


Figure 4.13: Continued: Precision (right column) along time for the 3 testing sequences.

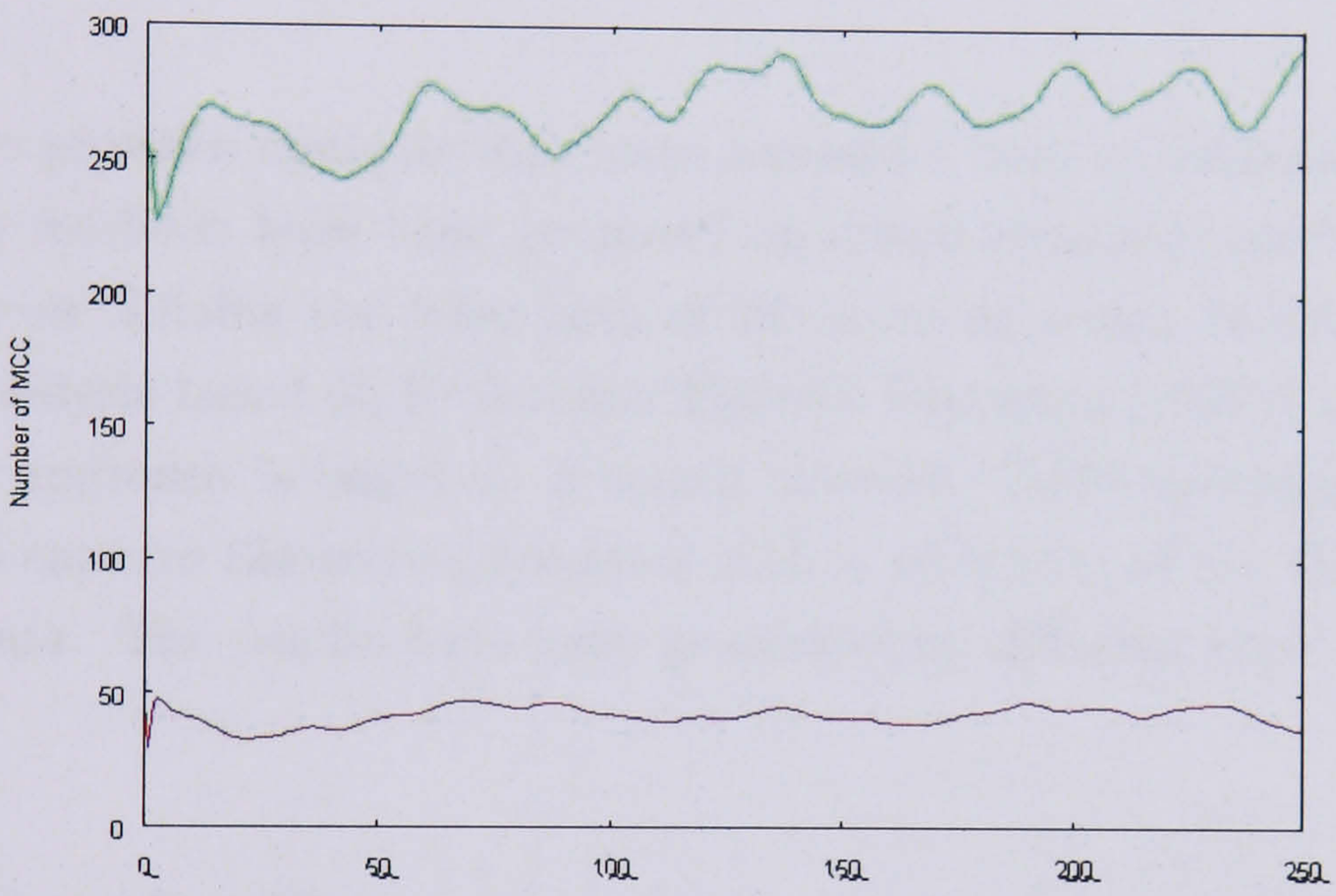


(a)



(b)

Figure 4.14: Number of MCCs along time for the 3 testing sequence, red lines for Algorithm 1; green lines for Algorithm 2 (From top to bottom: sequence 1, sequence 2 and sequence 3). Algorithm 3 detects many more MCCs for all of the three video sequences.



(a)

Figure 4.14: Continued: Number of MCCs along time for the 3 testing sequence

Chapter 5

Group and Crowd Modelling

This chapter provides methods that learn semantics from an extremely crowded scene. Two methods have been proposed on crowd dynamics modelling. The first focuses on defining the main path of the crowded scene. In this approach, statistical analysis based on Probability Density Functions (PDFs) is employed. The second approach is based on a neural network. Self-Organizing Maps are proposed to capture the crowd dynamics with a reduction of the dimensions of the input data. The results have been generated for different types of crowded scenes.

5.1 Introduction

A crowd is a familiar phenomenon studied in a variety of research disciplines including sociology, civil engineering and physics. Over the last two decades, computer vision has become increasingly interested in studying crowds and their dynamics because the phenomenon is of great scientific interest and offers new challenges. Moreover, an increase in video surveillance technology in public spaces has also led to its rise in popularity. A crowded scene is a huge challenge for computer vision techniques to be able to retrieve individual motion. On the other hand, in a crowded situation an individual's behaviour is most likely to be influenced by the overall crowd flow. In terms of crowd dynamics analysis, this chapter presents methods for capturing and learning the macro dynamics of the

whole crowd flow. The proposed work is implemented with simple machine vision algorithms that do not require sophisticated image understanding processing algorithms and that can be eventually implemented in hardware.

The overall objective of the research in this chapter is to model crowd dynamics in a macro scope. Statistical methods and self-organized maps are used to learn the dominant crowd dynamics. The statistical approach builds two Probability Density Functions(PDFs) - the occurrence PDF (PDF_{occ}) and orientation PDF(PDF_{or}) - accumulatively to represent the dynamics of the crowd. The method has provided a way to recover the major path of a crowded scene based on the PDFs. Experimental results and the evaluations are also presented.

A Self-Organizing Map (SOM) is widely used in mapping multidimensional data onto a low-dimensional map. Examples of applications include the analysis of banking data, linguistic data (80) and image classification (86). In this chapter, location and optical flow, a whole image frame and a whole motion field are used as input features to train the SOMs. Visualisation methods of the relevant SOMs, or neurons of the SOMs, are developed. The resulting SOMs from the location and optical flow inputs are compared structurally to classify the different scenes. Meanwhile, the resulting SOMs from the whole image frame and motion field are classified by the tracking of winning neurons.

This chapter is organised as follows: In Section 5.2, the statistical approach is presented, Section 5.3 introduces the Self-Organizing Maps approach and Section 5.4 gives the conclusion.

5.2 Statistical Approach

Crowds appear to move randomly within a scene. In fact, this is not exactly true because people move purposefully and their movements are guided by intentions. For instance, in a railway station or at an airport, people tend to enter and exit the scene at the gates and usually stop in front of a timetable, a shop or a cash point. Although at first chaotic, the video of a crowded place, if observed attentively, reveals main trajectories. In the following text, some crowd modelling work is introduced and statistical analysis that employs two Probability Density Functions (PDFs) is applied. The proposed method has the following two elements:

an occurrence PDF (PDF_{occ}), which is an accumulator of the foreground pixels, and an orientation PDF (PDF_{or}), which is an accumulator of block matching.

5.2.1 Occurrence PDF

This method assumes that the scene is not too crowded and the Gaussian mixture model (127) is used to build a model of the background of the scene. The foreground data is further processed to reduce noise. In particular, connected components have been implemented. Connectivity of the foreground pixels gives more accuracy to the foreground data and assures that only large foreground blobs are accepted for further analysis, while smaller blobs are rejected as likely noise.

For each frame foreground, features are accumulated for every pixel, so that after a relatively long video sequence the accumulator of the foreground occurrence throughout the whole image will have some information. The occurrence PDF (PDF_{or}) is thus constructed.

5.2.2 Orientation PDF

The image plane is segmented into a regular grid of cells ($N \times M$). The dimension of each cell is a multiple of 2 pixels and each cell is square-shaped ($K \times K$). The idea is to speed up the matching process employed as a coarse estimator of motion between the frames. Motion is estimated between consecutive frames, using the foreground blocks of the first frame as a reference/template and searching for an optimal match in the second frame. In the current implementation, block matching is carried out in a 3×3 neighbourhood, around the selected foreground cell. A cell is labelled as the foreground if the majority of its pixels are indeed foreground. Matching performance is improved by matching only between foreground cells and ignoring background cells. A correlation measure (123) is used to calculate the distance between cells. The correlation method used for each pixel is:

$$C(p_1, p_2) = \frac{1}{1 + (p_1 - p_2)}. \tag{5.1}$$

where p_1 and p_2 are respectively the pixel in the reference cell and the pixel in the neighbouring cell. Correlation for an entire cell is then calculated by summing over all the pixels of the cell:

$$C(cell_1, cell_2) = \sum C(p_i, p_j). \tag{5.2}$$

Each cell is therefore associated with a histogram, representing the eight possible directions of motion. The intention here is to build a local representation of motion, similar to a discrete reinforcement learning technique (131), where each cell of the table is associated with a quality array, indicating the likelihood of a transition from the current cell to a neighbouring cell. The final outcome is an orientation PDF, which could be interpreted as the global optical flow of the scene.

5.2.3 Path Discovery

The work described in the previous sections provides two PDFs - one for the occurrence and one for the orientation of a scene. To discover the main paths, the information and extracts of those corresponding to a higher likelihood/probability need to be combined. Ideally, the paths are identified that correspond to the modes of a probability density function that combines both occurrence and orientation information.

In order to estimate the main paths, a number of assumptions were made.

- *Path origin*: The assumption is that all paths originate from the boundaries of the scene. Consequently, path discovery starts from a cell on the boundary of the scene and has a high occurrence probability. This assumption would not work if the scene had an entrance or exit in the middle of the image, but this can be overcome relatively easily by using user-defined boundaries.

- *Graceful continuation/Smooth trajectory*: As observed, the paths have a high probability to maintain their orientation (e.g. people are more likely to go in a straight line, and seldom go backwards.) So the expected direction

0	1	2	3	4
0.6830	0.1335	0.02	0.0045	0.0001

Table 5.1: Likelihood as a function of orientation distance

of motion is modelled with a Poisson distribution with its maximum in the neighbouring cell along the current direction of motion.

The idea is to spread the likelihood of a change in direction unevenly, maintaining the previous orientation as the one at the highest probability and forcing the other directions (change in direction) to have a lower likelihood. Table 5.1 illustrates the probabilities used given the distance from the current orientation. From the start point, the probability is calculated for each neighbouring block using the occurrence PDF (PDF_{occ}), the block matching accumulator (P_b) and the orientation probability (PDF_{or}). Furthermore, to avoid repeating calculations from the same block, the visited cells are marked, and their probability is set to to 0 each time the path discovery process has to deal with them. The probability is defined as:

$$P_i = \frac{m_i \cdots P_i^b \cdots PDF_i^{occ} \cdots PDF_i^{or}}{\sum m_k \cdots P_k^b \cdots PDF_k^{occ} \cdots PDF_k^{or}} \; k \in [0,8], \; m_i = \begin{cases} 0, \; marked \\ 1, \; unmarked. \end{cases} \tag{5.3}$$

The process will follow the highest probability block. In addition, a way of deciding when to split a trajectory into two or more sub-trajectories is devised. This technique works on a threshold that estimates whether two or more paths are viable given their associated likelihood. However, in order not to generate too many branches, only a single split along a trajectory is admitted.

Once all the paths are identified, a fitting process takes place. This serves two purposes: (i) to have a compact representation of the path, and (ii) to have a faster way of estimating the distance between a blob/bounding rectangle, identified by new foreground data and the spline, and consequently estimating an error. The final path is represented as a curve by fitting a uniform Cubic B-spline.

5.2.4 Evaluation

The paths extracted using the method described in the previous sections corresponds to the main modes of trajectories followed by people in the analysed scene. Rather than using the two PDFs (occurrence and orientation) to estimate an error and evaluate the performance of the technique, a simplified evaluation is provided. The idea of a stripe, which is along the discovered paths using a decay factor (a Gaussian weighting) along the perpendicular to the trajectory, is employed.

The stripe is illustrated pictorially in Figure 5.1. Suppose the black area represents the discovered path $f = f(x, y, t)$. A Gaussian distribution $G(\mu, \sigma)$ is then centred on the trajectory (μ corresponding to the generic path pixel), and is a pre-determined standard deviation directly proportional to the size of the blobs estimated by the connected component process. An approximated estimate of the error between a new sequence of the same scene and the built model can then be calculated by weighting the contribution of a foreground blob, making use of the described weighting scheme. Since error estimation can be performed offline, when the model already exists a mask for the entire image can be built before testing. Masks for all are built only once at the end of the path modelling process.

- An image look-up table (LUT) is built, where each pixel is assigned a label, identifying the closest path in the scene.
- For each path a stripe mask is built. The mask contains the weights, inversely proportional to the distance between a pixel and the path/spline. To calculate the weights the curve of the path is sampled at equally spaced intervals Δt and uses the line segment between samples to calculate the weight.

Each FG blob detected is examined pixel-by-pixel with the image label LUT, and determines the closest path by taking the most frequent label of its pixels. The following two tables show the results achieved by using the current evaluation methods. Table 5.2 illustrates tests on 10 short video sequences of equal periods of time (50 frames) from two types of videos (5 sequences each). The first row ("FG

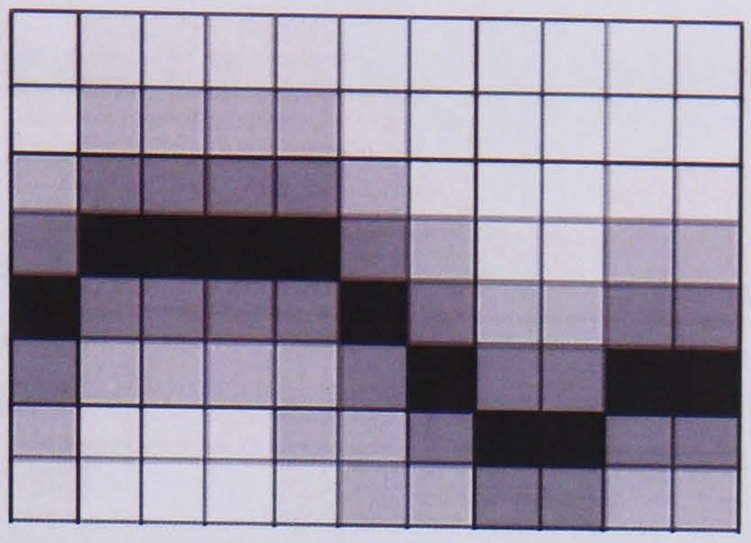


Figure 5.1: Stripe

Table 5.2: Results using stripe evaluation

Scene A	FG size	47680	76835	99574	103956	112968
	Fit	0.849	0.850	0.851	0.848	0.853
Scene B	FG size	32835	44663	66736	98387	134316
	Fit	0.838	0.833	0.838	0.849	0.848

size") of each sequence represents the total amount of foreground pixels tested in that period, while the second row ("Fit") is the normalised fit rate (1: perfect fit). The fit rate keeps at more than 0.8 in all the test sequences, and is not affected by the amount of the foreground pixels (which represent the cluttered levels in the sequences). In the second test, a number of scenes have been analysed. Following the conventional machine learning approach, each sequence was split in two halves to build and test the model. Different percentages of frames were used to build the model and to test the robustness of the approach. The Kullback-Leibler (KL) dissimilarity measure is chosen to estimate the similarity between the PDF_{occ} and PDF_{or} of the model and the corresponding PDFs built using a fixed percentage of test data. For the two probability functions p and q, Kullback-Leibler (KL) dissimilarity measures the expected difference between them. This is defined as:

$$D(p||q) = \sum p(t) \log_2(\frac{p(t)}{q(t)}) \tag{5.4}$$

And it is applied here as:

$$\widehat{D}_{KL} = (D(PDF^{model}||PDF^{test})) \oplus (D(PDF^{test}||PDF^{model})) \tag{5.5}$$

Table 5.3: Table of KL distance

Scene A

	PDF_{occ}			PDF_{or}		
N_{frames}	$D(p\|\|q)$	$D(q\|\|p)$	$D(p\|\|q) \oplus D(q\|\|p)$	$D(p\|\|q)$	$D(q\|\|p)$	$D(p\|\|q) \oplus D(q\|\|p)$
200	1.38744	6.41181	3.89963	0.992265	3.71003	2.35115
400	1.22145	4.72941	2.97543	0.74336	2.3779	1.56063
600	1.30149	4.22187	2.76168	0.550128	0.870776	0.710452
800	5.32319	1.50177	3.41248	0.960938	0.405281	0.68311
1000	5.43141	1.27666	3.40404	1.61853	0.448835	1.03368
1200	5.83901	1.39313	3.61607	2.20614	0.508669	1.3574
1400	5.87275	1.38677	3.62976	2.48443	0.547993	1.51621

Scene B

	PDF_{occ}			PDF_{or}		
N_{frames}	$D(p\|\|q)$	$D(q\|\|p)$	$D(p\|\|q) \oplus D(q\|\|p)$	$D(p\|\|q)$	$D(q\|\|p)$	$D(p\|\|q) \oplus D(q\|\|p)$
200	1.76456	4.49901	3.13179	1.07256	6.88155	3.97705
400	1.75548	3.60504	2.68026	0.665695	2.40205	1.53387
600	1.9543	2.89671	2.42506	0.434972	0.825922	0.630447
800	2.15736	2.32519	2.24128	0.395621	0.502728	0.449173
1000	3.9971	1.39806	2.69758	0.529596	0.429419	0.479507
1200	3.65854	1.29503	2.47678	0.653405	0.403238	0.528322
1400	3.62649	1.30134	2.46391	0.680506	0.403089	0.541798

and, for PDF_{occ} the sum is over the entire image, and for PDF_{or} is a weighted sum over all the cells.

Table 5.3 illustrates some preliminary results. The table illustrates results for two scenes, indicating the dissimilarity for PDF_{occ} and PDF_{or} independently. The composite, shown with the symbol $\oplus$, is a type of balanced non-negative dissimilarity measure that, in theory, should decrease as the model is refined; this better represents the studied scene. These preliminary outcomes illustrate that a decreasing trend is present for PDF_{or} but not quite for PDF_{occ}. The number of frames used is still fairly low due to the lack of video data. A longer sequence would be used if possible:

5.3 Self-Organizing Map Approach

The approach in Section 5.2is based on background modelling, which cannot work properly under extremely crowded situations. Also, the number of dimensions is $W \times H$ for PDF_{occ} and $W \times H \times 2$ for PDF_{or} (where W is the width and H is the height of the image sequence). As a result, the dimension of the model is dependent on the dimension of the image sequence. These, however, are disadvantages that can be overcome by the method described in this section.

In this section, a Self-Organizing Map is proposed to learn the dominant crowd dynamics. The Self-Organizing Map (SOM) model (51) is a well known dimensionality reduction method, proven to bear a resemblance to some of the features of the human brain that represent different sensory inputs by topologically ordered computational maps. SOMs are widely used in mapping multidimensional data onto a low-dimensional map, examples of which include applications such as the analysis of banking data, linguistic data (80) and image classification (86). This section proposes a system that learns crowd dynamics with the SOM. The system uses dynamics information as an input and generates SOMs that capture the dominant recurrent dynamics.

5.3.1 Background

The most common SOMs have neurons organised as nodes in a one- or two-dimensional lattice. The neurons of an SOM are activated by input patterns in the course of a competitive learning process. At any moment in time, only one output neuron is active - the so called winning neuron. Input patterns come from a n-dimensional input space and are then mapped to the one- or two-dimensional output space of the SOM. Every neuron has a weight vector that belongs to the input space (51).

There are two phases for tuning the SOM with an input pattern X: competing and updating. In the competing phase, every neuron is compared with X, the similarity of X and the weights of all of the neurons are computed, and the neuron $N(i_w, j_w)$(denoted by the neuron's coordinates of the lattice) with the highest similarity is selected as the winning neuron. In the work discussed in this chapter, a two-dimensional lattice is used. For each neuron $N(i, j)$, Ecludian

distance is employed:

$$d^2 = (i - i_w)^2 + (j - j_w)^2 \tag{5.6}$$

the topological neighbourhood function is then defined as:

$$h(n) = \exp(-\frac{d^2}{2\sigma^2(n)}) \tag{5.7}$$

where n denotes the time, which can also be explained as the number of iterations. and $\sigma^2(n)$ decreases with the time. In this work the dependence of σ on discrete time n is chosen as:

$$\sigma(n) = \sigma_0 \exp(-\frac{n}{\tau_1}) \tag{5.8}$$

where σ_0 is the initial value of σ and τ_1 is a time constant. The weight of each neuron $N(i, j)$ at time $n + 1$ is then defined by:

$$w(n + 1) = w(n) + \eta(n)h(n)(x - w(n)) \tag{5.9}$$

where $w(n)$ and $w(n + 1)$ is the weight of the neuron at times n and $n + 1$. $\eta(n)$ is the function of the learning rate, which always decreases with time. The decreasing of η has been defined similarly as for σ, where:

$$\eta(n) = \eta_0 \exp(-\frac{n}{\tau_2}) \tag{5.10}$$

where η_0 is the initial value of η and τ_2 is another time constant.

5.3.2 Optical Flow Input

The SOM in this approach should capture the two major components of the crowd dynamics, occurrence and orientation. Thus, a four-dimensional input space is chosen to be the weight space of the SOM, which can be represented as $f : (x, y, \theta, \rho)$. Each piece of data from the input space can be explained as the location where the crowd moves and motion vectors in the form of an angle (θ) and magnitude (ρ). To reduce the computation load from a dense flow, a feature-based optical flow is employed (22). The dimension of the weight space is $N \times 4$, where N equals the number of features in the frame. In this approach, N is

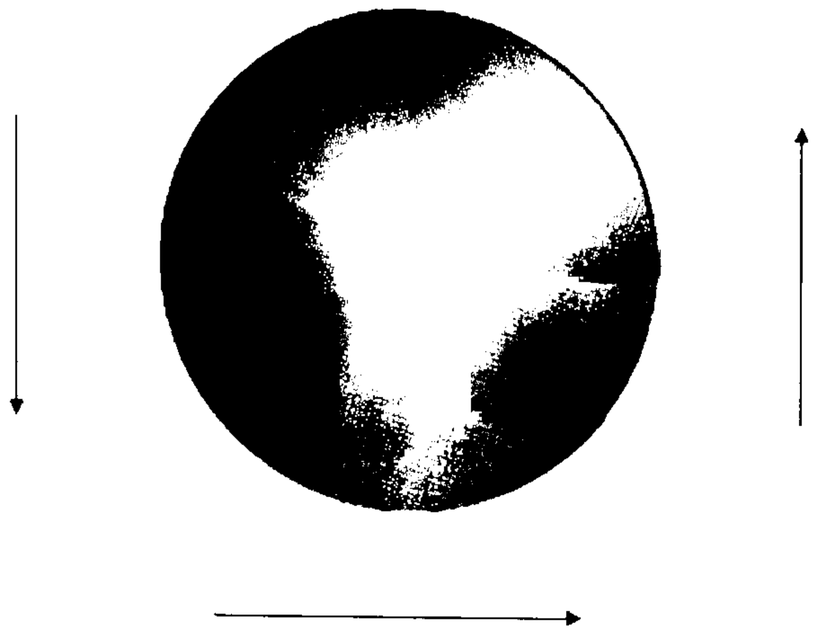


Figure 5.2: HSV representation of the orientations of motion vectors, relative orientations are the tangents anticlockwise: e.g. red for moving left.

related to the dimension of the image sequence, and typically equals $\frac{W \times H}{K}$, where K is a constant. Moreover, the dimension of the output space of the SOM is $n \times n$, which is equal to the dimension of the lattice. In the experiments presented here, N is normally between 5000 and 10000 and n is 10-20. The dimensional reduction is from $N \times 4$ to $n \times n$.

5.3.2.1 Visualization

Figure 5.3 illustrates three different video sequences with different dynamics. These video sequences have been input into the system, and Figure 5.4 shows the output SOMs. In the figure, SOMs are visualised in the input space, i.e. showing the weight vector of each neuron. In the visualisation, the coloured arrows and their locations are derived from the weight vector of the neurons, and the locations of the arrows are from the first two components of the weight vectors (x, y). The arrows show the second two components - the components of motion (θ, ρ). The different colours of the arrows also indicate the different orientation of the motion. The visualisation of the motion vectors is based on a HSV colour space representation, which is illustrated in Figure 5.2. In the first video (the left column in Figure 5.3), the major crowd is moving from the bottom left to the top right of the scene. There is another crowd flow from the bottom right of the scene which joins the major flow. In its SOM (the first one in Figure 5.4),

the neurons with green arrows are clearly from the major flow and the ones with red and purple arrows are from the minor flow. The second video (the middle column in Figure 5.3 is the area of an entrance to a public space. Most of the people move from the top to the bottom of the scene. The crowd in the upper part of the scene is sparser and moves faster when compared to the crowd in the lower part of the scene. There is also a minor flow, which joins the major flow from the right of the scene. In the built SOM (the second SOM in Figure 5.4), again the flows are clearly indicated. Furthermore, the SOM takes an "umbrella" shape, which represents the shape of the flow constrained by the obstacles in the scene. In the third video (the right column in Figure 5.3), the scene is of a large open area with multiple crowd flows. The major flow moves from right to left; however, there are several minor flows, most of which are in the lower part of the scene. Again, the SOM (the third in Figure 5.4) captures the major dynamics and also some minor flows. From the three examples, it can be concluded that the SOMs not only preserve the dominant motion vector, but also represent the shape of the regions with a dominant motion of the scenes.

5.3.2.2 Scene classification

Visualisations of the SOMs have already provided some information on recurrent motion and scene classification has been carried out using the characters captured by the SOMs. To achieve this, comparisons with the SOMs built for different scenes have been carried out. The classification is based on the similarities of the SOMs. The topological structures of the lattice of SOMs, as well as the weights of the neurons of the SOMs, are used for the comparison. The topological structure is an important feature of SOM, and a large number of methods have been proposed to measure it (107). In this work a C-Measure is used, which is defined as: The similarity of the C-Measures of two different SOMs is calculated as:

$$C = \sum_{(i,j \in_{i \neq j}) A \times A} F_A(i,j) F_V(w_i, w_j) \tag{5.11}$$

where F_A and F_V are the similarities between the input space (i.e. weight space) and output space (i.e. SOM lattice), respectively. The i and j are the indexes of

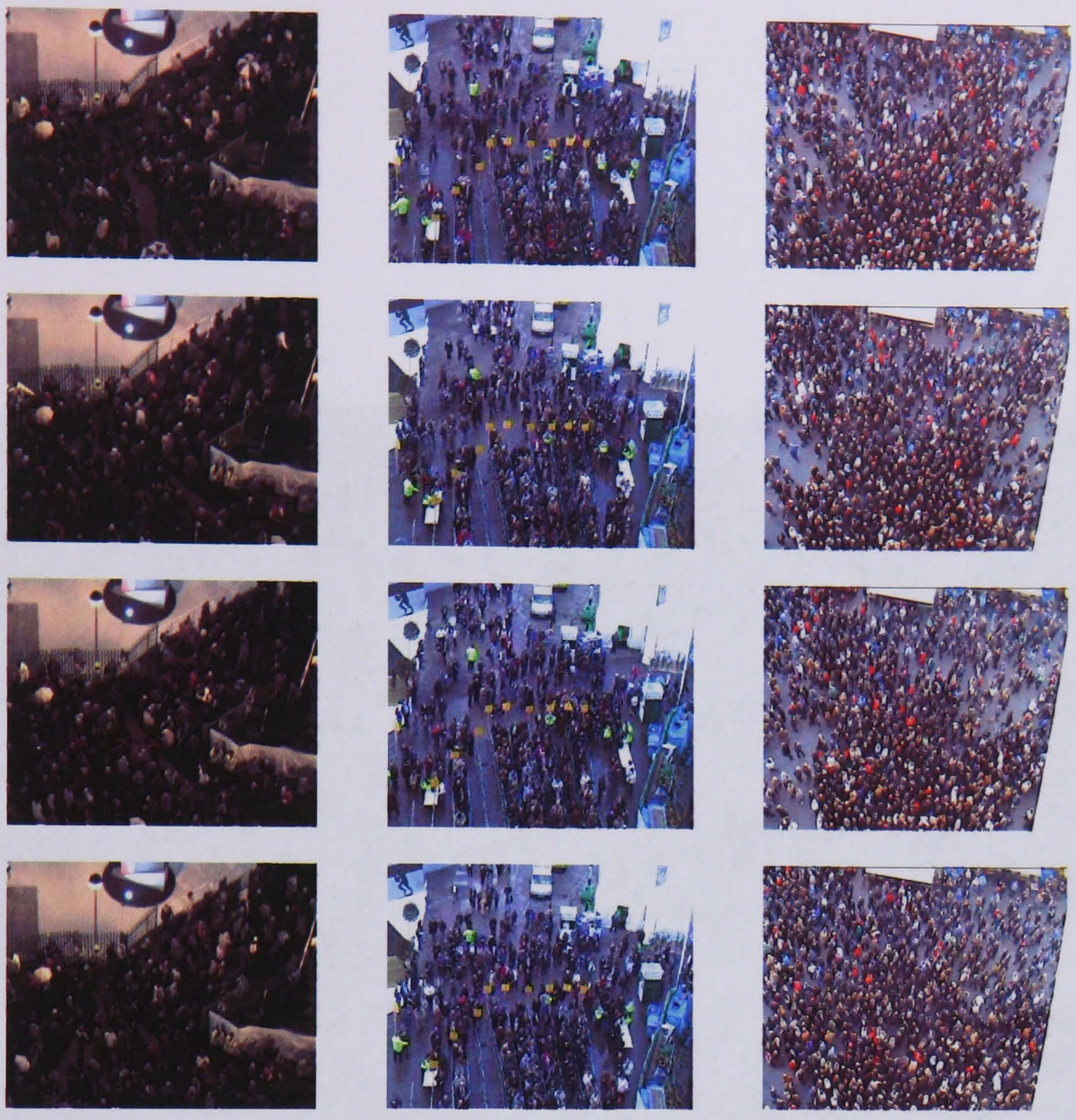


Figure 5.3: The example frames from three different scenes.

the neurons and w_i and w_j are the weights of the indexed neurons. The similarity between corresponding SOM neurons is calculated using the same method as in Equation 4.3 in Chapter 4:

$$Sim_w = F_V(w_i, w_j) = \sum_{k=0}^{k<Dim_w} \frac{min(w_i^k, w_j^k)}{max(w_i^k, w_j^k)} \tag{5.12}$$

where Dim_w is the dimension of the weight space, and w_i^k and w_j^k are the $k-th$ element of the weights w_i and w_j, respectively. An average over the lattice has been calculated. This equation is used for calculating the similarity of the weights of two neurons. The similarity of the structure is calculated as the similarity of

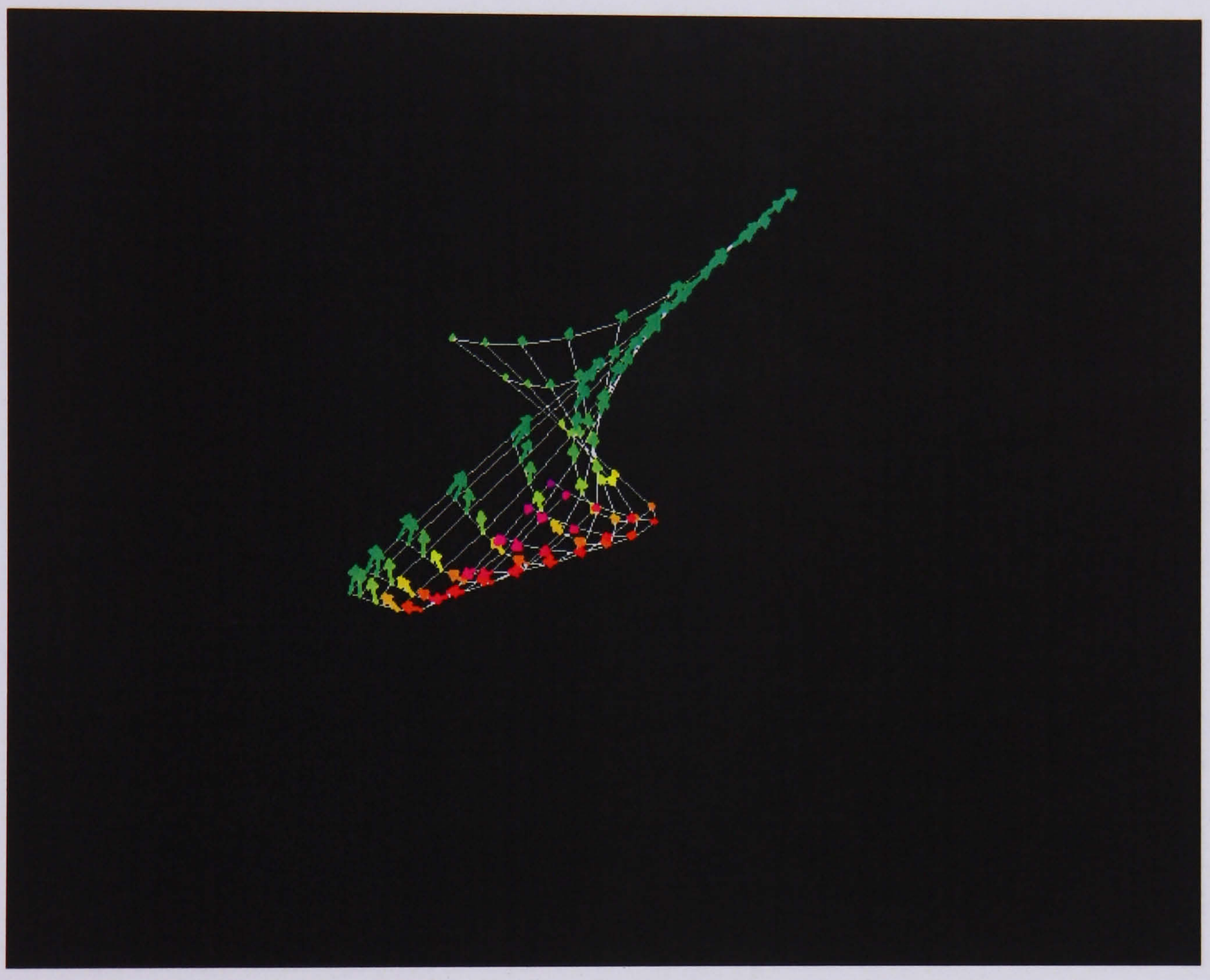


Figure 5.4: The visualisation of built SOMs for the scene illustrated in the left row of 5.3

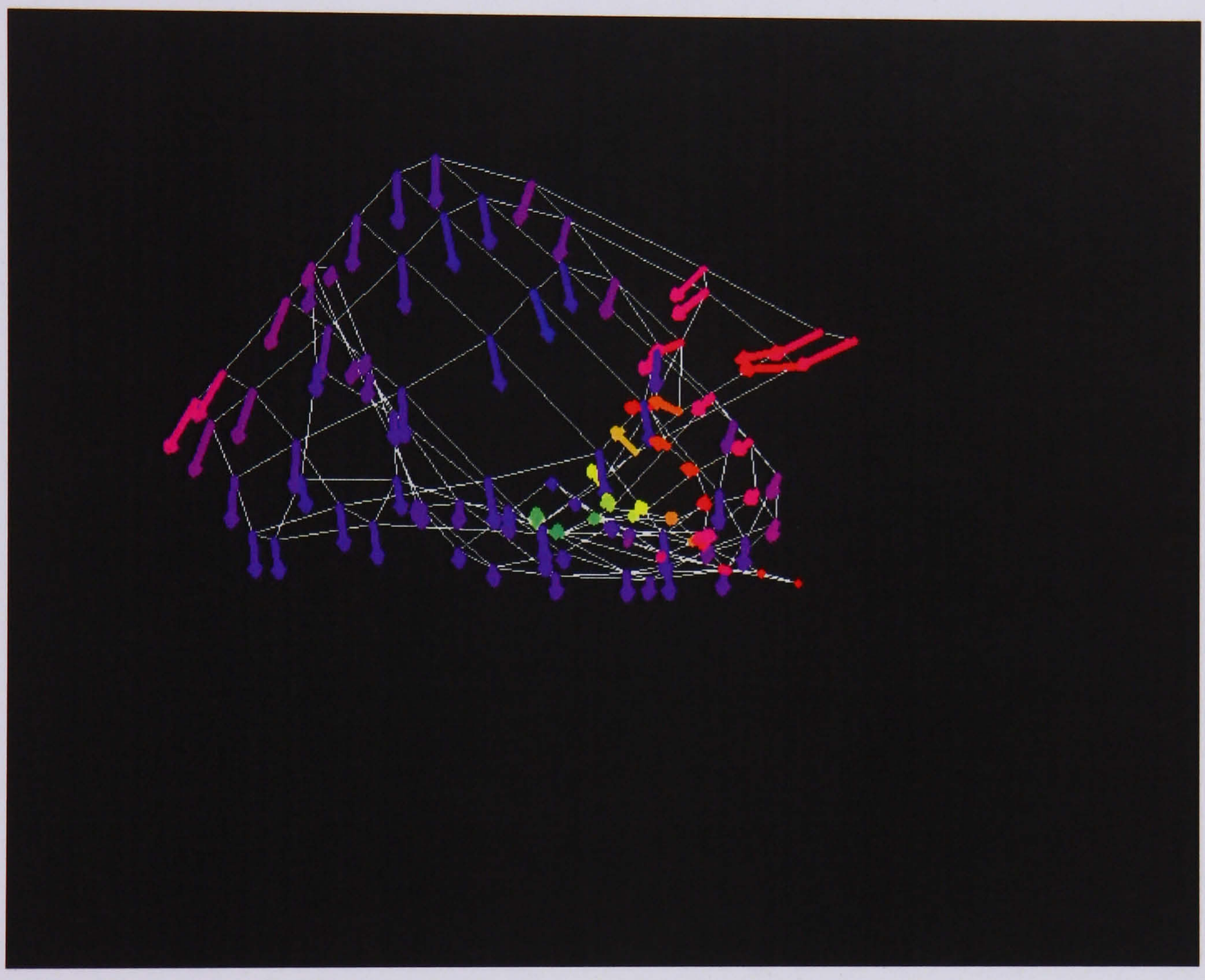


Figure 5.4: The visualisation of built SOMs for the scene illustrated in the middle row of 5.3

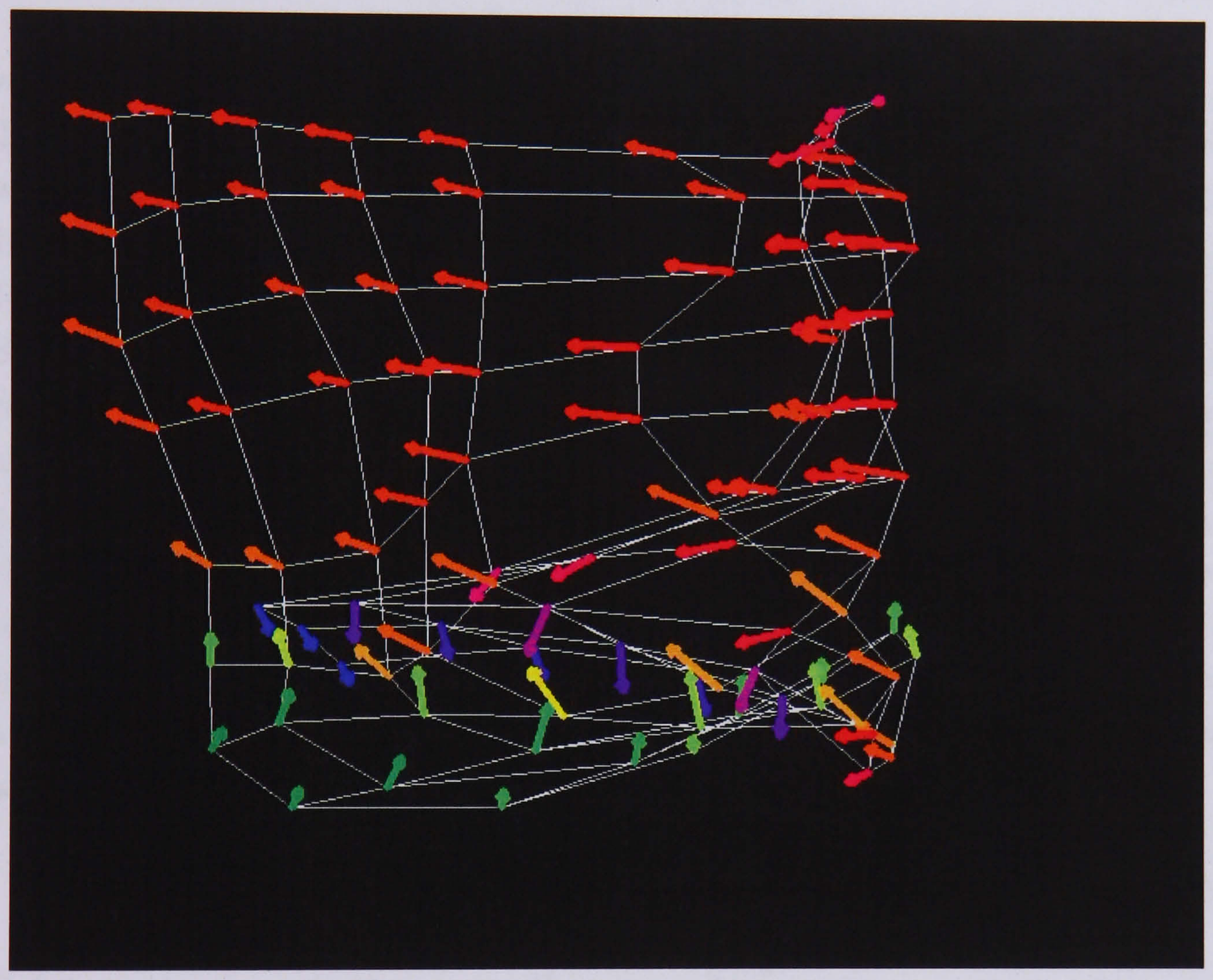


Figure 5.4: The visualisation of built SOMs for the scene illustrated in the right row of 5.3

the C-Measures:

$$Sim_c = \frac{min(C_i, C_j)}{max(C_i, C_j)} \tag{5.13}$$

Sim_w and Sim_c are in the range [0,1]. A combination of the two similarities - weight similarity and structure similarity - are calculated by:

$$Sim = \sqrt{Sim_c \times \overline{Sim_w}} \tag{5.14}$$

where $\overline{Sim_w}$ is the average of Sim_w over the lattice. Again, Sim falls in the [0,1]. The correspondence of the neurons is defined by the closeness of the values of their weights. Particularly for neuron (i) in SOM A, the corresponding neuron in SOM B is the one with the closest weight value. The matching could be asymmetrical. For example, assuming for neuron (i) in SOM A, its corresponding neuron in SOM B is neuron (j); however, for the neuron (j) its corresponding neuron in SOM A is not necessarily neuron (i). This can be caused by the situation whereby the neurons from SOM A and B are not in the same value scale. In some extreme cases, all the neurons in one SOM could even be matched to the same neuron in the other SOM. As a result, another interesting figure is the number of matched neurons in SOM B. The value number of matched $N_{matched}$ neurons is normalised by dividing the total number of neurons N_{total}. This figure will indicate if the values of the weights in SOM A are in the same scale as SOM B, and is combined with the last measure by:

$$S = \frac{Sim + N_{matched}/N_{total}}{2} \tag{5.15}$$

S falls in the range [0,1] as well. The comparison is not symmetric, which also means that if SOM A is compared with SOM B, the result will be different from using SOM B to compare with SOM A. Consequently, two similarities are generated from the comparison of the two SOMs. This experiment takes three scenes, and two sequences are extracted from each scene so that there are 6 sequences in total in the experiment. The following confusion matrix illustrates the relative results. In Table 5.3.2.2, each row has the similarity value of an SOM with the other sequences. There are two values: the similarity of SOM A compared to SOM B and the similarity of SOM B compared to SOM A. The values

Table 5.4: Confusion matrix of SOMs from different scenes (Scn abbreviates Scene)

	Scn A - 1	Scn A - 2	Scn B - 1	Scn B - 2	Scn C - 1	Scn C - 2
Scene A - 1	1	**0.653097**	0.484192	0.433234	0.468101	0.458993
Scene A - 2	**0.633261**	1	0.372017	0.315155	0.426438	0.400024
Scene B - 1	0.330897	0.33033	1	**0.645102**	0.4264	0.465297
Scene B - 2	0.35838	0.332804	**0.641464**	1	0.467114	0.455613
Scene C - 1	0.369259	0.400326	0.443745	0.426589	1	**0.715606**
Scene C - 2	0.366577	0.318921	0.414272	0.429349	**0.687943**	1

above 0.5 are in bold font in the table, and they are all from the video sequences from the same scenes. From both visualisation and quantitative comparison, it can be concluded that the SOMs have captured the major dynamics of the crowded scenes.

5.3.3 Raw image as Input

In this application, the whole image is regarded as an input feature for the SOM. The raw data has been used with three channel colour images. In other words, the weight of the SOM is in a $W \times H \times 3$ space, where W and H are the width and height of the image, respectively. The dimensions of the input video data are reduced from $W \times H \times 3$ (Image space) to $(n \times n)$ (lattice space). The neurons of the SOMs retain the different status of the particular scene. Some selected neurons from SOMs constructed by raw images are illustrated in Figure 5.5 and 5.6. In the first scene, the neurons illustrate the different crowd status of the square, as well as some trajectories of the crowd. In the second case, the changes in position of the camera are captured, which can be inferred from the changing of locations of the grid on the floor. The above experiments are carried out over a video sequence with only one single crowded scene. In the following experiment, the SOM is built from video sequences consisting of more than one crowd scene. Figure 5.7 shows two neurons from the SOM built by a video sequence that contains two different crowded scenes. The two neurons indicate that the built SOM has modelled the two different scenes (Example frames from the two scenes can be found in the first two columns of Figure 5.3).

Figure 5.5: Selected SOM neurons built from a single scene, which captured the different trajectories and groups of people that are not moving

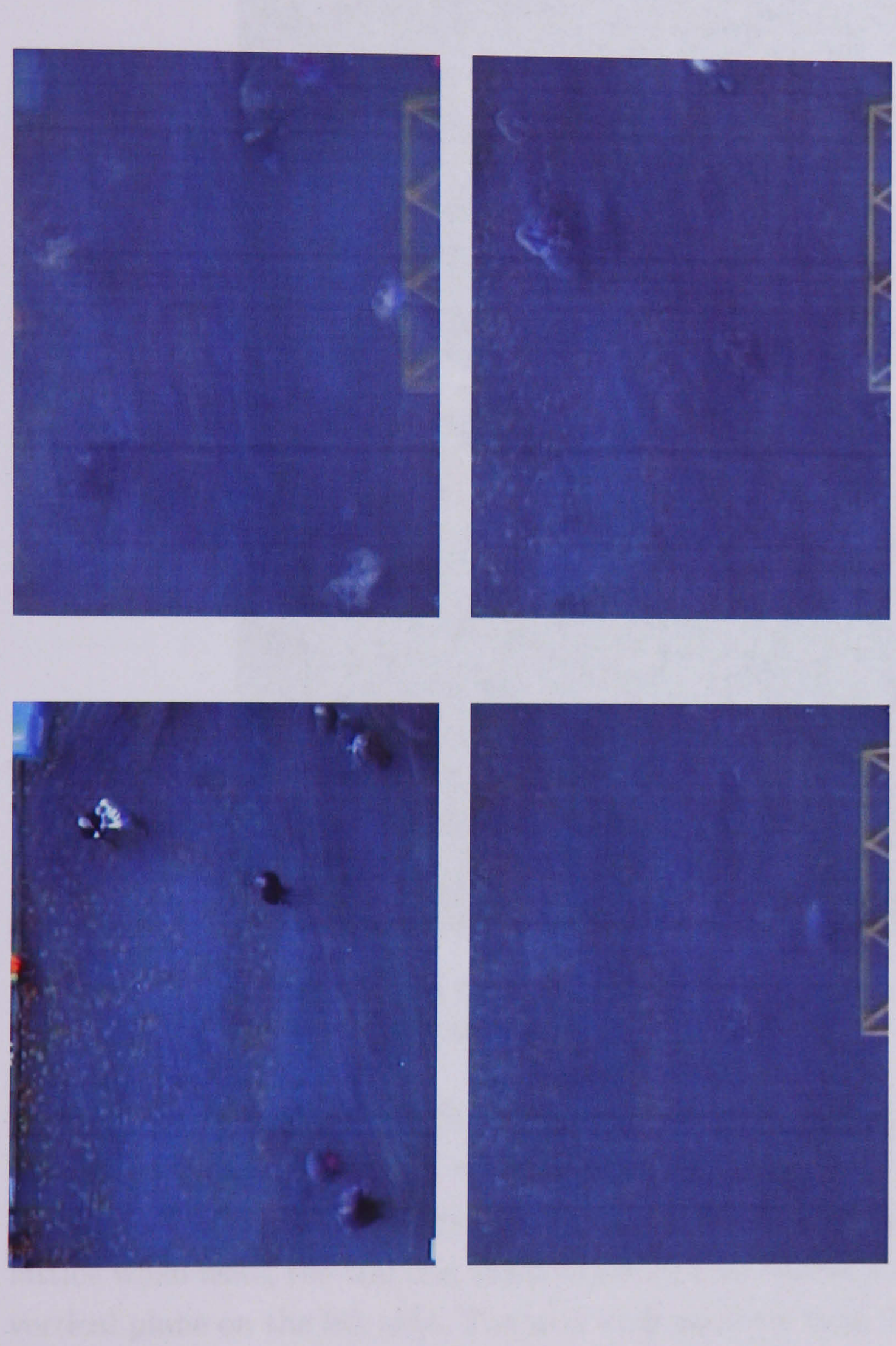


Figure 5.6: Selected SOM neurons from another single scene, which captured the different camera views

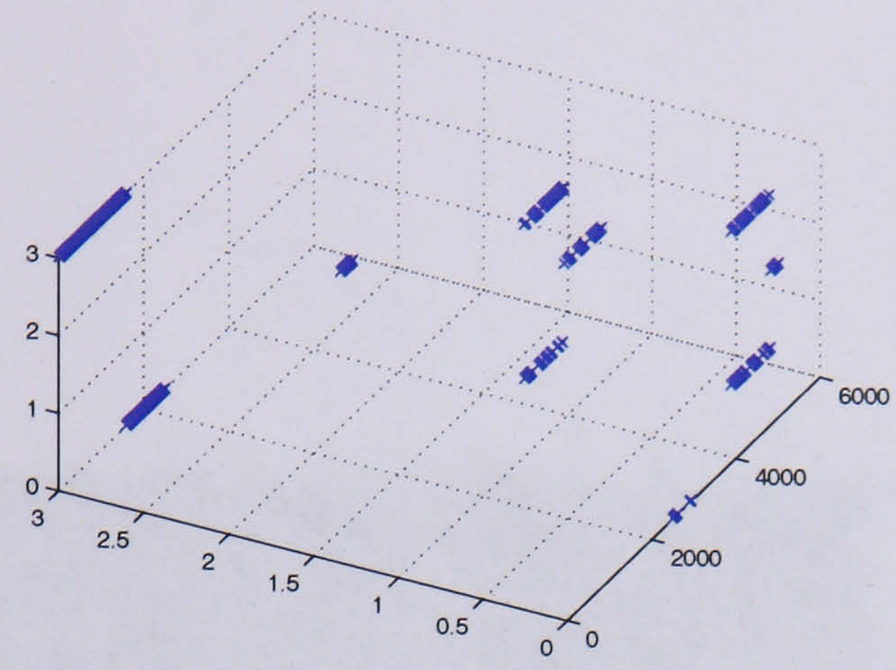
Figure 5.7: Two neurons from the SOM built by a video sequence which contains the first and second scenes in 5.3

Different neurons of the SOM represent different dynamics in the video sequence. The tracking of the winning neurons indicates the transition between dynamics. Figure 5.8(a) shows the changes of the winning neurons on the SOM lattice when using the training video sequence (the coordinates are shown on the vertical plane on the left side. The axis with numbers from 0 to 5000 is the time line.) There is an obvious transition between the winning neurons in the middle of the time line where it represents the changing of the scenes. New image sequences from the two scenes are used as inputs to the SOM to test its ability

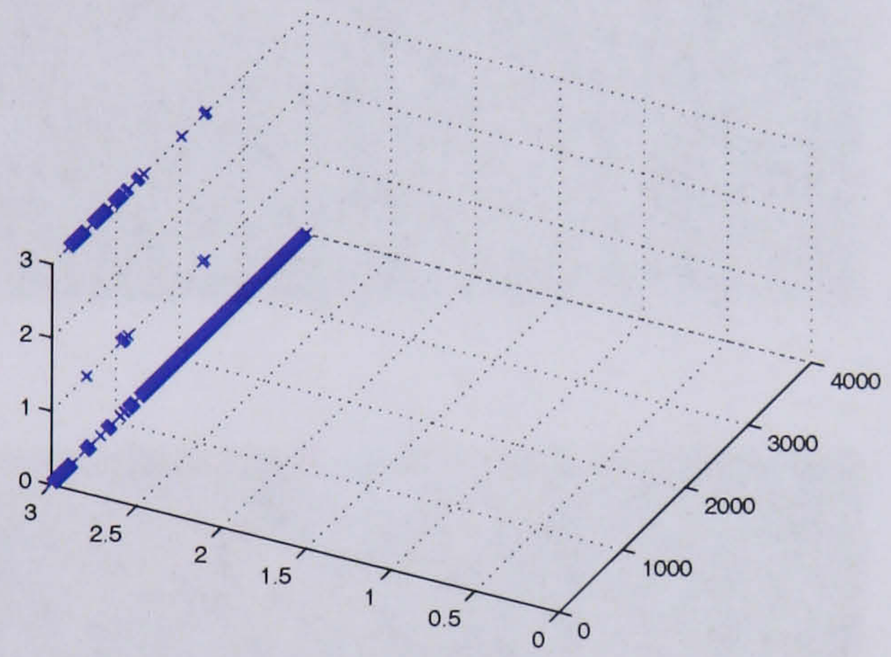
for scene classification. Figures 5.8(b) and 5.8(c) are the result of tracking the winning neuron over time. The winning neurons of the first scene are on the same plane, and for the second scene the winning neurons never get to the previous plane. Figure 5.9 and Figure 5.10 illustrate the results from another test, with another video consisting of two crowded scenes.

5.3.4 Motion field input

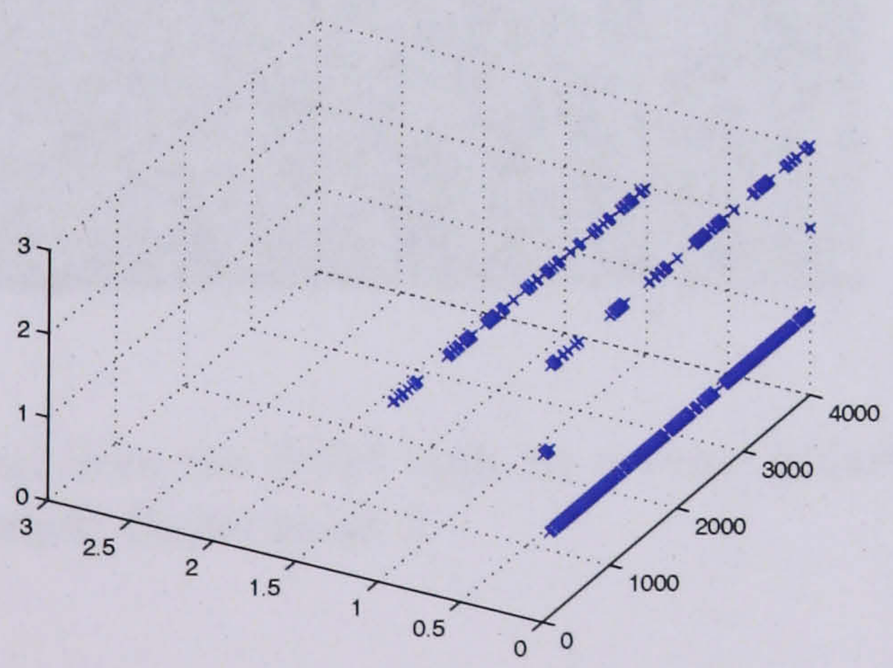
This time motion field for every frame is selected as the input feature. For example, if there are N frames in a video sequence and each frame is compared with the next frames to generate a motion field, there will be $N-1$ inputs for the SOM. The motion field is pixel-based, so each pixel has a pair of motion vectors $< vx, vy >$. Consequently, the dimension of the weight space is $Dim = W \times H \times 2$. By using the SOM, the dimensions of the input video data are reduced from $W \times H \times 4$ (motion field space) to $(n \times n)$ (lattice space). Some selected neurons from the built SOMs are visualised in Figure 5.11 and Figure 5.12, which represent the different dynamics statuses of the crowded scene. As with the visualisation method in the optical flow input section, the different colours show the different orientation of the motion (which uses the representation in Figure 5.2). Trajectories can be easily observed from the visualized neurons.



(a) Train video sequence containing two scenes



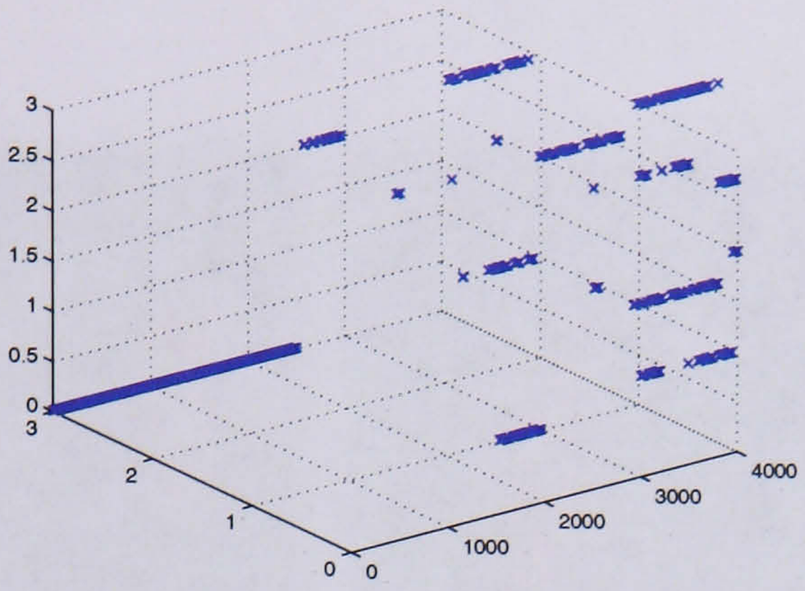
(b) Test video sequence from scene A



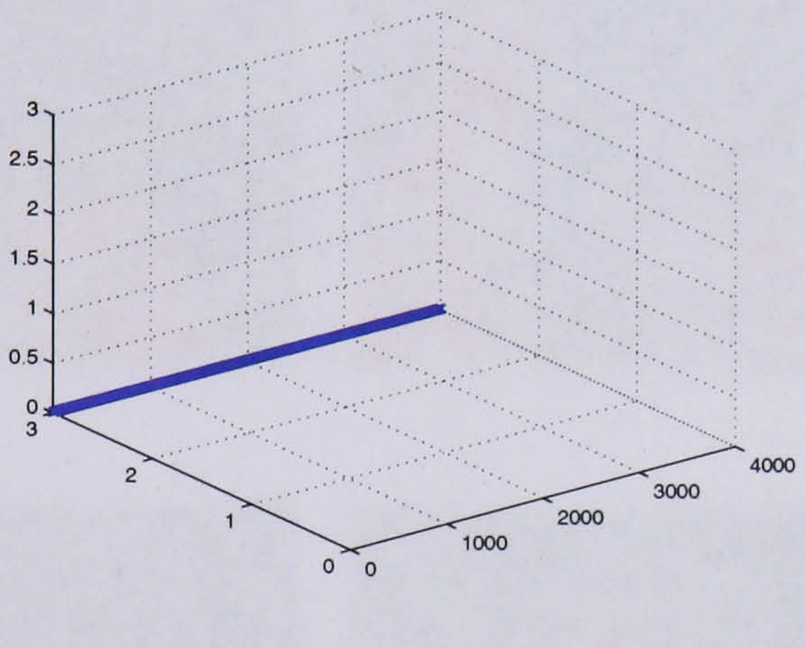
(c) Test video sequence from scene B

Figure 5.8: Tracking of the winning neuron over time: with the right vertical plane as the plane of the indices of the neurons (from $(0,0)$ to $(3,3)$. Different scenes produce different winning neurons.

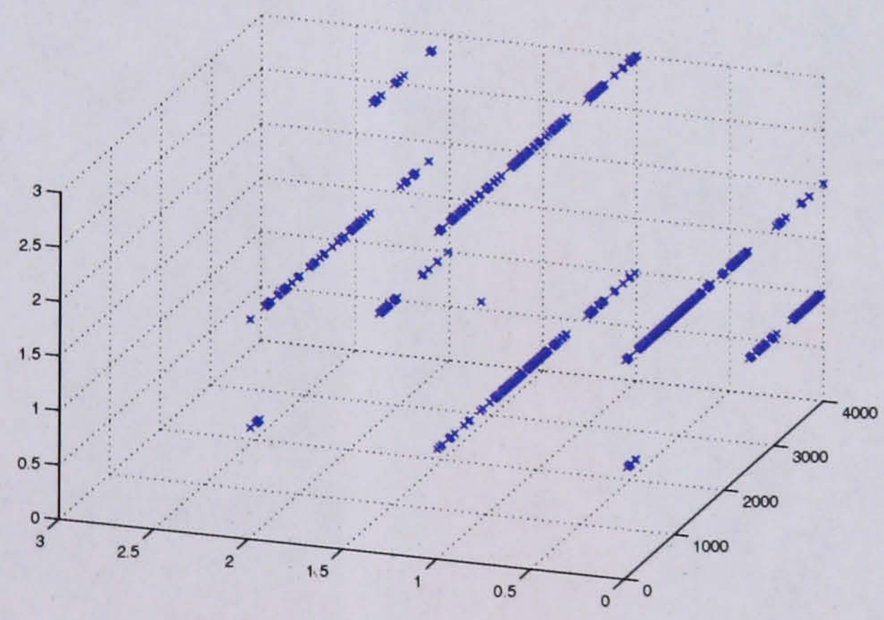
Figure 5.9: Two neurons from the SOM built by a video sequence that contains two crowded scenes: Experiment 2



(a) Train video sequence containing two scenes



(b) Test video sequence from the first scene



(c) Test video sequence from the second scene

Figure 5.10: Tracking of the winning neuron over time: with the right vertical plane as the plane of indices of neurons (from $(0,0)$ to $(3,3)$. Different scenes active different winning neurons. Experiment 2

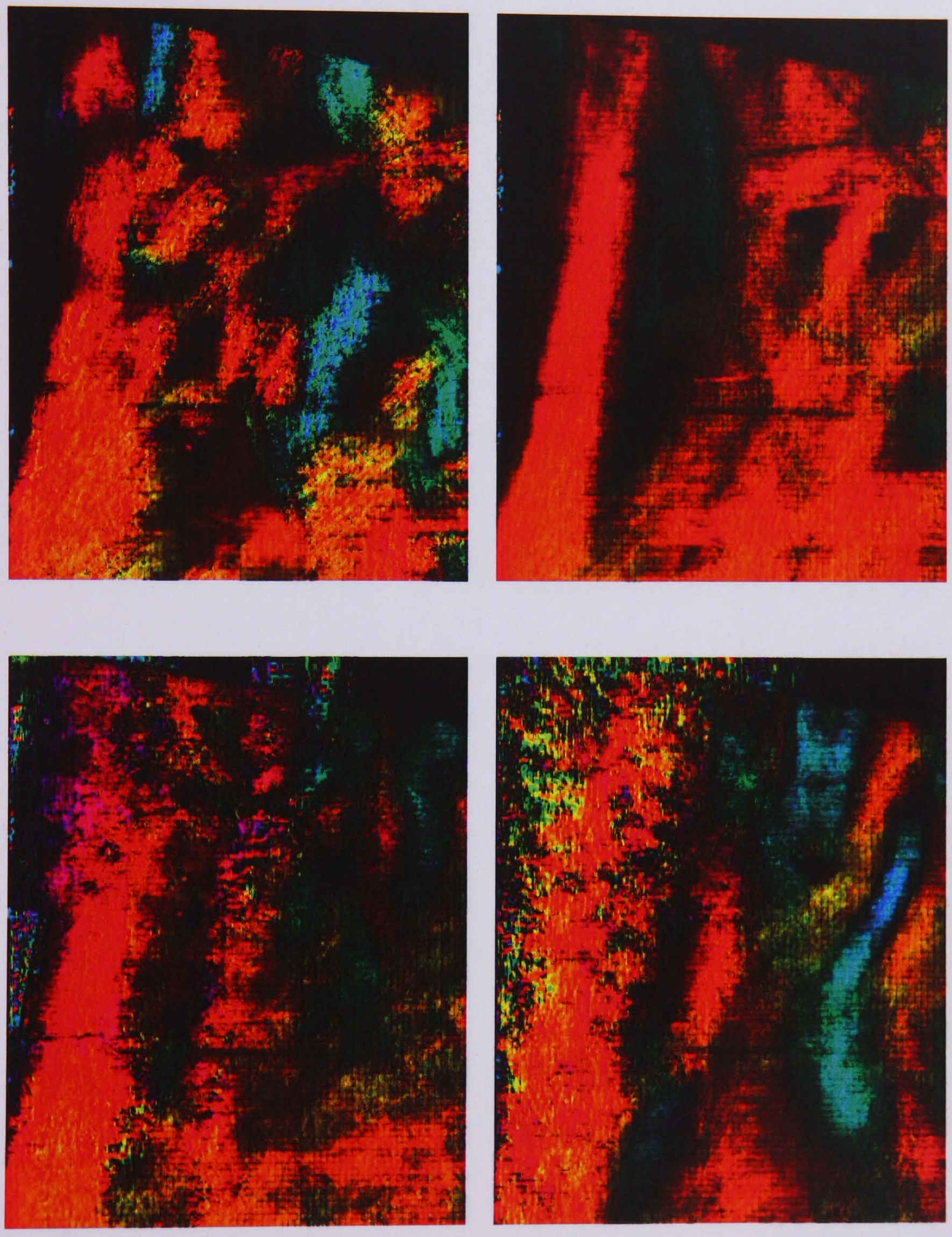


Figure 5.11: Selected SOM neurons from the same Sequence in Fig. 5.5

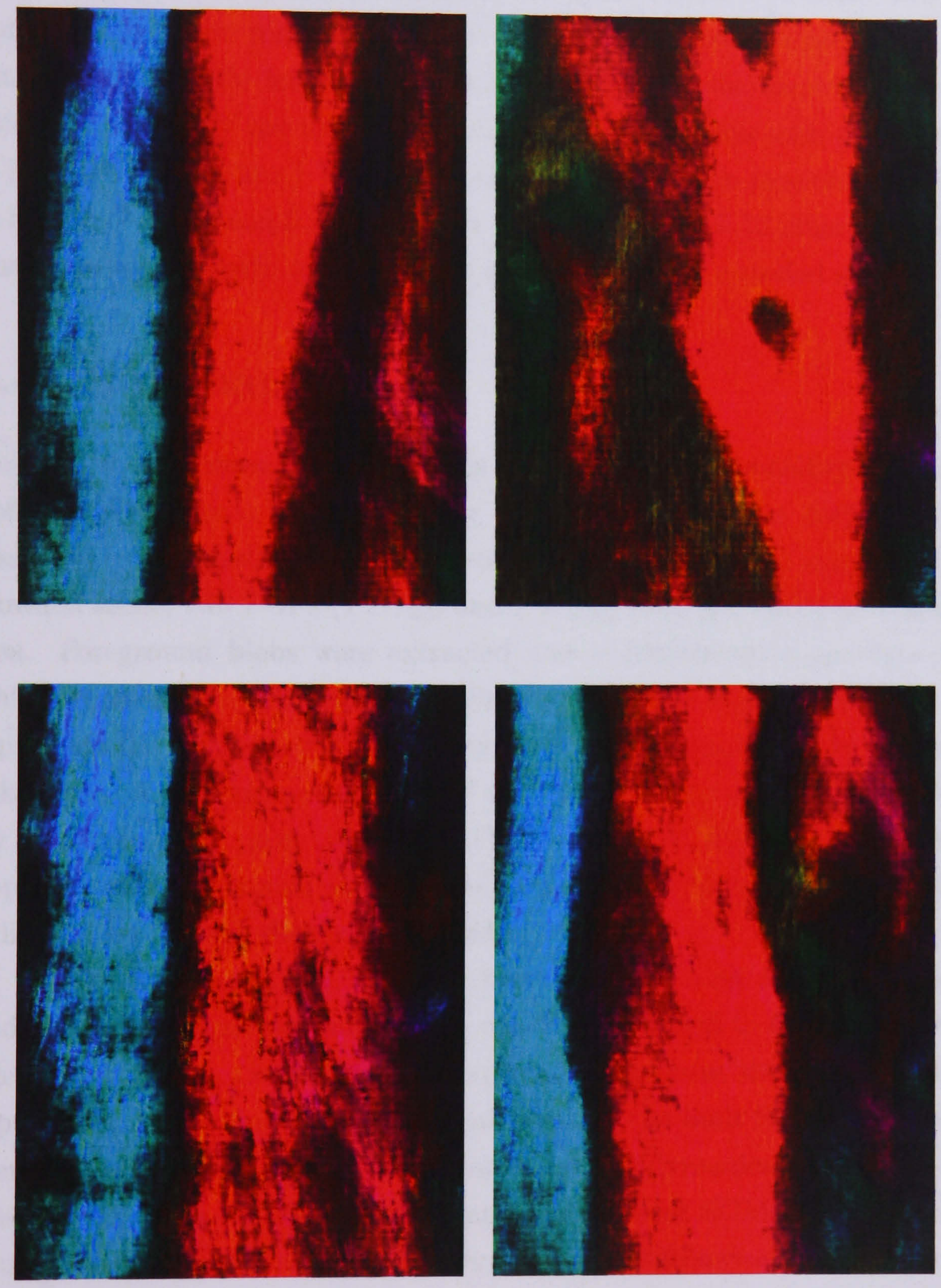


Figure 5.12: Selected SOM neurons from the same Sequence in Fig. 5.6

In the following experiments, again the SOMs are built from video sequences consisting of more than one crowd scene. The experiments use the same sequences as those used in the raw image input experiments. Figure 5.13 shows two neurons from the first video sequence. Figure 5.14(a) shows the changes in the winning neurons on the SOM lattice when using the training video sequence (the coordinates are shown by the left sided vertical plane. The axis with numbers from 0 to 5000 is the time line.) The results of testing the image sequences are included in Figures 5.14(b) and 5.14(c). Figure 5.15 and Figure 5.16 illustrate the results from another test with another video consisting of two crowded scenes.

5.4 Summary

This chapter presented crowd analysis work using the Probability Density Function and Self-organizing Maps. For the first approach, a statistical concept was used to accumulate both the occurrence and the motion information of the crowded scene; two PDFs (PDF_{occ} and PDF_{or}) were generated during this process. Foreground blobs were extracted and accumulated to generate PDF_{occ}, which represents the accumulated probability of occurrence over the scene. PDF_{or} represents the accumulated probability of the orientations that the motion would take place. A path recovering method was developed by calculating the probability along the path using the PDFs. The results show that this work is a simple approach with reasonable results. Compared to the SOM approach, its scalability is limited by its high dimensional results.

For the SOM approach, the experiments were carried out using optical flow and location, raw image frames and a whole motion field to train the SOM. With adequate samples, the SOM were expected to capture the distribution of the input data. In the first case, the visualisation of the built SOM of each crowded scene showed its capability of capturing the major dynamics. Scene classification was carried out by quantitatively comparing the built SOMs. Both values of the neurons and the structure of the SOMs were taken into account in the comparison.

In the latter cases, a SOM can capture major dynamics from more than one scene. Experimental results show that the frames from different scenes activate the neurons from different locations of the lattice so that they can be labelled

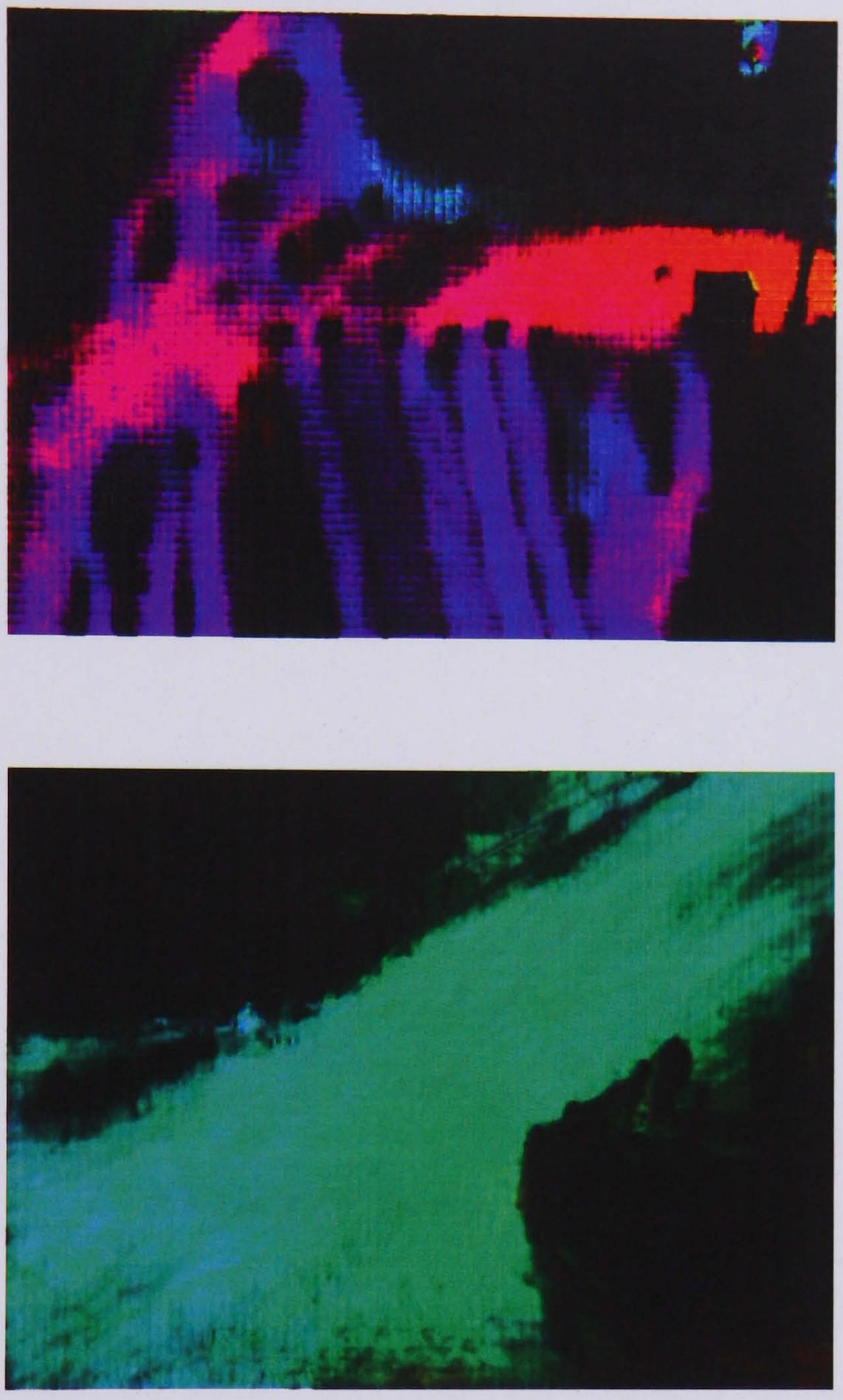
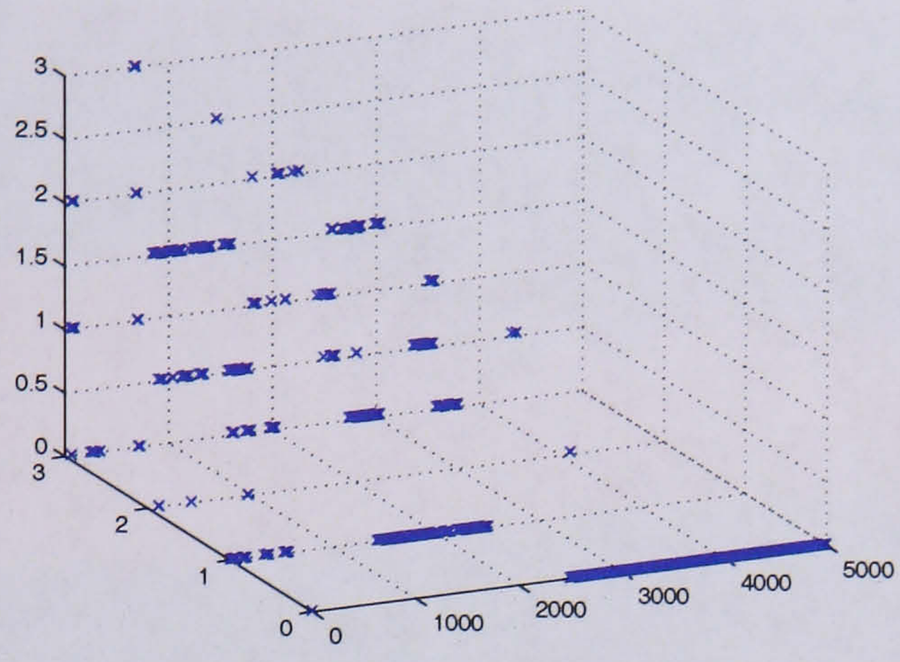


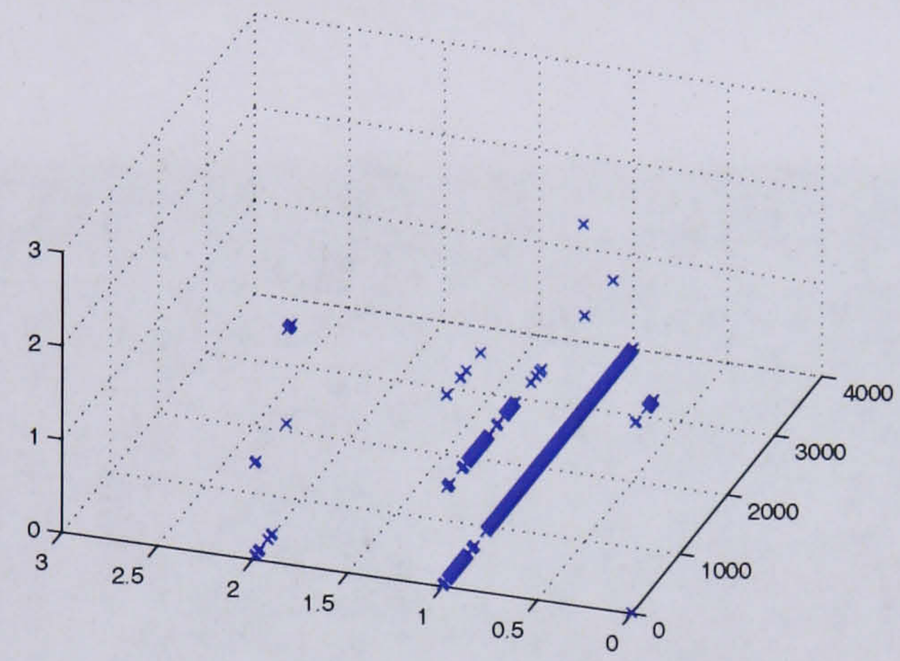
Figure 5.13: Two neurons from the SOM built by the video sequence also used in 5.7

and classified. The tracking of the winning neuron can be a solution for video segmentation. The work on the SOM is the first attempt to employ an SOM in crowd analysis applications. It reveals the great potential of an SOM in handling this problem.

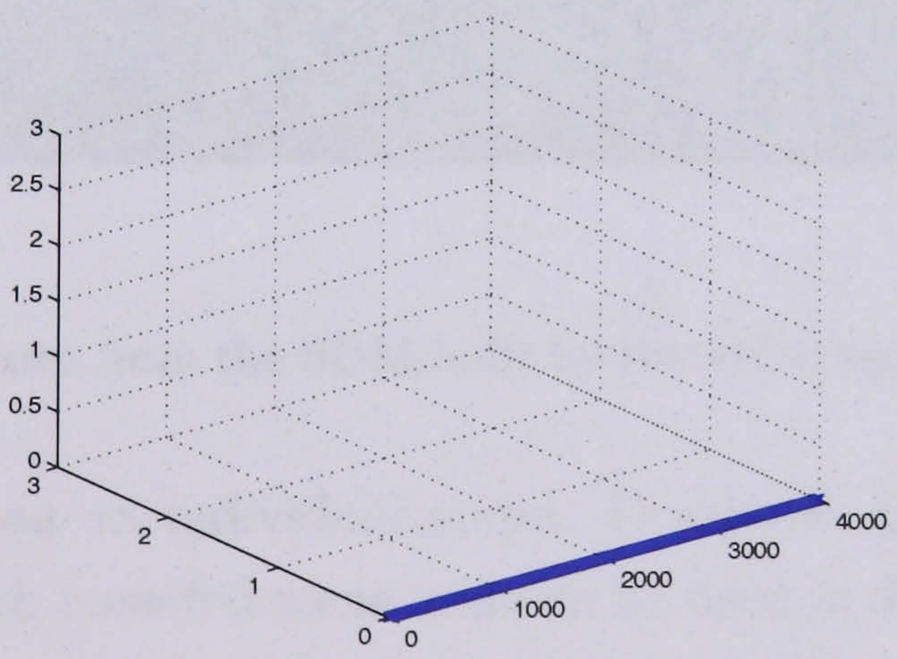
The major contribution of this chapter is the provision of computer vision techniques to learn semantics from crowded scenes. The analysis is based on a



(a) Train video sequence



(b) Test video sequence from scene A

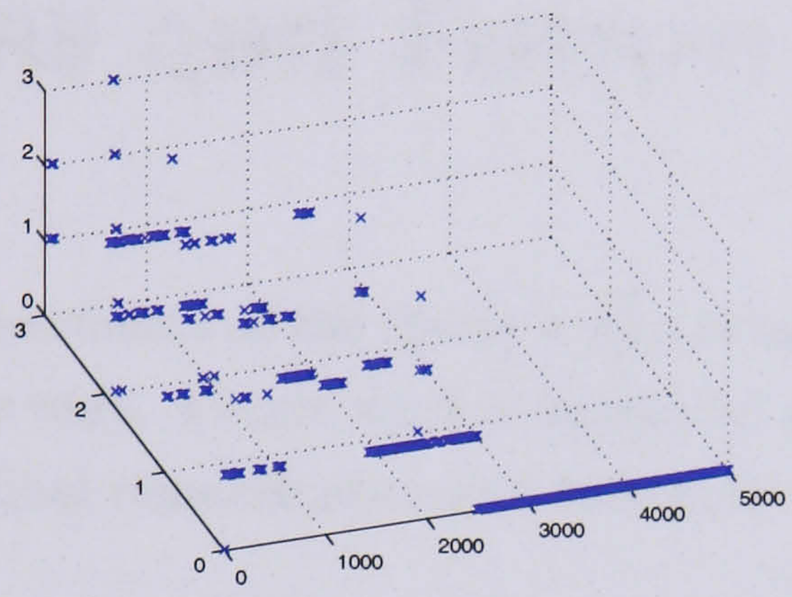


(c) Test video sequence from scene B

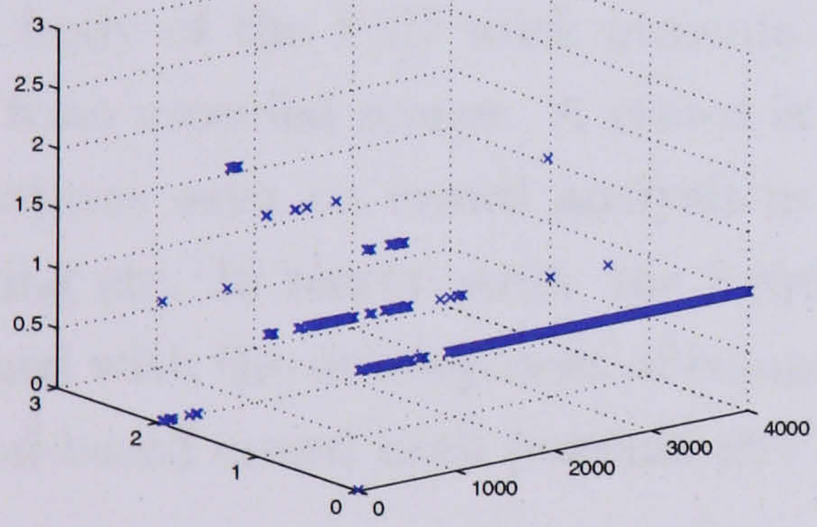
Figure 5.14: Tracking of the winning neuron

Figure 5.15: Two neurons from the SOM built by the video sequence also used in 5.9

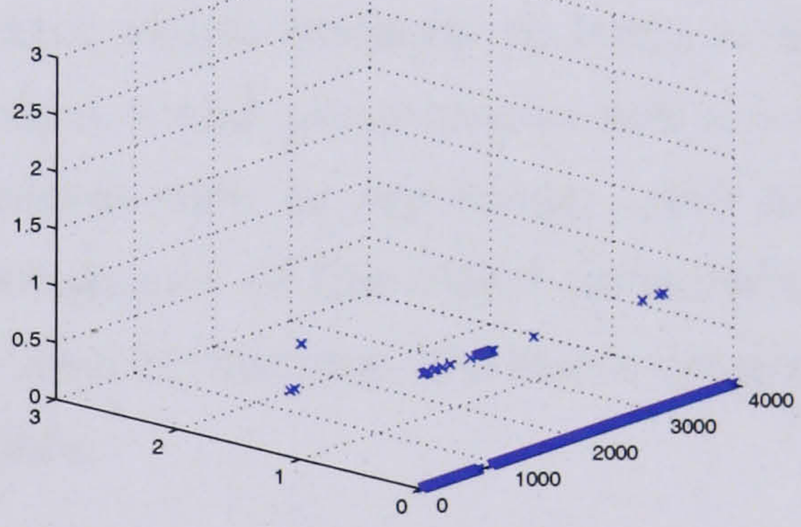
global scope rather than an individual scope. Dominant dynamics is captured and represented for each crowded scene and can be used in different applications. More experiments, for example with different input features, can be carried out. In addition, a deeper analysis of the relationships between neurons can be involved to build a better dynamics model.



(a) Train video sequence



(b) Test video sequence from scene A



(c) Test video sequence from scene B

Figure 5.16: Tracking of the winning neuron

Chapter 6

Conclusions and Future Work

In this chapter, the achievements of the thesis work are summarised, followed by a discussion of the thesis work. Future work is suggested after the discussion. At the end of the chapter, final remarks are made to conclude the whole thesis.

6.1 Achievements

The body of the PhD work presents computer vision methods to learn semantics from crowded scenes. A crowd is a distinct social phenomenon and a lot of literatures exist on crowd analysis in disciplines such as psychology, civil engineering etc. In recent years, the frequent occurrence of the crowd phenomenon, aligned with the development of computer vision techniques, has made computer vision-based crowd both possible and desirable.

State of the art Chapter 2 reviews the state of the art of crowd analysis - the process of crowd analysis is broken down to feature extraction, crowd modelling and event interpretation. The reviewed computer vision techniques on crowd analysis include density measurement, individual detection and tracking. The reviewed traditional work on crowd analysis was derived from areas such as civil engineering and social science that include physical-based, agent-based, cellular-based and nature-based crowd modelling methods. The review also discusses some crowd modelling using HMM and event interpretation work with pre-compiled crowd models from

the computer vision sector that are still at an initial stage. However, the review of the work indicates that there are lots of possibilities of connecting with existing work.

Group behaviour analysis In Chapter 3, a group behaviour analysis work is presented, including methods that extract both micro and macro group information and count people in a group scene. The motion extraction for group analysis is based on colour modelling and tracking. An expectation-maximisation algorithm is used to determine colour models from sample images. A map of the colour Probability Density Function (PDF) can be generated by comparing the image data with the built models. The detected connected components are then used as an input box for a modified CAMSHIFT tracking algorithm. Micro information is retrieved by an individual level dynamics estimator, which is based on the analysis of the curvature of the trajectories and the speed of the individual. The density of the local curvature maxima of each individual trajectory is accumulated and the different behaviours are classified by the resulting density. Macro information is retrieved by a global level dynamics estimator, which employs entropy to estimate the distribution of the colour in the scene, so as to estimate the level of cluttering of the scene.

A simple counting algorithm is used to build a histogram of colour PDF over a horizontal axis. The peaks of the histogram are picked up and each peak is counted as an individual. The algorithm is very easy to implement and the results are fair when the light condition is stable and there are no severe occlusions in the scene. However, when more people and structures are involved in the scene and/or there are more interactions between people, this algorithm is fragile and the results can only be used as a rough estimation. A more accurate counting algorithm is developed based on the assumption that two persons are not likely to stay next to one another for a very long time period. The counting is achieved by monitoring the spatial relations between blobs. The distance between every pair of blobs is calculated for each frame. A temporal distance pyramid is then constructed for each pair of the blobs, and a probabilistic clustering scheme is devised to bound the

blobs in the scene. The blobs that stick to one another in their life time are regarded as individual. The number of the individual is then retrieved by the algorithm.

Crowd motion estimation Although tracking and optical flow have been proposed in many applications to extract dynamics from the video data, a crowded scene offers new challenges for computer vision to measure the motion vectors. Chapter 4 proposed two methods for measuring crowd motion. Both of the algorithms can be explained as matching local descriptors with refined constraints. The first algorithm adapts Harris corners as local descriptors; after an initial matching by the R value, a topological matching is carried out to refine the matching results. The second algorithm makes use of shape information by extracting local descriptors as the local maxima of curvature of detected edges in the image. The refining matching uses the original connections between the local descriptors on the same edge, and compare them in groups as "edgelets". The performance of the two algorithms is compared by testing based on local descriptors and on Motion Connected Components (MCCs). Testing based on local descriptors employs two measures: Mean Similarity (MS) and Mean Absolute Error (MAE), where the second algorithm has a higher MS and lower MAE over all the testing sequences. For testing based on MCCs, the employed measures are Precision and Recall, where the second algorithm has high Recalls over all the testing sequence and lower Precision over two of the testing sequences. Both of the algorithms generated satisfied results while in general, the second one worked better.

Crowd modelling Chapter 5 introduces crowd modelling work. For a group scene where background modelling is still possible, foreground objects are extracted and accumulated to form a Probability Density Function (PDF_{occ}). Meanwhile, the motion of the foreground objects is extracted by block matching, and another Probability Density Function (PDF_{or}) is built. Thus, the two PDFs capture the dynamics of the scene by modelling the foreground occupation of the scene and the foreground motion. With a dedicated path discovery algorithm the main path, as macro information of

the group scene, can be found. In the second half of this chapter, a neural network approach is proposed to capture the main dynamics of a crowd scene. Optical flows are used as a feature in the format of $f : (x, y, \theta, \rho)$. For each pair of frames, the features are generated by the optical flow algorithm, and input into a Self-Organizing Map. The visualisation of the resulting SOMs is developed by placing coloured arrows to represent the last two dimensions of the input space, explained as motion (θ, ρ) over the location (x, y). With the SOMs from optical flow inputs, scene classification is carried out by comparing the resulting SOMs. In addition to the first experiment, raw images and a motion field are used as input features to feed the Self-Organizing Maps (SOMs). Scene classification is carried out based on tracking the winning neurons in the resulting SOMs.

In summary, the achievements of this work include algorithms that are able to extract macro and micro information from a group and extremely crowded scenes, as well as algorithms that model group and crowd dynamics.

6.2 Discussion

A crowd is a complex dynamic; studying crowd dynamics is going to improve the life experiences of humans. Computer vision techniques have the advantage of automatic information extraction, which would highly accelerate the process of building crowd models, and accessing and calibrating these built models. On the other hand, crowd analysis offers big challenges to computer vision techniques. In this thesis, learning about crowd dynamics is achieved by extracting crowd information and modelling crowd dynamics. The methods in this work are based on 2D information from a single static camera without any calibration. As a result, the original methods presented in this thesis could be invalid when there is a serious problem caused by perspective. However, the algorithms are not limited to 2D data and they can be extended to work with 3D information.

A wide variety of research has been combined in this work: image processing techniques, statistical concepts, machine learning, neuron networking and concepts from traditional research such as "Level of service". The work in this thesis

certainly opens doors to more research opportunities. Future research includes employing more crowd analysis reviews from a traditional research area such as civil engineering in computer vision crowd research. In particular, scene geometry effects can be further adapted. For example, trajectory analysis can be used to infer the geometry information of the scene. In addition, the concept of "Level of Service" can be also connected to crowd modelling, and the influence of the number of people retrieved by the counting algorithm to the dynamics can be further investigated.

6.3 Future Work

Based on existing literatures of crowd analysis, the novel approach described in this thesis retrieves semantic information from crowded scene. For group scenes, two dynamics estimators, two people counting algorithms and a modelling algorithm are proposed. For a general crowded situation, motion extraction and dynamics modelling methods are presented. Though the performance of the approach is fairly good, in this section some possible development and improvement is discussed.

For the group behaviour analysis, colour modelling and tracking has been successful in extracting individual behaviour. However, in the general applications, building models for particular colours should be extended to building colour models for randomly selected people. When choosing equipment, a stereo camera or multiple camera view can be adapted to provide 3D information. The behaviour analysis and dynamics estimation can then be projected into a 3D space. The spatial relationships of the blobs used for counting can also be used in analysing the collective behaviour of the group, i.e. the interactions of an individual. A possible extension of the work could be to combine the information about the spatial relationships of the blobs and the curvature of trajectories of the blobs; thus, the analysis of individual and collective behaviour can be connected.

For crowd motion measurement, the second algorithm works better than the first. This is as a result of using shape information in the second algorithm. In the second algorithm, unique length segments of the edge are used as edgelets

for reducing the problem of edge merging and splitting caused by occlusions. However, for the long edges that are not split, the matching of small segments causes a lot of redundant calculations. To solve this problem, pyramid-based matching can also be employed in the second algorithm so that the matching can take place from the coarse scale to the fine scale. For both of the algorithms, instead of matching, a short-period tracking of the local descriptors, which can generate trajectories' fragments, could be useful to the dynamics analysis.

For the statistical approach of group dynamics modelling, the concept of PDF_{occ} could be far beyond the accumulation of foreground objects. All the features that are used in motion extraction, for example the colour blobs and local descriptors, can be used to build PDF_{occ}. Furthermore, 3D $PDFs$ (PDF_{occ} and PDF_{or}) can be built up by 3D information, which can contribute to discovering a major path in 3D space. For the neuron network approach, more investigations about the topological properties need to be carried out. A greater number of different scenes can be input into the SOM to test its ability to classify a scene, and the probable relationship between the different numbers of neurons and scenes can be an interesting topic.

6.4 Final Remarks

The overall aim of this work is to develop crowd analysis methods that are able to automatically learn about crowd dynamics from video data. This thesis proposes algorithms for group behaviour analysis, modelling, crowd motion estimation, and crowd dynamic modelling. This is the first known work on computer vision-based crowd analysis employing traditional crowd analysis work and a neuron network. This work allows a new stage of automatic crowd analysis to be possible, and to be further developed.

References

[1] Otto M.J. Adang and Clifford Stott. A European study of the interaction between police and crowds of foreign nationals considered to pose a risk to public order. http://policestudies.homestead.com/Euro2004.html. 11

[2] ADVISOR. http://advisor.matrasi-tls.fr/. 11

[3] AEA and Techology. A technical summary of the aea egress code. *Technical Report*, **1**, 2002. 28

[4] S. Ali and M. Shah. A Lagrangian Particle Dynamics Approach for Crowd Flow Segmentation and Stability Analysis. *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–6, 2007. 22

[5] S. Ali and M. Shah. Floor Fields for Tracking in High Density Crowd Scenes. 2008. 29

[6] Mitsubishi Electric Research Laboratories Ambient Intelligence for Better Buildings. http://www.merl.com/projects/ulrs/. 32

[7] MIT Ambient Intelligence group. http://ambient.media.mit.edu/. 32

[8] Autonomous University of Madrid Ambient Intelligence Laboratory. http://amilab.ii.uam.es/. 32

[9] NTT Research Ambient Intelligence Research Group. http://www.brl.ntt.co.jp/cs/ai/index.html. 32

[10] Philips Research Ambient Intelligence Research in ExperienceLab. http://www.research.philips.com/technologies/syst_softw/ami/. 32

[11] E. L. Andrade, S. Blunsden, and R. B. Fisher. Performance analysis of event detection models in crowded scenes. In *Proc. Workshop on Towards Robust Visual Surveillance Techniques and Systems at Visual Information Engineering 2006*, pages 427–432, Bangalore, India, 2006. 23

[12] E.L. Andrade and R.B. Fisher. Simulation of crowd problems for computer vision. In *First International Workshop on Crowd Simulation*, **3**, pages 71–80, 2005. 28

[13] E.L. Andrade and R.B. Fisher. Hidden Markov models for optical flow analysis in crowds. In *Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)-Volume 01*, pages 460–463. IEEE Computer Society Washington, DC, USA, 2006. 23

[14] E.L. Andrade and R.B. Fisher. Modelling crowd scenes for event detection. In *Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)-Volume 01*, pages 175–178. IEEE Computer Society Washington, DC, USA, 2006. 23

[15] G. Antonini, M. Bierlaire, and M. Weber. Simulation of pedestrian behaviour using a discrete choice model calibrated on actual motion data. In *4th STRC Swiss Transport Research Conference*, Ascona, 2004. 29

[16] G. Antonini, S. Venegas, and M. Thiran, J.P.and Bierlaire. A discrete choice pedestrian behaviour model in visual tracking systems. In *Advanced Concepts for Intelligent Vision Systems*, pages 273–280, Brussels, Belgium, 2004. 29

[17] A.PAPOULIS. *Probability, Random Variables, and Stochastic Processes.* Electrical and Electronic Engineering Series. McGraw Hill, third edition edition, 1991. 51

[18] S. BANARJEE, C. GROSAN, AND A. ABARHA. Emotional ant based modeling of crowd dynamics. In *Seventh International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC'05)*, pages 279–286, 2005. 28

[19] BEHAVE. http://www.homepages. informatics. ed.ac.uk /rbf /BEHAVE/. 11

[20] SS BLACKMAN. Multiple hypothesis tracking for multiple target tracking. *Aerospace and Electronic Systems Magazine, IEEE*, **19**(1):5–18, 2004. 20

[21] B.A. BOGHOSSIAN AND S.A. VELASTIN. Motion-based machine vision techniques for the management of large crowds. In *the 6th IEEE International Conference on Electronics, Circuits and Systems*, **2**, 1999. 24

[22] J.Y. BOUGUET. Pyramidal implementation of the lucas kanade feature tracker description of the algorithm. *Intel Corporation, Microprocessor Research Labs, OpenCV Documents*, 1999. 106

[23] J.Y. BOUGUET. Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm. *Intel Corporation, Microprocessor Research Labs*, 2000. 70

[24] G.R. BRADSKI. Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal*, **2**:1–15, 1998. 37

[25] M. BRENNER, N. WIJERMANS, T. NUSSLE, AND B. DE BOER. Simulating and controlling civilian crowds in robocup rescue. In *inproceedings of RoboCup 2005: Robot Soccer World Cup IX*, Osaka, 2005. 27

[26] A. BROGGI, M. BERTOZZI, A. FASCIOLI, AND M. SECHI. Shape-based pedestrian detection. In *inproceedings of the. IEEE Intelligent Vehicles Symposium 2000*, Dearbon (MI), USA, 2000. 16

[27] G.J. Brostow and R. Cipolla. Unsupervised Bayesian Detection of Independent Motion in Crowds. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Volume 1*, pages 594–601. IEEE Computer Society Washington, DC, USA, 2006. 17

[28] Yizheng Cai, Nando de Freitas, and James J. Little. Robust visual tracking for multiple targets. In *European Conference on Computer Vision*, **3954** of *LNCS*, pages 107–118. Springer, 2006. 19

[29] Michael T. Chan, Anthony Hoogs, Rahul Bhotika, Amitha Perera, John Schmiederer, and Gianfranco Doretto. Joint recognition of complex events and track matching. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1615–1622, Washington, DC, USA, 2006. IEEE Computer Society. 24

[30] T.H. Chang, S. Gong, and E.J. Ong. Tracking multiple people under occlusion using multiple cameras. In *British Machine Vision Conference*, pages 566–575, 2000. 22

[31] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**(8):790–799, 1995. 37

[32] J. Chu, J. Li, M. Xu, and L. Zhao. Simulating escape panic based on the mechanism of asymmetric information distribution. In *Complex Systems Summer School Final Project Papers*, Santa Fe, NM, 2005. Santa Fe Institute. 26

[33] C.M.Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006. 36

[34] Crowd and Dynamics. http://www.crowddynamics.com/. 27

[35] Crowd and MAGS. http://www2.ift.ulaval.ca/ muscamags/Dnd - crowdmags - project.htm. 11

[36] F. CUPILLARD, F. BREMOND, AND M. THONNAT. Behaviour recognition for individuals, groups of people and crowd. *IEE Seminar Digests*, **7**, 2003. 24

[37] F. CUPILLARD, F. BREMOND, M. THONNAT, AND F. INRIA. Group behavior recognition with multiple cameras. *Applications of Computer Vision, 2002.(WACV 2002). Proceedings. Sixth IEEE Workshop on*, pages 177–183, 2002. 24, 31, 36

[38] A.C. DAVIES, J.H. YIN, AND S.A. VELASTIN. Crowd monitoring using image processing. *Electronics & Communication Engineering Journal*, **7**(1):37–47, 1995. 11, 24

[39] JESSE DAVIS AND MARK GOADRICH. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006. 64

[40] LAN DONG, VASU PARAMESWARAN, VISVANATHAN RAMESH, AND IMAD ZOGHLAMI. Fast Crowd Segmentation Using Shape Indexing. Rio de Janeiro, Brazil, 2007. 13

[41] A. DOUCET, S. GODSILL, AND C. ANDRIEU. On sequential Monte Carlo sampling methods for Bayesian filtering, 2000. 19

[42] AHMED ELGAMMAL AND LARRY DAVIS. Probabilistic framework for segmenting people under occlusion. In *Eighth IEEE International Conference on Computer Vision, 2001. ICCV 2001. Proceedings.*, **2**, pages 145–152, 2001. 16

[43] J. FAN, H. LUO, AND AK ELMAGARMID. Concept-oriented indexing of video databases: toward semantic sensitive retrieval and browsing. *IEEE Trans Image Process*, **13**(7):974–92, 2004. 31, 36

[44] FHWA. Traffic analysis tools primer,traffic analysis toolbox (1), 2004. http://ops.fhwa.dot.gov /trafficanalysistools /tat-vol1/index. 25

[45] J. FRUIN. The causes and prevention of crowd disasters. *Engineering for crowd safety. Elsevier, New York*, 1993. 2

[46] K. Fukunaga and L.D. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, **21**:32–40, 1975. 37

[47] P. Gabriel, J.B. Hayet, J. Piater, and J. Verly. Object tracking using color interest points. *Proceedings. IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 159–164, 2005. 74

[48] P. Gabriel, J. Verly, J. Piater, and A. Genon. The state of the art in multiple object tracking under occlusion in video sequences. *Advanced Concepts for Intelligent Vision Systems*, pages 166–173, 2003. 18

[49] V. Gouet and N. Boujemaa. About optimal use of color points of interest for content-based image retrieval. *Technical Report*, pages RP–4439, 2002. 72

[50] M. Han, W. Xu, H. Tao, and Y. Gong. An algorithm for multiple object trajectory tracking. *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, **1**, 2004. 20

[51] S. Haykin. *Neural Networks: A Comprehensive Foundation.* Prentice Hall PTR Upper Saddle River, NJ, USA, 1994. 105

[52] B. Heisele and C. Woehler. Motion-based recognition of pedestrians. *Fourteenth International Conference on Pattern Recognition, 1998. Proceedings.*, **2**:1325–1330, 1998. 16

[53] D. Helbing, I. Farkas, and T. Vicsek. Simulating Dynamical Features of Escape Panic. *Letters to Nature*, **407**:487–490, 2000. 26

[54] D. Helbing and P. Molnár. Social force model for pedestrian dynamics. *Physical Review E*, **51**(5):4282–4286, 1995. 25

[55] Dirk Helbing. Models for pedestrian behavior. 1992. 25

[56] Dirk Helbing and Peter Molnar. Self-organization phenomena in pedestrian crowds. 1997. 26

[57] LF HENDERSON. The statistics of crowd fluids. *Nature*, **229**(5284):381–383, 1971. 1

[58] LF HENDERSON. On the fluid mechanics of human crowd motion. *Transportation Research*, 8:509–515, 1974. 1

[59] B.K.P. HORN AND B.G. SCHUNCK. Determining Optical Flow. *Artificial Intelligence*, **17**(1-3):185–203, 1981. 70

[60] MIN HU, SAAD ALI, AND MUBARAK SHAH. Detecting Global Motion Patterns in Complex Videos. In *9th International Conference on Pattern Recognition*, Tampa, 2008. 22

[61] MIN HU, SAAD ALI, AND MUBARAK SHAH. Learning Motion Patterns in Crowded Scenes Using Motion Flow Field. In *9th International Conference on Pattern Recognition*, Tampa, 2008. 23

[62] W. HU, T. TAN, L. WANG, AND S. MAYBANK. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, **34**(3):334–352, 2004. 12

[63] W. HU, D. XIE, Z. FU, W. ZENG, AND S. MAYBANK. Semantic-Based Surveillance Video Retrieval. *Image Processing, IEEE Transactions on*, **16**(4):1168–1181, 2007. 31, 36

[64] CHANG HUANG, HAIZHOU AI, YUAN LI, AND SHIHONG LAO. Vector boosting for rotation invariant multi-view face detection. In *Tenth IEEE International Conference on Computer Vision*, **1**, pages 446–453, 2005. 15

[65] X. HUANG, L. LI, AND T. SIM. Stereo-based human head detection from crowd scenes. *International Conference on Image Processing, 2004. ICIP'04.*, **2**:1353–1356, 2004. 15

[66] RL HUGHES. A continuum theory for the flow of pedestrians. *Transportation Research Part B: Methodological*, **36**(6):507–535, 2002. 26

[67] INRIA. http://www.inria.fr /rapportsactivite /RA2005 /orion /uid1.html. 11

[68] Department of Computer Science Intelligent Inhabited Environments Group. http://iieg.essex.ac.uk/idorm.htm. 32

[69] Michael Isard and Andrew Blake. A mixed-state CONDENSATION tracker with automatic model-switching. In *IEEE International Conference on Computer Vision*, pages 107–112, 1998. url: citeseer.ist.psu.edu/isard98mixedstate.html. 19

[70] ISCAPS. http://www.iscaps.reading.ac.uk/home.htm. 11

[71] M. Jones and P. Viola. Fast multi-view face detection. *Mitsubishi Electric Research Lab TR-20003-96*, 2003. 15

[72] I.K. Jung and S. Lacroix. A robust interest points matching algorithm. *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, **2**, 2001. 74

[73] H. Kang, D. Kim, and S.Y. Bang. Real-time multiple people tracking using competitive condensation. *Proc. of the Intl. Conference on Pattern Recognition*, **1**:413–416, 2002. 19

[74] R. Karlsson and F. Gustafsson. Monte Carlo data association for multiple target tracking. *Target Tracking: Algorithms and Applications (Ref. No. 2001/174), IEE*, **1**, 2001. 20

[75] Saad M. Khan and Mubarak Shah. A multiview approach to tracking people in crowded scenes using a planar homography constraint. In *9th European Conference on Computer Vision*, **3954** of *LNCS*, pages 133–146. Springer, 2006. 22

[76] Z. Khan, T. Balch, and F. Dellaert. MCMC-based particle filtering for tracking a variable number of interacting targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**(11):1805–1819, 2005. 20

[77] KYUNGNAM KIM AND LARRY S. DAVIS. Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering. In *European Conference on Computer Vision*, **3953** of *LNCS*, pages 98–109. Springer, 2006. 22

[78] A. KIRCHNER AND A. SCHADSCHNEIDER. Simulation of evacuation processes using a bionics-inspired cellular automaton model for pedestrian dynamics. *Physica A: Statistical Mechanics and its Applications*, **312**(1-2):260–276, 2002. 28

[79] J.A. KIRKLAND AND A.A. MACIEJEWSKI. A simulation of attempts to influence crowd dynamics. *IEEE International Conference on Systems, Man, and Cybernetics*, pages 4328–4333, 2003. 26

[80] T. KIRT, E. VAINIK, AND L. VÕHANDU. A method for comparing self-organizing maps: case studies of banking and linguistic data. In *Eleventh East-European Conference on Advances in Databases and Information Systems ADBIS*, page 107C115, Varna, Bulgaria: Technical University of Varna, 2007. 98, 105

[81] E.B. KOLLER-MEIER AND F. ADE. Tracking multiple objects using the Condensation algorithm. *Robotics and Autonomous Systems*, **34**(2-3):93–105, 2001. 19

[82] D. KONG, D. GRAY, AND H. TAO. Counting Pedestrians in Crowds Using Viewpoint Invariant Training. *British Machine Vision Conference*, 2005. 14

[83] D. KONG, D. GRAY, AND H. TAO. A viewpoint invariant approach for crowd counting. *Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)-Volume 03*, pages 1187–1190, 2006. 14

[84] TOBIAS KRETZ AND MICHAEL SCHRECKENBERG. F.a.s.t. - floor field- and agent-based simulation tool, 2006. 28

[85] G. LE BON. La Psychologie des Foules (The Crowd: A Study of the Popular Mind), 1895. 1

[86] G. LEFEBVRE, C. LAURENT, J. ROS, AND C. GARCIA. Supervised Image Classification by SOM Activity Map Comparison. *Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)-Volume 02*, pages 728–731, 2006. 98, 105

[87] LEGION. http://www.legion.biz/about /index.html. 3, 27, 69

[88] B. LEIBE, E. SEEMANN, AND B. SCHIELE. Pedestrian detection in crowded scenes. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005.*, **1**, 2005. 16

[89] STAN Z. LI, LONG ZHU, ZHENQIU ZHANG, ANDREW BLAKE, HONGJIANG ZHANG, AND HARRY SHUM. Statistical learning of multi-view face detection. In *Proceedings of the 7th European Conference on Computer Vision-Part IV*, pages 67–81, London, UK, 2002. Springer-Verlag. 15

[90] S.F. LIN, J.Y. CHEN, AND H.X. CHAO. Estimation of number of people in crowded scenes using perspective transformation. *IEEE Transactions on Systems, Man and Cybernetics, Part A,*, **31**(6):645–654, 2001. 13

[91] B.D. LUCAS AND T. KANADE. An iterative image registration technique with an application to stereo vision. *International Joint Conference on Artificial Intelligence*, **81**:674–679, 1981. 70

[92] R. MA, L. LI, W. HUANG, AND Q. TIAN. On pixel count based crowd density estimation for visual surveillance. *IEEE Conference on Cybernetics and Intelligent Systems*, **1**, 2004. 13

[93] C. MACKAY. Extraordinary Popular Delusions and Madness of Crowds 2 vols. *London: National Standard Library*, 1852. 1

[94] A. MARANA, L. DA COSTA, R. LOTUFO, AND S. VELASTIN. On the Efficacy of Texture Analysis for Crowd Monitoring. *Proceedings of the International Symposium on Computer Graphics, Image Processing, and Vision-Volume 00*, page 354, 1998. 14

[95] A.N. MARANA, L. DA FONTOURA COSTA, R.A. LOTUFO, AND S.A. VELASTIN. Estimating crowd density with Minkowski fractal dimension. *IEEE International Conference on Acoustics, Speech, and Signal Processing, 1999. ICASSP'99. Proceedings.*,, **6**:3521–3524, 1999. 13

[96] A.N. MARANA, S.A. VELASTIN, L.F. COSTA, AND R.A. LOTUFO. Estimation of crowd density using image processing. *IEE Colloquium on Image Processing for Security Applications (Digest No: 1997/074)*,, page 11, 1997. 13

[97] A.N. MARANA, S.A. VELASTIN, L.F. COSTA, AND R.A. LOTUFO. Automatic estimation of crowd density using texture. *Safety Science*, **28**(3):165–175, 1998. 13

[98] J.S. MARQUES, P.M. JORGE, A.J. ABRANTES, AND JM LEMOS. Tracking Groups of Pedestrians in Video Sequences. *IEEE 2003 Conference on Computer Vision and Pattern Recognition Workshop*, **9**:101, 2003. 21

[99] T. MATHES AND J. PIATER. Robust non-rigid object tracking using point distribution models. *Proc. of British Machine Vision Conference (BMVC)*, **2**, 2005. 18

[100] B. MAURIN, O. MASOUD, AND N. PAPANIKOLOPOULOS. Monitoring crowded traffic scenes. *The IEEE 5th International Conference on Intelligent Transportation Systems, 2002. Proceedings.*, pages 19–24, 2002. 24

[101] S.J. MCKENNA, S. JABRI, Z. DURIC, A. ROSENFELD, AND H. WECHSLER. Tracking groups of people. *Computer Vision and Image Understanding*, **80**(1):42–56, 2000. 21

[102] A. MITTAL AND L.S. DAVIS. M 2 Tracker: A Multi-View Approach to Segmenting and Tracking People in a Cluttered Scene. *International Journal of Computer Vision*, **51**(3):189–203, 2003. 21

[103] F. MOKHTARIAN, S. ABBASI, AND J. KITTLER. Robust and efficient shape indexing through curvature scale space. *Proc. British Machine Vision Conference*, **62**, 1996. 79

[104] SR MUSSE AND D. THALMANN. A Model of Human Crowd Behavior: Group Inter-Relationship and Collision Detection Analysis. *Proc. Workshop of Computer Animation and Simulation of Eurographics*, **97**:39–51, 1997. 26

[105] K. OKUMA, A. TALEGHANI, N. DE FREITAS, J.J. LITTLE, AND D.G. LOWE. A boosted particle filter: Multitarget detection and tracking. *European Conference on Computer Vision*, **1**:28–39, 2004. 19

[106] X. PAN, C.S. HAN, K. DAUBER, AND K.H. LAW. Human and social behavior in computational modeling and analysis of egress. *Automation in Construction*, **15**(4):448–461, 2006. 26

[107] DANIEL POLANI. Measures for the organization of self-organizing maps. *Self-Organizing neural networks: recent advances and applications*, pages 13–44, 2002. 108

[108] A. POLUS, J.L. SCHOFER, AND A. USHPIZ. Pedestrian Flow and Level of Service. *Journal of Transportation Engineering*, **109**(1):46–56, 1983. 13

[109] PRISMATICA. http://prismatica.king.ac.uk/. 11

[110] EC FUNDED CAVIAR PROJECT. http://homepages.inf.ed.ac.uk/rbf/CAVIAR. 81

[111] H. RAHMALAN, MS NIXON, AND JN CARTER. On Crowd Density Estimation for Surveillance. *The Institution of Engineering and Technology Conference on Crime and Security*, pages 540C–545, 2006. 13

[112] A. RAKOTONIRAINY AND R. TAY. In-vehicle ambient intelligent transport systems (I-VAITS): towards an integrated research. *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*, pages 648–651, 2004. 32

[113] C. RASMUSSEN AND GD HAGER. Joint probabilistic techniques for tracking multi-part objects. *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 16–21, 1998. 20

[114] D. Reid. An algorithm for tracking multiple targets. *Automatic Control, IEEE Transactions on*, **24**(6):843–854, 1979. 20

[115] P. Reisman, O. Mano, S. Avidan, A. Shashua, M.E.V.T. Ltd, and I. Jerusalem. Crowd detection in video sequences. *Intelligent Vehicles Symposium, 2004 IEEE*, pages 66–71, 2004. 17

[116] P. Remagnino, G.L. Foresti, and T. Ellis. *Ambient Intelligence: A Novel Paradigm.* Springer, 2005. 32

[117] Clements R.R. and R.L. Hughes. Mathematical modelling of a mediaeval battle: the battle of agincourt. *Mathematics and Computers in Simulation*, **64(2)**:259–269, 2004. 26

[118] S.Gong, S.J.McKenna, and A.Psarrou. *Dynamic vision: from images to face recognition.* Imperial College Press, 2000. 37

[119] A. Shashua, Y. Gdalyahu, and G. Hayun. Pedestrian detection for driving assistance systems: single-frame classification and system level performance. *Intelligent Vehicles Symposium, 2004 IEEE*, pages 1–6, 2004. 16

[120] H. Sidenbladh and S.L. Wirkander. Tracking random sets of vehicles in terrain. *Proc. 2003 IEEE Workshop on Multi-Object Tracking*, **9**:98, 2003. 19

[121] N.T. Siebel and S. Maybank. Fusion of multiple tracking algorithms for robust people tracking. *European Conference on Computer Vision*, pages 373–387, 2002. 20

[122] K. Smith, D. Gatica-Perez, and J.M. Odobez. Using particles to track varying numbers of interacting people. *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Volume 1-Volume 01*, pages 962–969, 2005. 20

[123] M. Sonka, V. Hlavac, and R. Boyle. Image Processing, Analysis, and Machine Vision, 1998. Technical report, ISBN 0-534-95393-X, 1998. 99

[124] M. SPENGLER AND B. SCHIELE. Towards robust multi-cue integration for visual tracking. *Machine Vision and Applications*, **14**(1):50–58, 2003. 20

[125] C. STANGOR. *Social Groups in Action and Interaction.* Psychology Press (UK), 2004. 2

[126] C. STAUFFER AND W.E.L. GRIMSON. Adaptive background mixture models for real-time tracking. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, **2**, pages 246–252, 1999. 70

[127] C. STAUFFER AND WEL GRIMSON. Adaptive background mixture models for real-time tracking. *1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1999. 99

[128] G.KEITH STILL. Crowd dynamics. *PHD thesis*, 2000. 2

[129] S. R. SUBRAMANYA, HIRAL PATEL, AND ILKER ERSOY. Performance evaluation of block-based motion estimation algorithms and distortion measures. In *ITCC '04: Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04) Volume 2*, page 2, Washington, DC, USA, 2004. IEEE Computer Society. 85

[130] JOSEPHINE SULLIVAN AND STEFAN CARLSSON. Tracking and labelling of interacting multiple targets. In *European Conference on Computer Vision*, **3953** of *LNCS*, pages 619–632. Springer, 2006. 21

[131] R.S. SUTTON AND A.G. BARTO. *Reinforcement Learning: An Introduction.* MIT Press, 1998. 100

[132] D. SWETS AND B. PUNCH. Genetic algorithms for object localization in a complex scene. *IEEE International Conference on Image Processing*, pages 595–598, 1995. 15

[133] JA SWETS. Measuring the accuracy of diagnostic systems. *Science*, **240**(4857):1285–1293, 1988. 75

[134] KINGSTON UNIVERSITY THE AMBIENT INTELLIGENCE RESEARCH GROUP. http://www.kingston.ac.uk/ambient-intelligence/. 32

[135] T.M.COVER AND J.A.THOMAS. *Elements of Information Theory.* Wiley Series in Telecommunications, 1991. 51

[136] STANFORD UNIVERSITY. CIFE Seed Project 2004-2005, 2005-2006, http://eil.stanford.edu/egress/. 11

[137] U.S. CENSUS BUREAU. http://www.census.gov/ ipc /www /world-pop.html. 9

[138] L. VAN VLIET, I. YOUNG, AND P. BEEK. Recursive gaussian derivative filters. In *In Proc. l4th International Conference on Pattern Recognition (ICPR'98), volume 1, IEEE Computer Society Press, Aug. 1998.*, pages 509–514, 1998. 72

[139] S.A. VELASTIN, J.H. YIN, A.C. DAVIES, M.A. VICENCIO-SILVA, R.E. ALLSOP, AND A. PENN. Automated measurement of crowd density and motion using imageprocessing. *Road Traffic Monitoring and Control, 1994., Seventh International Conference on*, pages 127–132, 1994. 11, 14

[140] S. VENEGAS, S.F. KNEBEL, AND J.P. THIRAN. Multi-object tracking using particle filter algorithm on the top-view plan. *Technical report,LTS-REPORT-2004-003, EPFL*, 2004. http :// infoscience.epfl.ch / getfile.py? mode= best & recid=87041. 19

[141] VT VU, F. BREMOND, AND M. THONNAT. Human Behaviour Visualisation and Simulation for Automatic Video Understanding. *Proc. of the 10th Int. Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG–2002), Plzen–Bory, Czech Republic*, pages 485–492, 2002. 28

[142] B. WU AND R. NEVATIA. Detection of Multiple, Partially Occluded Humans in a Single Image by Bayesian Combination of Edgelet Part Detectors.

Tenth IEEE International Conference on Computer Vision, 2005. ICCV 2005, **1**:90–97, 2005. 16, 19

[143] Bo Wu and Ram Nevatia. Tracking of multiple, partially occluded humans based on static body part detection. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, **1**, pages 951–958, 2006. 19

[144] Tao Xiang and Shaogang Gong. Video Behavior Profiling for Anomaly Detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **30**:893 – 908, 2007. 31, 36

[145] D.B. Yang, H.H. Gonzalez-Banos, and L.J. Guibas. Counting people in crowds with a real-time network of simple image sensors. *Ninth IEEE International Conference on Computer Vision, 2003. Proceedings.*, pages 122–129, 2003. 14

[146] J.H. Yin, S.A. Velastin, and A.C. Davies. Image Processing Techniques for Crowd Density Estimation Using a Reference Image. *Proc. 2nd Asia-Pacific Conf. Comput. Vision*, **3**:6–10, 1995. 11, 13

[147] T. Zhao and R. Nevatia. Tracking multiple humans in complex situations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **26**(9):1208–1221, 2004. 19

[148] T. Zhao and R. Nevatia. Tracking multiple humans in crowded environment. *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, **2**:II–406–II–413, 2004. 19

Appendix A

Appendix: Publications

Machine Vision and Applications manuscript No.
(will be inserted by the editor)

B. Zhan, D. N. Monekosso, P. Remagnino, S. A. Velastin · L-Q. Xu

Crowd Analysis: A Survey

Abstract in the year 1999 the world population reached 6 billion, doubling the previous census estimate of 1960. Recently, the United States Census Bureau issued a revised forecast for world population showing a projected growth to 9.4 billion by 2050 [88]. Different research disciplines have studied the crowd phenomenon and its dynamics from a social, psychological and computational standpoint respectively. This paper presents a survey on crowd analysis methods employed in computer vision research and discusses perspectives from other research disciplines and how they can contribute to the computer vision approach.

Keywords crowd studies · crowd dynamics · sociodynamics · crowd simulations · computer vision.

1 Introduction

The steady population growth, along with the worldwide urbanization, has made the crowd phenomenon more frequent. It is not surprising, therefore, that crowd analysis has received attention from technical and social research disciplines. The crowd phenomenon is of great interest in a large number of applications:

Crowd Management: Crowd analysis can be used for developing crowd management strategies, especially for increasingly more frquent and popular events such as sport matches, large concerts, public demonstrations and so on, to avoid crowd related disasters and insure public safety.

Public Space Design: Crowd analysis can provide guidelines for the design of public spaces, e.g. to make the layout of shopping malls more convenient to costumers or to optimize the space usage of an office.

Virtual Environments: Mathematical models of crowds can be employed in virtual environments to enhance the simulation of crowd phenomena, to enrich the human life experience.

Visual Surveillance: Crowd analysis can be used for automatic detection of anomalies and alarms. Furthermore, the ability to track individuals in a crowd could help the police to catch suspects.

Intelligent Environments: In some intelligent environments which involve large groups of people, crowd analysis is a pre-requisite for assisting the crowd or an individual in the crowd. For example, in a museum deciding how to divert the crowd based on to the patterns of crowd.

Crowd management and public space design are studied by sociologists, psychologists and civil engineers; virtual environments are studied by computer graphic researchers; visual surveillance and intelligent environments are of interest to computer vision researchers. The approach favored by psychology, sociology, civil engineer and computer graphic research is an approach based on human observation and analysis. Sociologists, for instance, study the characters of a crowd as a social phenomenon, exploring human factors. For example, the computational model developed by Seed Projects at Stanford University[1], incorporated human behavior in environments with emergency exits. The Crowd - MAGS Project, whiche is funded by GEOIDE and the Canadian Network of Centers of Excellence in Geomatics, aims to develop micro-simulations of crowd behaviours and the impact of police or military groups [23]. The Police Academy of the Netherlands and School of Psychology of University of Liverpool are cooperating on a project funded by the UK Home Office: "A European study of the interaction between police and crowds of foreign nationals considered to pose a risk to public order" [2].

On the other hand, computational methods such as those employed in computer graphics and vision methods focus on extracting quantitative features and detecting events in crowds, synthesizing the pehnomenon with mathematical and statistical models. For example, early

B. Zhan, D. N. Monekosso, P. Remagnino, S. A. Velastin
Digital Imaging Research Centre, Kingston University, UK
Tel.: +44 (0)20 8547 7930 Fax: +44 (0)20 8547 7824
E-mail: p.remagnino@kingston.ac.uk

L-Q. Xu
Research and Venturing, BT Group PLC , UK

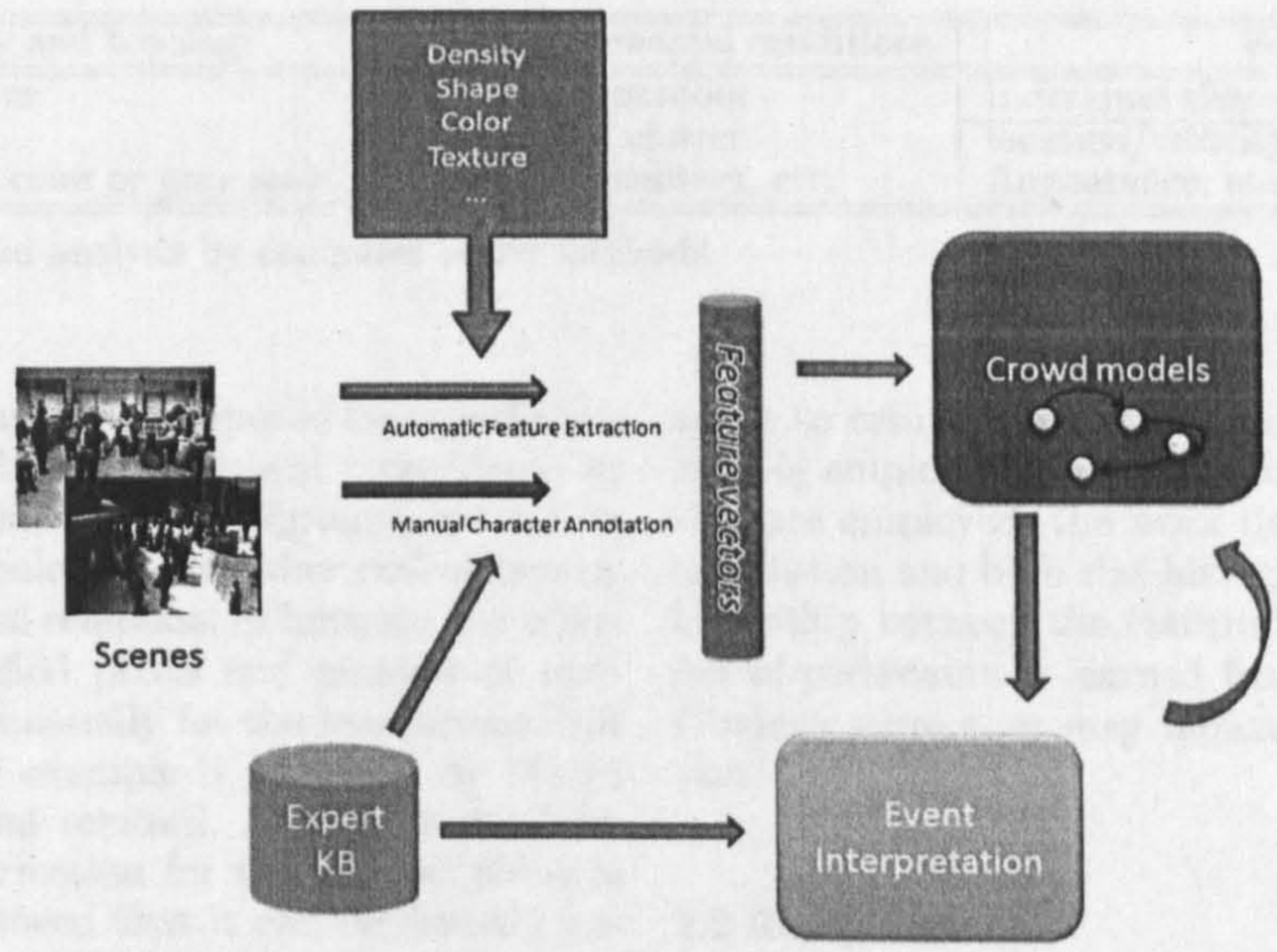


Fig. 1 A framework for Crowd analysis.

project funded by the EPSRC in the UK were concerned with measuring crowd motion and density and hence potentially dangerous situations[26][89][95]. The EU funded project PRISMATICA[75] and ADVISOR[3], completed in 2003, were concerned with the management of public transport networks through CCTV cameras. The UK EPSRC funded project BEHAVE, was concerned with pre - screening of video sequences for the detection of abnormal or crime-oriented behaviour[12]. ISCAPS[44] started in 2005, a consortium of 10 European ICT companies and academic organizations, aims to provide automated surveillance of crowded areas. SERKET, a recently started EU project aims to develop methods to prevent terrorism [42].

Figure 1 illustrates the processes involved in crowd analysis. In a crowd scene the attributes of importance are crowd density, location, speed, etc. This information can be extracted either manually or automatically using computer vision techniques. Crowd models can then be built based on the extracted information. Event discovery is achieved using pre-compiled knowledge of the scene or using the computational model, although both approaches can be combined. In both cases the model is updated with newly extracted information.

The paper is organised as follows. Section 2 introduces research in automatic crowd feature extraction. Section 3 discusses existing work on crowd modelling and crowd event inference. Section 4 and 5 provide some examples of how the two complementary approaches can be bridged.

2 Crowd Information Extraction

The components of crowd analysis from a computer vision perspective are described in Table 1. Essentially, the typology of sensors and their topology influence the scene capture processes; environmental conditions, such as natural and artificial illumination changes often introduce noise; the scene typology affects the type of process one requires to extract the most accurate information of a dyanmic scene.

Visual surveillance methods have been devloped to estimate motion of objects and people in the scene, in isolation or in groups; a review can be found in [38]. When video is analysed for very crowded scenes, conventional computer vision methods are not appropriate, in these cases methods must be designed to cope with extreme clutter. Features from conventional image processing are still employed, such as colour,shape and texture etc. However, sophisticated methods have been developed to retrieve crowd information. In the following sections we will review the existing state of the art.

2.1 Crowd Density Measurement

An important crowd feature is crowd density and it is natural to think that crowd of different density should receive a different level of attention. Polus et al. [74] provide a clear idea of the problem of *level of services* for a pedestrian flow defined as: free flow, restricted flow, dense flow, and jammed flow according to a density metric defined as the number of pedestrians per unit area. scene.Here we review some research either estimating the crowd density directly or counting number of pedestrians which provide information for density estimation.

Sensor typology and topology	Environmental conditions	Scene typology	
Moving or Static platform	Indoor/outdoor	Individual characters	Collective
Number of cameras	Level of clutter	location/velocity/etc.	Crowd density
Type of video sequence: color or gray scale, etc.	Light condition, etc.	Appearance, etc.	Average speed, etc.

Table 1 Features in crowd analysis by computer vision methods.

Research methods have been proposed for crowd analysis which employ background removal techniques. In [95] a reference image with only background is used to classify image pixels as belonging to either pedestrians or background. A functional relationship between the number of pedestrian-classified pixels and number of people is then established manually for the measurement of crowd density. Another example is proposed by Ma et al. [61] using background removal. A mathematical relation for geometric correction for the ground plane is derived. The authors proved that it can be directly applied to all foreground pixels. A linear relation between the number of pixels and number of persons was derived by applying the geometric correction. These works have a typical assumption that the number of foreground pixels are proportional to the number of people, which is only true when there are not serious occlutions between people.[27] makes use of examples to map the global shape feature to configurations of humans directly. This training based algorithm is a quite novel approach but the problem of how to decide the size of the training dataset remains unclear.

Image processing and pattern recognition techniques are also used for the analysis of the scene to estimate the crowd density. Marana et al. [64] assume that images of low-density crowds tend to present coarse texture, while images of dense crowds tend to present fine textures. Self-organising neural maps [65] combined with Minkowski fractal dimensions [63] are employed to deduce the crowd density from the texture of the image. The work by Marana is compared in [76] with another method that uses Chebyshev moments. An optimization of performance under different illumination conditions is discussed. Lin et al. [60] present a system that estimates the crowd size through the recognition of the head contour using Haar wavelet transform (HWT) and support vector machines (SVM).

Approach of information fusion has also be applied, e.g. Yang et al. [94] estimate the number of people directly from groups of image sensors. For each sensor, foreground objects are segmented from the background, and the resulting silhouettes are aggregated over the sensor network. A geometric algorithm is then introduced to limit the number and possible locations of people using silhouettes extracted by each sensor. Alternative methods combine several techniques, to achieve more accurate and reliable measurements. For example, in [89], an edge-based technique is integrated with background removal using a Kalman filter. Marana et al. [62] use different methods including Fourier and Fractal analysis and classifiers to estimate the crowd density level. Kong et al. in [54][55] employ background subtraction and edge detection are employed; the work defined the extracted edge orientation and blob size histograms as features. The relationship between the feature histograms and the number of pedestrian is learned from labelled training data. Obvious more cues may indicate a more accurate solution.

2.2 Recognition

Conventional visual surveillance focuses on object detection and tracking. In essence, image processing techniques are employed to extract the chromatic and shape information of the moving objects and the background for detecting and tracking purposes.

For crowd dynamics modeling, detecting and tracking are also important as they provide the location and velocity features of the dynamics. Crowded scenes add a degree of complexity to the conventional detection and tracking problem of single individuals. In the following sections we concentrate on methodologies for crowded situations.

2.2.1 Face and Head Recognition

Face is the most discriminating feature of the human body and many researchers try to detect pedestrian through face detection. Majority of the existing research employs supervised learning methods. Here we review a few attempts to detect the faces in complex scenes.

Early works like [87] in which a technique using genetic algorithms is employed for face localization in a complex scene. The system proceeds with a training phase to generate a simple object mean image using a single object image, and a test phase using arbitrary images.

However the previous work highly depends on the training set and if the faces appear at different sizes and orientations, it may require a very large training set and long processing time. Hence different techniques have been developed to address the problem of multi-view face detection. [59] proposes a pyramid structure that adopts coarse-to-fine strategy to handle pose variance. Another approach is by Jone et al. [45],in this work different detectors are for different views of the face, and a decision tree is trained to determine the viewpoint class. [39] uses Width-First Search tree structure to improve the performance in both speed and accuracy. These kind of work is quite likely to be adopt into crowd analysis, espeically

from a single camera view, as the problem of human pose and the perspective are both compensated here.

Methodologies for stereo face detections in crowd have also been developed. For example Huang et al. [40] propose a three steps technique: first extracting the likelihood evidence of heads from the stereo image by scale-adaptive filtering; then spurious clues are suppressed from the extracted points according to the average human height; finally the human heads are located by applying a mean-shift algorithm to the likelihood map.

2.2.2 Pedestrian and Crowd recognition

Pedestrian detection and tracking is a well studied problem in computer vision. Many methods have been proposed, such as using the afore mentioned background removal technique, or combining chromatic and shape information of the tracked pedestrians. The following sections discuss the methods that try to provide a solution for pedestrian detection in crowded scenes.

- **Occlusion handling.** Occlusion caused by the high clutter of the pedestrian in crowd scene is the major challenge for crowd detection problem.
  Some research addresses the problem by using human body parts. Wu et al. [92] propose a method to detect multiple-partially occluded human in a single image. Edgelet features are introduced in their work. Part detectors based on edgelet features are learned by a boosting method. Responses of part detectors are combined to form a joint likelihood model that includes cases of multiple, possibly inter-occluded humans. The human detection problem is then formulated as one of maximum a posteriori (MAP) estimation. The models of group of people in [29] are initialised based on segmenting the body into regions by modelling their appearance and spatial distribution. A framework uses maximum likelihood estimation to estimate the best arrangement of people in term of a 2D translation that yields segmentation for the foreground region. Occlusion reasoning is then conducted to recover relative depth information.
  Leibe et al. [58] present a different algorithm that integrates evidence in multiple iterations and from different sources. Local cue is based on a scale-invariant extension of Implicit Shape Model (ISM), and global consistency is enforced by adding the information from global shape cues. Local and global cues are combined via a probabilistic top-down segmentation to detect the pedestrian.
- **Moving Views.**Special solutions are required for moving platforms for some of the applications e.g. for on-board vision system to assist a driver.
  Some of the implementations make assumptions of human appearance.In Broggi et al.'s work [16] a coarse detection of pedestrian is computed through the processing of a single image based on shape of human body assumption of symmetry, size and ratio. Heisele et al. [33] apply spatio-temporal methodologies by recognizing walking pedestrian based on the characteristic motion of the legs of a pedestrian walking parallel to the image plane. Each image is segmented into region-like image parts by clustering pixels in a combined color/postion feature space. A classfier is then used to extract the clusters which are mostly like to be the pedestrian's legs.
  Different from above, Shashua et al. [81] describe a functional and architectural breakdown pedestrian detection system. Single classification is based on a scheme of breaking down the class variability by repeatedly training a set of relatively simple classification performance results. The path from single-frame to system level performance includes the integration of additional cues measure over time, situation specific features and via building up additional object categories consisting of vehicles and stationary background structures.
- **Spatial-temporal methods.** Besides conventional cues of pedestrian appearance, space-temporal cues are used for detection. Brostow et al. [17] tackle the problem by tracking simple image features and probabilistically grouping them into clusters representing independently moving entities. Space-time proximity and trajectory coherence through image space are used as the only probabilistic criteria for clustering. Moreover, this motion-based detection could be easily extended to tracking of individuals in dense crowds by merging the outcomes.
  In extremely cluttered scenes, individual pedestrian cannot be properly segmented in the image. However sometimes the *crowd* within which the pedestrians share a similar purpose can be recognizad. Reisman et al. [79] propose a scheme that uses slices in the spatio-temporal domain to detect inward motion as well as intersections between multiple moving objects. The system calculates a probability distribution function for left and right inward motion and uses these probability distribution functions to infer a decision for crowd detection.

2.3 Tracking

Tracking has been proposed to localize the interested object in time-space. Also the velocity feature can be derived afterwards. Though as a natural extension of detection, tracking has its own problem to recognize and identify pedestrians in the consecutive frames. Tracking could be regarded as the most popular topic in visual surveillance, however currently for crowd analysis, most of the techniques are validated only for multiple (e.g. up to 10) people.

As discussed in the last subsection, occlusions could occur very frequently when there are many objects and people in the scene. Tracking techniques have to overcome the problem in order to continuously track before,

during and after the occurrence of occlusions. A comprehensive review on occlusion handling can be found in [31]. A formulation of the occlusion problem is provided, and the techniques are divided in two groups: merge-split approach, which addresses the problem to reestablish object identities following a split, and straight-through approaches, which maintains object identities at all times.

The following text covers three aspects: the techniques which are developed to track multiple people(objects) without any assumptions of the dependence of their motion, e.g. interactions etc.; the techniques which try to explain the interactions between the pedestrians; and also some practical analysis of handling the problem of occlusion in the crowd situation.

2.3.1 Tracking Methodologies

Crowd scenes increase the complexity of tracking because there are multiple moving objects in the scene. Quite a few techniques are developed based on the colour, geometry and other features for tracking.

- **Likelihood.**Color, edge etc. are the most popular features in tracking. In crowd salient traceable image features are particular interested for tracking. As one of the good candidates, interest points (IPs) are employed in [31] and [67]. In both works the IPs are obtained by a popular colour Harris detector. Gabriel characterized IPs by their position relative to the estimated centre of the object and Mathes built a point distribution model between ASM and AAM. Both of the methods require a pre-defined region (or object) of interest. There salient features are benefit from their robustness under different light conditions. The tracking inference using these features can work better under occlusions than using the entire contour. Therefore the usage of those features could be more appliable to large amount of people in the scene.
- **Human body model.**Methods using models of human bodies or human body parts have been developed for tracking in complex crowded scenes, which are usually completed with probabilistic frameworks. Zhao et al. [99][100] have been working on the former approach, using explicit 3D human shape models. The problem of detection and tracking are formulated as one of Bayesian inference to find the best interpretation given the image observations, The latter one as the work from Wu et al. [93] extend the previous detection work in [92] (which has been discussed) using edgelet features to human body part detectors. Tracking is implemented by probabilistic data association, i.e. matching the object hypotheses with the detected response.
- **Tracking inference strategies.** Tracking inference strategies have been developed for the problem of tracking multiple objects. For non-linear and non-Gaussian dynamic models, particle filter technique ,also known as CONDENSATION[43],is one of the most popular among those. Particle filters are sequential Monte Carlo methods based upon a point mass (or 'particle') representations of probability densities [28]. Large portion of multiple object tracking work have employed this technique. For example, Venegas et al.[90] use particle filter to track the moving objects by generating hypotheses on the top-view reconstruction of the scene . Okuma et al. [72] combine mixture particle filters and Adaboost algorithm. Sidenbladh et al. [82] extend the particle filter formulation according to finite set statistics (FISST) for tracking. Cai et al. [18] tackle the problem by embedding the meanshift algorithm into the particle filter framework. Koller-Meier et al. [53] introduce an extension of the CONDENSATION algorithm that relied on a single probability distribution of describe the likely states of multiple objects. Kang et al. [46] propose the discrete shape model and the competition rule to improve the performance of the condensation tracker for real time tracking.
To address data association problem, There are Multiple Hyphotheses Tracker (MHT)and Joint Probabilistic Data Association Filter(JPDAF). MHT tries to keep the track of all the possible hyphotheses over time [78].A details summary and a discussion of MHT for multiple target tracking is included in [13].MHT suffers from the storage of the redundant track, hence some of the work propose extensions and modifications to the algorithm to get better performances, e.g. [32].JPDAF computes a Bayesian estimation of correspondence between the different features and the different objects, e.g. Rasmussen and Hager [77] apply this technique with color region and snake-based tracker. An approach has been introduced by Karlsson [47], which uses Monte Carlo method.
The fusion of the different cues from a number of detection and tracking algorithms are also used to produce a more robust tracker. Siebel et al. [83] propose a tracking system containing three co-operating parts: an Active Shape Tracker, a Region Tracker, and a Head detector.[85] proposes an approach based on the principles of self-organization of the integration mechanism and self-adaptation of the cue models during the tracking. Cues from different sensors and models can increase dimension of information, which is preferable in the multiple objects situations. However the goodness of integration scheme is very crucial in these algorithms.

2.3.2 Tracking Interacting People

In certain cases, interaction happens frequently in crowded scene. Researchers have shown great interest in studying these interactions to get the new perspectives on tracking techniques.

Some of the work formulate the interaction to enhance the tracking scheme. For example both Smith et al. [84] and Khan et al. [49] propose to use Markov Chain Monte Carlo (MCMC) and the particle filter. Smith used a joint multi-object state-space formulation and a trans-dimensional MCMC particle filter to recursively estimate the multi-object configuration and search efficiently the state-space. Khan developed a joint tracker that included a motion model to maintain the identity of targets throughout and interaction, thus to reduce tracker failure. Predefined motion models are used in this approach, with the trade-off between improving the tracking performance in crowd with known interations and the adaption of the motion model to arbitrary crowd.

Some researchers interpret interactions as relationships between pedestrians and a group (pedestrian merging/splitting into groups). Marques et al. [66] propose a two-layer solution to overcome the problem. The first layer produces a set of spatio temporal strokes based on low level operations to track the active regions. The second layer performs a consistent labelling of detected segments using a statistical model based on Bayesian networks which is recursively computed during the tracking operation. Mckenna et al. [69] perform tracking at three levels: regions, people and groups. Background subtraction is used to cope with shadows and unreliable colour cues. Colour information is used to disambiguate occlusions and to provide qualitative estimates of depth ordering and position. Pedestrian merging and group splitting are frequent phenomena in the crowded scene, however the major challenge for this kind of methods is to recover the object lable after splitting from the group.

Sullivan et al. [86] label tracking targets by exploring the trajectories. Trajectories of when a target is isolated are found and it is claimed that these trajectories end when targets interact. A graph structure has been formed by the interactions of these trajectories. This method could be very useful for offline crowd analyzing but for online processing it may have a bottleneck in the storage of the trajectories.

2.3.3 Tracking from Multiple Views

For large public areas the use of a multi-camera system is required to cover most of the monitored areas.

For the multi-camera system arrangement, Mittal et al. [70] present a system named M2Tracker using multiple synchronized cameras located far from each other for segmenting, detecting and tracking multiple people in a cluttered scene. First, a region-based stereo algorithm is introduced for finding 3D points inside an object. Then, a scheme is developed dynamically assigning priors for different objects at each pixel. Finally, the evidences gather from different camera pairs are combined using occlusion analysis to obtain a globally optimum detection and tracking of objects. A different arrangement of cameras is used in [20]. The method uses both static and Pan-Tilt-Zoom (PTZ) cameras. The static cameras are used to locate people in the scene, while the PTZ cameras *lock-on* to the individuals and provide visual attention. The underlying visual processes rely on colour segmentation, movement tracking and shape information to locate target candidates and colour indexing methods to register these candidates with the PTZ cameras.

Meanwhile special techniques have been developed for the tracking from multiview, normally a planar homography constraint would be included. For example in [48], feet regions of the people are located by the constraint. The contiguous spatio-temporal region formed by the feet regoins belonging to the same person are clustered as the track of the person. In [50] people's ground points are located and a multi-hypothesis framework using particle filter is developed for tracking.

3 Crowd modelling and events inference

Dynamics in public spaces can indeed be recurrent. Crowd information can be better exploited to indicate the status of the crowd so that crowd events can be inferred. Crowd models have been built to represent these status, either implicitly or explicitly. On the other hand, some research makes direct use of crowd information instead of building models. In such cases, the events are usually inferred based on some prior knowledge of the properties of the particular scene and the crowd.In this section, crowd models and events inference in computer vision will be presented as well as some crowd models from non vision areas.

3.1 Crowd models and crowd events inference in computer vision

In computer vision crowd modelling is achieved based on the extracted information from visual data and normally can be employed in crowd events inference. Meanwhile there are also some approaches attempt to infer events without construction of models. Here examples are given for both of the cases.

- In computer vision approach crowd models are built as representations of recurrent behaviours by analysing video data of the crowd through vision methods. Zhan et al. [96][98][97] propose a crowd model using accumulated motion and foreground (moving objects) information of a crowded scene. This was implemented by two probability density functions (PDFs): Occurrence PDF and Orientation PDF associated with every non-overlapped block ($n \times n$ pixels) of the image. The Occurrence model indicates the frequency of the block covered by the foreground features, and the Orientation PDF indicates the probability of each orientation of the foreground feature on that block could take. A preliminary data mining of the PDF models

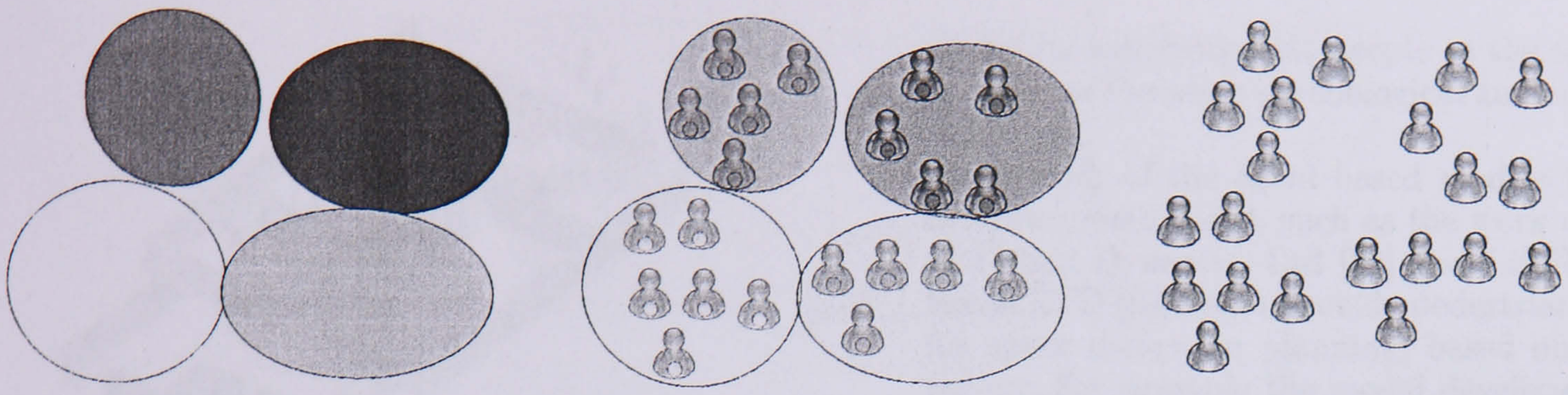


Fig. 2 (left) Macroscopic, (centre) Mesoscopic, (right) Microscopic.

is given to find the major (most frequent) path of the crowd.

Andrade et al. [7][6][8] characterize crowd behaviour by observing the crowd optical flow associated with the crowd and use unsupervised feature extraction to encode normal crowd behaviour. The unsupervised feature extraction applies spectral clustering to find the optimal number of models to represent normal motion patterns. The motion models are HMMs to cope with the variable number of motion samples that might be present in each observation window. The objective of this model is to detect abnormal event in crowd scenes.

- Apart from building models, in crowd monitoring systems of computer vision, the extracted information is used to recognize the event, usually under some assumptions of involved crowds. Early work on crowd monitoring using image processing is reviewed by Davies et al. [26].

More recent work like in Boghossian et al. [14], a system is presented using computer vision techniques to estimate the paths and directions of crowd flows in CCTV images and improve the perception of scene dynamics by offering on-line illustrations.

Maurin et al. [68] propose a system to detect, track, and monitor both pedestrians (crowds) and vehicles. The system contains a detection scheme based on optical flow that can locate vehicles, individual pedestrians and crowd. The detection phase is followed by the tracking phase that tracks all the detected entities. Traffic objects are tracked and a rich set of descriptors are computed for each object including a wealth of information(position, velocity, acceleration/deceleration, bounding box, and shape).

Cupillard et al. carry out event recognition by means of *behaviour*, in [25][24] an approach using multiple cameras is presented. The algorithm relies on both low level motion detection and tracking, and a high level module which recognizes pedefined scenarios corresponding to specific behaviours.

Michael et al. [19] present a method jointly performing recognition of complex events and linking fragmented tracks. The recognition work is implemented by combing appearance and kinematic constraints from tracking and constraints from a hypothesized event model.

In these methods specially assumptions of crowd are usually involved, indicating that some prior knowledge are required for events inference. These methods may be very efficient and computational unexpensive for some particular systems that the interested events are simple and clear, though this is not always the case in general situations.

3.2 Crowd models from non vision approach

Computational models aim at describing and predicting the collective effects of crowd behaviour by identifying the relationship between crowd features. There are three distinct philosophies for modelling a crowd; traffic analysis [30] proposes a categorisation, where crowd models can be defined as microscopic, mesoscopic and macroscopic. The microscopic model deals with pedestrians as discrete individuals; the macroscopic model deals with a crowd as a whole and the mesoscopic model combines the properties of the previous two, either keeping a crowd as a homogeneous mass but considering an internal *force* or keeping the characters of the individuals while maintaining a general view of the entire crowd (Figure 2). In the following some typical techniques of crowd modelling will be introduced and some examples will be given.

- **Physics inspired models.** Several quantitative factors of crowds and pedestrians are measurable. This fact encourages researchers to look for the mathematical models of crowd dynamics.

Helbing has a series of work upon this topic. His first experiment is in [34], with a stochastic formulation at microscopic level, a gas kinetic formulation at the mesoscopic level, and fluid dynamic equations at the macroscopic level for the crowd model. Later he [36] proposes another more popular microscopic model: social force model based on the social field theory. The social force represents the effect of the environment; it is a quantity that describes the concrete motivation to act. In [37] the model is used to reproduce the emergence of several empirically observed collective patterns of motion. Moreover, simulations

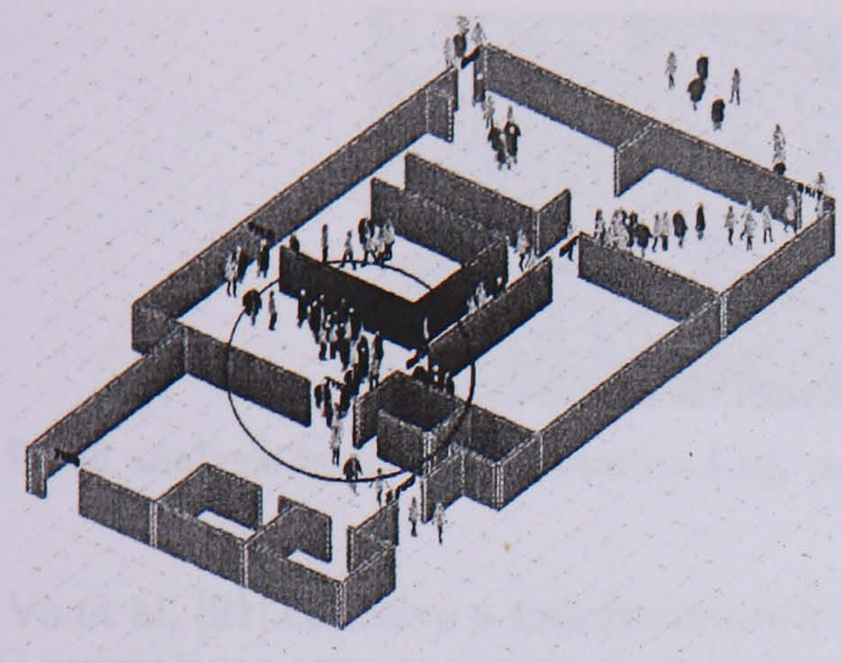


Fig. 3 A screenshot of XiaoShan Pan's work: human agents try to self-organise into exiting lines.

of crowd dynamics based on a generalized force model for the escape panic phenomenon are presented in [35]. Also quite a few works have been developed upon this work, for example in [21] additional pattern is introduced by considering the unequal information distribution in a crowd.

In contrast to the former works, macroscopic models often draw an analogy between the crowd a continuum responding to local influence. Hughes [41] is more interested in modelling rational, goal-directed pedestrians. His theory does not govern the behaviours of any individual pedestrians, as it is a macroscopic model; instead the crowd is divided into (approximate) pedestrian types where pedestrians in each type have the same walking habits.

Physics inspired models are widely used to study crowds from different perspectives, e.g. to study the effects of introducing autonomous robots into crowds [52], or to model a historic scene [80]. The interrelations of the factors and equations (e.g. employing the same factors in different level equations) imply the possibility of having a model encompassing all the levels. Also the quantitative analysis of crowd dynamics can be relatively easy to be adapted into computer-based algorithms.

- **Agent based models.** These are qualitative models include employing fuzzy methods to describe the relations of factors and crowd motion instead of pure mathematical methods. Agent-based models use agents to represent the pedestrian or the crowd. Many examples are from the former, e.g. in [71] crowd, crowd individuals have their own emotional parameter to govern behaviour while they belong to a collection of goal-directed groups on mesoscopic level.

In [73] the agents are modelled following the concept of non-adaptive behaviours. Non-adaptive crowd behaviours refer to the destructive actions that a crowd may experience in emergency situations. The human and social models are categorised into the individual, the interactions among individuals, and the group and the environment three non-independent levels. (Figure 3). Brenner et al. [15] provide an example model by assuming that people at the same location experience the same psychological and environmental influences.

Some work of the agent-based models have already been commercialised, such as the work of Keith Still at Crowd Dynamics Ltd [22] and LEGION international LTD [57], both provide pedestrian simulations for space design an planning, based on agent technology. For example the model developed by Crowd Dynamics Ltd aims to simulate how people react to their environment in a variety of conditions (Figure 4).

Usually, these examples employ agent to act as individual pedestrians and only concern the microscopic level.

- **Cellular automation models.** Another research approach employs the construction of local models, where active area has been virtually devided into cells. An example is a commercialized tool EGRESS of AEA Technology Plc [4]. In EGRESS the floor area of an environment is covered with cells equivalent to the minimum occupancy area of a person. The used cells can represent free floor area, a wall or a blockage, a cell with a person, or a region with some other attributes. Pedestrians move between cells following predefined rules. Krez et al. [56] present a model of pedestrian motion using both floor field and agents. The model consists of three floor fields: *Static floor field* for each cell contains the information of the distance to the exit; *Dynamic floor field* changes by the motion of the pedestrains and the third floor fields saves the distance of a cell to the next wall.
- **Nature based models.** Some of the models take their inspiration from nature. The emotional ant model [11] extends the psychological information using biologically inspired ant agent as a crowd. Four different cognitive behaviours of crowd have been modelled and transition behaviour is modelled using fuzzy logic.

Kirchner et al. [51] apply a bionics approach to the cellular automation model by describing the interaction between the pedestrians using ideas from chemotaxis. The simulation of the evacuation from a large room is also presented to show the ability of the model to represent different types of behaviours.

4 Examples of bridging the research

Computer simulation can be used to evaluate the developed system's performance. Considering that real visual evidences for abnormal scenarios are rare or unsafe to reproduce in a controllable way, Andrade et al. [5] have developed an approach generating simulations to allow training and validation of computer vision systems applied to crowd monitoring. The simulation is generated by a pedestrian path model and a pedestrian body

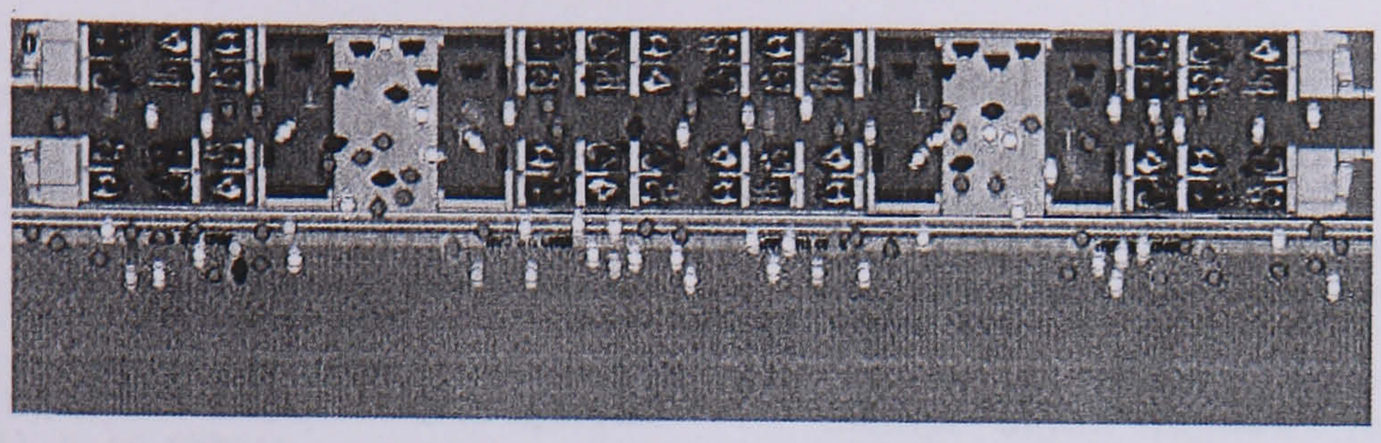


Fig. 4 Dwell analysis by Crowd Dynamics Ltd, using agent to assess the throughput of specific geometric designs.

model. Vu et al. [91] conceive a test framework that generats 3D animations corresponding to behaviours recognised by an interpretation system. In other words, this is a test system for a given interpretation system by generating test animations.

Non-vision models can be borrowed for computer vision analysis. Anotonini et al. [9][10] propose a framework using discrete choice model, which is widely used in traffic simulations, for pedestrian dynamics modelling. The framework models short-term behaviours of individuals as a response to the presence of other pedestrians. The model is calibrated using data from actual pedestrian movements, manually taken from video sequences. The work is applied to the problem of the target detection in the particular case of pedestrian tracking.

5 Conclusions and Discussion

This paper provides a review on current crowd analysis work in computer vision. Perspectives from sociology, psychology and computer graphics are presented, as these research fields also have contributed to an in-depth study on crowd analysis and modelling. Sociological and psychological studies on the crowd phenomenon make use of human observations. Their studies indicate various ways to represent and model people relationships in isolation and as part of a more or less large group of people. The microscopic, mesoscopic and macroscopic levels are defined to characterise people as individuals part of crowd. The computer vision approach tackles the problem of extracting automatically information sufficient to characterise some special crowd events.

Anotonini gives a good example of employing non-vision model, however, his work only uses very limited information and only acts as a *clear* tracker. The works of non-vision analysis present in our paper show that all of the factors or information extracted from the real world using computer vision techniques are inter-related. Moreover, they have proposed the probable relationships in their works, which represent the human understanding of crowd dynamics. On the other hand, computer vision techniques have the ability of exploiting the special environmental constraints, which could be applied to calibrate the proposed models. We can claim that it is possible that to develop intelligent systems combining these works with computer vision approaches. The system would be capable of automatically understanding and modelling the crowd behaviours which works at both instantaneous and recurrent level.

Acknowledgement

- This research was partially funded by the British Telecommuncation Group PLC.
- The authors would like to thank the Associate Editor Dr. Hai Tao and the two reviewers for valuable comments that helped improve the clarity of presentation of this paper.

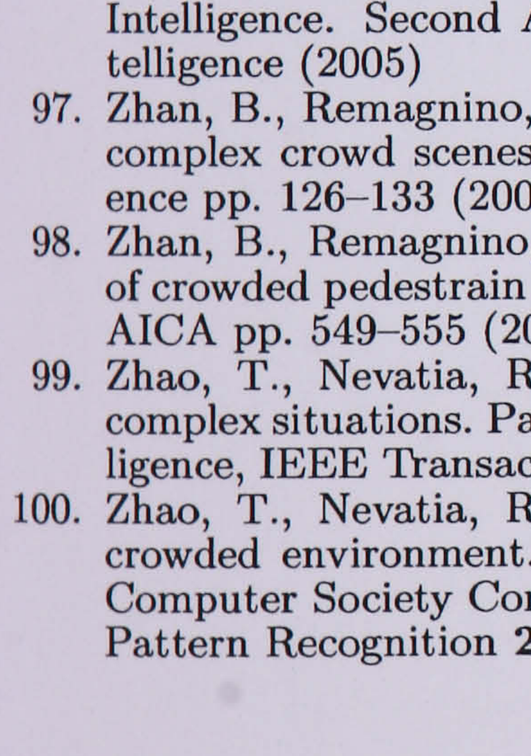
References

1. 2004-2005, 2005-2006, CIFE, Seed, Project, Stanford, University: Http://eil.stanford.edu/egress/
2. Adang, O.M., Stott, C.: A European study of the interaction between police and crowds of foreign nationals considered to pose a risk to public order. Http://policestudies.homestead.com/Euro2004.html
3. ADVISOR: Http://advisor.matrasi-tls.fr/
4. AEA, Techology: A technical summary of the aea egress code. Technical Report **1** (2002)
5. Andrade, E., Fisher, R.: Simulation of crowd problems for computer vision. In: First International Workshop on Crowd Simulation, vol. 3, pp. 71–80 (2005)
6. Andrade, E., Fisher, R.: Hidden Markov models for optical flow analysis in crowds. In: Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)-Volume 01, pp. 460–463. IEEE Computer Society Washington, DC, USA (2006)
7. Andrade, E., Fisher, R.: Modelling crowd scenes for event detection. In: Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)-Volume 01, pp. 175–178. IEEE Computer Society Washington, DC, USA (2006)
8. Andrade, E.L., Blunsden, S., Fisher, R.B.: Performance analysis of event detection models in crowded scenes. In: Proc. Workshop on Towards Robust Visual Surveillance Techniques and Systems at Visual Information Engineering 2006, pp. 427–432. Bangalore, India (2006)
9. Antonini, G., Bierlaire, M., Weber, M.: Simulation of pedestrian behaviour using a discrete choice model calibrated on actual motion data. In: 4th STRC Swiss Transport Research Conference. Ascona (2004)
10. Antonini, G., Venegas, S., Thiran J.P.and Bierlaire, M.: A discrete choice pedestrian behaviour model in visual tracking systems. In: Advanced Concepts for Intelligent Vision Systems, pp. 273–280. Brussels, Belgium (2004)

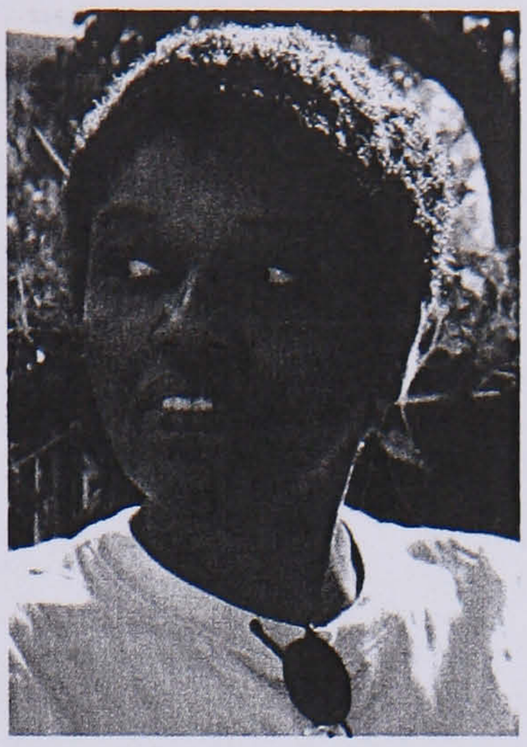
11. Banarjee, S., Grosan, C., Abarha, A.: Emotional ant based modeling of crowd dynamics. In: Seventh International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC'05), pp. 279–286 (2005)
12. BEHAVE: Http://www.homepages. informatics. ed.ac.uk /rbf /BEHAVE/
13. Blackman, S.: Multiple hypothesis tracking for multiple target tracking. Aerospace and Electronic Systems Magazine, IEEE **19**(1), 5–18 (2004)
14. Boghossian, B., Velastin, S.: Motion-based machine vision techniques for the management of large crowds. In: the 6th IEEE International Conference on Electronics, Circuits and Systems, vol. 2 (1999)
15. Brenner, M., Wijermans, N., Nussle, T., de Boer, B.: Simulating and controlling civilian crowds in robocup rescue. In: inproceedings of RoboCup 2005: Robot Soccer World Cup IX. Osaka (2005)
16. Broggi, A., Bertozzi, M., Fascioli, A., Sechi, M.: Shape-based pedestrian detection. In: inproceedings of the. IEEE Intelligent Vehicles Symposium 2000. Dearbon (MI), USA (2000)
17. Brostow, G., Cipolla, R.: Unsupervised Bayesian Detection of Independent Motion in Crowds. In: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Volume 1, pp. 594–601. IEEE Computer Society Washington, DC, USA (2006)
18. Cai, Y., de Freitas, N., Little, J.J.: Robust visual tracking for multiple targets. In: European Conference on Computer Vision, *LNCS*, vol. 3954, pp. 107–118. Springer (2006)
19. Chan, M.T., Hoogs, A., Bhotika, R., Perera, A., Schmiederer, J., Doretto, G.: Joint recognition of complex events and track matching. In: CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1615–1622. IEEE Computer Society, Washington, DC, USA (2006). DOI http://dx.doi.org /10.1109 /CVPR. 2006.160
20. Chang, T., Gong, S., Ong, E.: Tracking multiple people under occlusion using multiple cameras. In: British Machine Vision Conference, pp. 566–575 (2000)
21. Chu, J., Li, J., Xu, M., Zhao, L.: Simulating escape panic based on the mechanism of asymmetric information distribution. In: Complex Systems Summer School Final Project Papers. Santa Fe, NM (2005). Santa Fe Institute
22. Crowd, Dynamics: Http://www.crowddynamics.com/
23. Crowd, MAGS: Http://www2.ift.ulaval.ca/ muscam-ags/Dnd - crowdmags - project.htm
24. Cupillard, F., Bremond, F., Thonnat, M.: Behaviour recognition for individuals, groups of people and crowd. IEE Seminar Digests **7** (2003)
25. Cupillard, F., Bremond, F., Thonnat, M., INRIA, F.: Group behavior recognition with multiple cameras. Applications of Computer Vision, 2002.(WACV 2002). Proceedings. Sixth IEEE Workshop on pp. 177–183 (2002)
26. Davies, A., Yin, J., Velastin, S.: Crowd monitoring using image processing. Electronics & Communication Engineering Journal **7**(1), 37–47 (1995)
27. Dong, L., Parameswaran, V., Ramesh, V., Zoghlami., I.: Fast Crowd Segmentation Using Shape Indexing. Rio de Janeiro, Brazil (2007)
28. Doucet, A., Godsill, S., Andrieu, C.: On sequential Monte Carlo sampling methods for Bayesian filtering (2000)
29. Elgammal, A., Davis, L.: Probabilistic framework for segmenting people under occlusion. In: Eighth IEEE International Conference on Computer Vision, 2001. ICCV 2001. Proceedings., vol. 2, pp. 145–152 (2001)
30. FHWA.: Traffic analysis tools primer,traffic analysis toolbox (1) (2004). Http://ops.fhwa.dot.gov /traffic-analysistools /tat-vol1/index
31. Gabriel, P., Verly, J., Piater, J., Genon, A.: The state of the art in multiple object tracking under occlusion in video sequences. Advanced Concepts for Intelligent Vision Systems pp. 166–173 (2003)
32. Han, M., Xu, W., Tao, H., Gong, Y.: An algorithm for multiple object trajectory tracking. Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on **1** (2004)
33. Heisele, B., Woehler, C.: Motion-based recognition of pedestrians. Fourteenth International Conference on Pattern Recognition, 1998. Proceedings. **2**, 1325–1330 (1998)
34. Helbing, D.: Models for pedestrian behavior (1992). URL http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/9805089
35. Helbing, D., Farkas, I., Vicsek, T.: Simulating Dynamical Features of Escape Panic. Letters to Nature **407**, 487–490 (2000)
36. Helbing, D., Molnár, P.: Social force model for pedestrian dynamics. Physical Review E **51**(5), 4282–4286 (1995)
37. Helbing, D., Molnar, P.: Self-organization phenomena in pedestrian crowds (1997). URL http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/9806152
38. Hu, W., Tan, T., Wang, L., Maybank, S.: A survey on visual surveillance of object motion and behaviors. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews **34**(3), 334–352 (2004)
39. Huang, C., Ai, H., Li, Y., Lao, S.: Vector boosting for rotation invariant multi-view face detection. In: Tenth IEEE International Conference on Computer Vision, vol. 1, pp. 446–453 (2005)
40. Huang, X., Li, L., Sim, T.: Stereo-based human head detection from crowd scenes. International Conference on Image Processing, 2004. ICIP'04. **2**, 1353–1356 (2004)
41. Hughes, R.: A continuum theory for the flow of pedestrians. Transportation Research Part B: Methodological **36**(6), 507–535 (2002)
42. INRIA: Http://www.inria.fr /rapportsactivite /RA2005 /orion /uid1.html
43. Isard, M., Blake, A.: A mixed-state CONDENSATION tracker with automatic model-switching. In: IEEE International Conference on Computer Vision, pp. 107–112 (1998). Url: citeseer.ist.psu.edu/isard98mixedstate.html
44. ISCAPS: Http://www.iscaps.reading.ac.uk/home.htm
45. Jones, M., Viola, P.: Fast multi-view face detection. Mitsubishi Electric Research Lab TR-20003-96 (2003)
46. Kang, H., Kim, D., Bang, S.: Real-time multiple people tracking using competitive condensation. Proc. of the Intl. Conference on Pattern Recognition **1**, 413–416 (2002)
47. Karlsson, R., Gustafsson, F.: Monte Carlo data association for multiple target tracking. Target Tracking: Algorithms and Applications (Ref. No. 2001/174), IEE **1** (2001)
48. Khan, S.M., Shah, M.: A multiview approach to tracking people in crowded scenes using a planar homography constraint. In: 9th European Conference on Computer Vision, *LNCS*, vol. 3954, pp. 133–146. Springer (2006)
49. Khan, Z., Balch, T., Dellaert, F.: MCMC-based particle filtering for tracking a variable number of interacting targets. IEEE Transactions on Pattern Analysis and Machine Intelligence **27**(11), 1805–1819 (2005)

50. Kim, K., Davis, L.S.: Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering. In: European Conference on Computer Vision, *LNCS*, vol. 3953, pp. 98–109. Springer (2006)
51. Kirchner, A., Schadschneider, A.: Simulation of evacuation processes using a bionics-inspired cellular automaton model for pedestrian dynamics. Physica A: Statistical Mechanics and its Applications **312**(1-2), 260–276 (2002)
52. Kirkland, J., Maciejewski, A.: A simulation of attempts to influence crowd dynamics. IEEE International Conference on Systems, Man, and Cybernetics pp. 4328–4333 (2003)
53. Koller-Meier, E., Ade, F.: Tracking multiple objects using the Condensation algorithm. Robotics and Autonomous Systems **34**(2-3), 93–105 (2001)
54. Kong, D., Gray, D., Tao, H.: Counting Pedestrians in Crowds Using Viewpoint Invariant Training. British Machine Vision Conference (2005)
55. Kong, D., Gray, D., Tao, H.: A viewpoint invariant approach for crowd counting. Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)-Volume 03 pp. 1187–1190 (2006)
56. Kretz, T., Schreckenberg, M.: F.a.s.t. - floor field- and agent-based simulation tool (2006)
57. Legion: Http://www.legion.biz/about /index.html
58. Leibe, B., Seemann, E., Schiele, B.: Pedestrian detection in crowded scenes. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005. **1** (2005)
59. Li, S.Z., Zhu, L., Zhang, Z., Blake, A., Zhang, H., Shum, H.: Statistical learning of multi-view face detection. In: Proceedings of the 7th European Conference on Computer Vision-Part IV, pp. 67–81. Springer-Verlag, London, UK (2002)
60. Lin, S., Chen, J., Chao, H.: Estimation of number of people in crowded scenes using perspective transformation. IEEE Transactions on Systems, Man and Cybernetics, Part A, **31**(6), 645–654 (2001)
61. Ma, R., Li, L., Huang, W., Tian, Q.: On pixel count based crowd density estimation for visual surveillance. IEEE Conference on Cybernetics and Intelligent Systems **1** (2004)
62. Marana, A., da Costa, L., Lotufo, R., Velastin, S.: On the Efficacy of Texture Analysis for Crowd Monitoring. Proceedings of the International Symposium on Computer Graphics, Image Processing, and Vision-Volume 00 p. 354 (1998)
63. Marana, A., Da Fontoura Costa, L., Lotufo, R., Velastin, S.: Estimating crowd density with Minkowski fractal dimension. IEEE International Conference on Acoustics, Speech, and Signal Processing, 1999. ICASSP'99. Proceedings., **6**, 3521–3524 (1999)
64. Marana, A., Velastin, S., Costa, L., Lotufo, R.: Estimation of crowd density using image processing. IEE Colloquium on Image Processing for Security Applications (Digest No: 1997/074), p. 11 (1997)
65. Marana, A., Velastin, S., Costa, L., Lotufo, R.: Automatic estimation of crowd density using texture. Safety Science **28**(3), 165–175 (1998)
66. Marques, J., Jorge, P., Abrantes, A., Lemos, J.: Tracking Groups of Pedestrians in Video Sequences. IEEE 2003 Conference on Computer Vision and Pattern Recognition Workshop **9**, 101 (2003)
67. Mathes, T., Piater, J.: Robust non-rigid object tracking using point distribution models. British Machine Vision Conference **2** (2005)
68. Maurin, B., Masoud, O., Papanikolopoulos, N.: Monitoring crowded traffic scenes. The IEEE 5th International Conference on Intelligent Transportation Systems, 2002. Proceedings. pp. 19–24 (2002)
69. McKenna, S., Jabri, S., Duric, Z., Rosenfeld, A., Wechsler, H.: Tracking groups of people. Computer Vision and Image Understanding **80**(1), 42 56 (2000)
70. Mittal, A., Davis, L.: M 2 Tracker: A Multi-View Approach to Segmenting and Tracking People in a Cluttered Scene. International Journal of Computer Vision **51**(3), 189–203 (2003)
71. Musse, S., Thalmann, D.: A Model of Human Crowd Behavior: Group Inter-Relationship and Collision Detection Analysis. Proc. Workshop of Computer Animation and Simulation of Eurographics **97**, 39 51 (1997)
72. Okuma, K., Taleghani, A., de Freitas, N., Little, J., Lowe, D.: A boosted particle filter: Multitarget detection and tracking. European Conference on Computer Vision **1**, 28 39 (2004)
73. Pan, X., Han, C., Dauber, K., Law, K.: Human and social behavior in computational modeling and analysis of egress. Automation in Construction **15**(4), 448 461 (2006)
74. Polus, A., Schofer, J., Ushpiz, A.: Pedestrian Flow and Level of Service. Journal of Transportation Engineering **109**(1), 46 56 (1983)
75. PRISMATICA: Http://prismatica.king.ac.uk/
76. Rahmalan, H., Nixon, M., Carter, J.: On Crowd Density Estimation for Surveillance. The Institution of Engineering and Technology Conference on Crime and Security pp. 540C 545 (2006)
77. Rasmussen, C., Hager, G.: Joint probabilistic techniques for tracking multi-part objects. Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on pp. 16 21 (1998)
78. Reid, D.: An algorithm for tracking multiple targets. Automatic Control, IEEE Transactions on **24**(6), 843 854 (1979)
79. Reisman, P., Mano, O., Avidan, S., Shashua, A., Ltd, M., Jerusalem, I.: Crowd detection in video sequences. Intelligent Vehicles Symposium, 2004 IEEE pp. 66 71 (2004)
80. R.R., C., Hughes, R.: Mathematical modelling of a mediaeval battle: the battle of agincourt. Mathematics and Computers in Simulation **64(2)**, 259 269 (2004)
81. Shashua, A., Gdalyahu, Y., Hayun, G.: Pedestrian detection for driving assistance systems: single-frame classification and system level performance. Intelligent Vehicles Symposium, 2004 IEEE pp. 1 6 (2004)
82. Sidenbladh, H., Wirkander, S.: Tracking random sets of vehicles in terrain. Proc. 2003 IEEE Workshop on Multi-Object Tracking **9**, 98 (2003)
83. Siebel, N., Maybank, S.: Fusion of multiple tracking algorithms for robust people tracking. European Conference on Computer Vision pp. 373 387 (2002)
84. Smith, K., Gatica-Perez, D., Odobez, J.: Using particles to track varying numbers of interacting people. Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Volume 1-Volume 01 pp. 962 969 (2005)
85. Spengler, M., Schiele, B.: Towards robust multi-cue integration for visual tracking. Machine Vision and Applications **14**(1), 50–58 (2003)
86. Sullivan, J., Carlsson, S.: Tracking and labelling of interacting multiple targets. In: European Conference on Computer Vision, *LNCS*, vol. 3953, pp. 619 632. Springer (2006)
87. Swets, D., Punch, B.: Genetic algorithms for object localization in a complex scene. IEEE International Conference on Image Processing pp. 595 598 (1995)
88. U.S. Census Bureau: Http://www.census.gov/ ipc /www /worldpop.html
89. Velastin, S., Yin, J., Davies, A., Vicencio-Silva, M., Allsop, R., Penn, A.: Automated measurement of crowd

density and motion using imageprocessing. Road Traffic Monitoring and Control, 1994., Seventh International Conference on pp. 127–132 (1994)
90. Venegas, S., Knebel, S., Thiran, J.: Multi-object tracking using particle filter algorithm on the top-view plan. Technical report,LTS-REPORT-2004-003, EPFL (2004). Http :// infoscience.epfl.ch / getfile.py? mode= best & recid=87041
91. Vu, V., Bremond, F., Thonnat, M.: Human Behaviour Visualisation and Simulation for Automatic Video Understanding. Proc. of the 10th Int. Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG–2002), Plzen–Bory, Czech Republic pp. 485–492 (2002)
92. Wu, B., Nevatia, R.: Detection of Multiple, Partially Occluded Humans in a Single Image by Bayesian Combination of Edgelet Part Detectors. Tenth IEEE International Conference on Computer Vision, 2005. ICCV 2005 **1**, 90–97 (2005)
93. Wu, B., Nevatia, R.: Tracking of multiple, partially occluded humans based on static body part detection. In: CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 951–958 (2006)
94. Yang, D., Gonzalez-Banos, H., Guibas, L.: Counting people in crowds with a real-time network of simple image sensors. Ninth IEEE International Conference on Computer Vision, 2003. Proceedings. pp. 122–129 (2003)
95. Yin, J., Velastin, S., Davies, A.: Image Processing Techniques for Crowd Density Estimation Using a Reference Image. Proc. 2nd Asia-Pacific Conf. Comput. Vision **3**, 6–10 (1995)
96. Zhan, B., Remagnino, P., Velastin, S.: Analysing Crowd Intelligence. Second AIxIA Workshop on Ambient Intelligence (2005)
97. Zhan, B., Remagnino, P., Velastin, S.: Mining paths of complex crowd scenes. Lecture notes in computer science pp. 126–133 (2005). ISBN/ISSN 3-540-30750-8
98. Zhan, B., Remagnino, P., Velastin, S.: Visual analysis of crowded pedestrain scenes. XLIII Congresso Annuale AICA pp. 549–555 (2005)
99. Zhao, T., Nevatia, R.: Tracking multiple humans in complex situations. Pattern Analysis and Machine Intelligence, IEEE Transactions on **26**(9), 1208–1221 (2004)
100. Zhao, T., Nevatia, R.: Tracking multiple humans in crowded environment. Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition **2**, II–406–II–413 (2004)

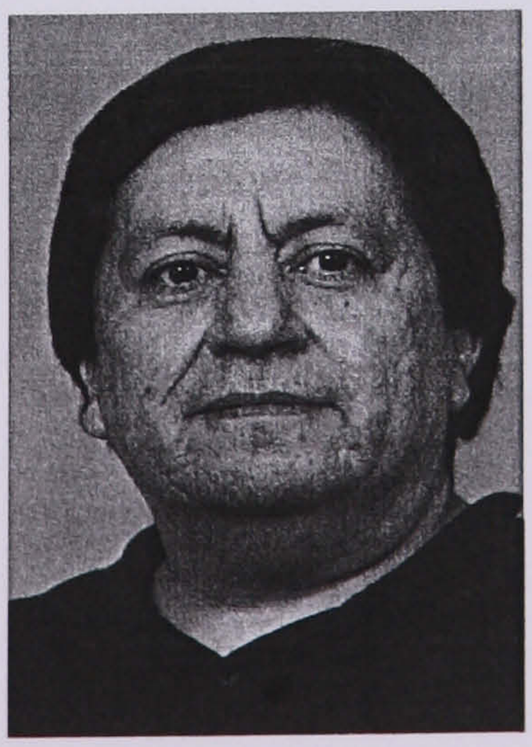


Beibei Zhan received the B.Eng. degree in Computer Science and Technology from Xi'an Jiaotong University, China, in 2003, and M.Sc. degree in Vision Imaging and Virtual Environments from University College London,Uk, in 2004. Currently she is a Ph.D. student at the Digital Imaging Research Centre at Kingston University UK. Her current research interests include vision based behaviour analysis, machine learning, ambient intelligence and visual surveillance.



Dorothy Monekosso (PhD, 1999) is a Senior Lecturer in the Faculty of Computing, Information Science and Mathematics at Kingston University. Prior to joining Kingston University, Dr Monekosso worked for over 10 years in the field of spacecraft engineering and spacecraft autonomy. Her current research interests include machine learning, intelligent systems, and robotics.

Paolo Remagnino (1988, Laurea Diploma and 1993, PhD) is a Reader in the Faulty of Computing, Information Systems and Mathematics at Kingston University. Dr Remagnino is one of the main promoters of Ambient Intelligence and a research in Video Surveillance. Dr Remagnino is a member of the editorial board of the Machine Vision and Applications and the Expert Systems journals and an Associate Editor of the International Journal of Robotics and Automation. His research interests include machine vision, learning and intelligence.



Sergio A. Velastin received the B.Sc. degree in Electronics, M.Sc. (Research) degree in Digital Image Processing and the Ph.D. from the University of Manchester, UK, in 1978, 1979 and 1982 respectively. Currently he is a Reader (Associate Professor) at the Digital Imaging Research Centre at Kingston University UK and also its Director. His research interests include computer vision for crowd and pedestrian monitoring and personal security as well as distributed visual surveillance systems. Dr Velastin is a member of the IEEE and the British Machine Vision Association (BMVA).

Li-Qun Xu is a Principal Researcher in Visual computing and Multimedia Research of BT Group Research and Venturing Division. He joined BT in 1996 and has since managed and initiated a broad range of research projects in BT's drive for excellence and business growth in managed networked ICT services. His technical expertise includes dynamic visual scene understanding, video analytics for security and safety, human-centred ambient intelligence, and multimedia indexing and retrieval. He holds over 16 European and international patents and pending applications. He serves as independent expert for EU Commission Services for the last five years, participating in project review and proposal evaluation in the diverse areas of networked media, intelligent content and security monitoring. He is an executive team member of IET visual information engineering professional network and expert group member of Transport for London (TfL) Realtime Integration Programme. Prior to his joining BT he worked in Chinese Academy of Sciences and several UK Universities as research and faculty member for 8 years.

The Analysis of Crowd Dynamics: From Observations to Modelling

B. Zhan, P. Remagnino, D.N. Monekosso and S. Velastin

Abstract Crowd is a familiar phenomenon studied in a variety of research disciplines including sociology, civil engineering and physics. Over the last two decades computer vision has become increasingly interested in studying crowds and their dynamics: because the phenomenon is of great scientific interest, it offers new computational challenges and because of a rapid increase in video surveillance technology deployed in public and private spaces. In this chapter computer vision techniques, combined with statistical methods and nerual network, are used to automatically observe, measure and learn crowd dynamics. The problem is studied to offer methods to measure crowd dynamics and model the complex movements of a crowd. The refined matching of local descriptors is used to measure crowd motion and statical analysis and a kind of neural network, self-organizing maps were employed to learn crowd dynamics models.

1 Introduction

We are interested in devising methods to measure and model automatically the crowd phenomenon. Crowded public places are increasingly monitored by security and safety operators. There are companies (for example LEGION) that employed large resources to study the phenomenon and generate realistic simulations: for instance to optimize the flow of people of a public space. Section 2 presents some details about crowd related work, including the applications, research in computer vision and research in other areas like civil engineering and socialogy. The purpose of Section 2 is to give an overview to the state of are on crowd analysis and to in-

Beibei Zhan
Kingston University, Penrhyn Road, Kingston-upon-Thames, Surrey, KT1 2EE, UK. e-mail: B.Zhan@kingston.ac.uk

Acknowledgements This work was partially supported by the British Telecom Group PLC.

vestigate the probability to bridge the research from computer science to areas ike civil engineering and socialogy.

Computer Vision research offers a large number of techniques to extract and combine information of a video sequence acquired to observe a complex scene. The life cycle of a computer vision system includes the acquisition of the monitored scene with one or more homogeneous or heterogeneous cameras, the extraction of features of interest and then the classification of objects, people and their dynamics. In simple scenes the background is extracted with statistical methods and then foreground data and related information are inferred to describe and model the scene. Background is usually defined as stationary data, for instance man made structure, such as buildings, in a typical video surveillance application, or the indoor structure of a building in a safety application, for instance deployed to monitor and safeguard elderly people in a home.

Unfortunately, background modeling becomes rapidly less effective in complex scenes and its usefulness seems to be inversely proportional to the clutter measured in the scene. Figure 1 shows a small experiment testing the effectiveness of background modeling with different types of scenes. Three frames per chosen sequence and the resulting background image built with roughly 1000 frames, are illustrated. The background modeling works well with the first scene; it fails to recover the background of some regions in the second scene because of the frequent occupancy over these regions; and in the third scene, due to the continuous clutter, the background model can be barely recovered. When the monitored scene becomes very cluttered, then one could think of measuring dynamics with optical flow methods, designed to extract information about the dynamics of the scene, typically using gradient information. Unfortunately, popular and conventional optical flow techniques such as Horn and Schunck [36]and Lucas and Kanade [60] also work poorly with heavily crowded scenes. On the other hand, feature based optical flow techniques using multi-resolution work quite well with relatively high frame rate (typically around 25fps) video sequences [15]. Section 3 presents two methods that can automatically measure crowd dynamics. The methods are feature based and employ more sophisticated constraints. They are briefly presented in the chapter and for more details the reader is referred to [98] [100]. Both methods have been assessed with video sequences capturing different types of crowded situations. A comparison of the two methods was carried out and also described in the chapter, for more details the reader should refer to [99]. The performances of both methods produce satisfactory results, even with low frame rate video sequences (typically 4 to 8 fps).

Optical flow or optic flow is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (an eye or a camera) and the scene. In the survey of Beauchemin [11] existing optical flow techniques are investigated, including: 1) differential methods; 2) frequency based methods; 3)correlation based method; 4)multiple motion methods and 5) template refined methods.

Fig. 1 The example frames and the built background images from three different scenes. Left to right: three different scenes; top to bottom, three example frames and the built background images, respectively.

Section 4 describes the methods used to model crowd dynamics. First a statistical method is introduced. This method is focused on defining the main path of the crowded scene [95]. Then a neural network based approach is proposed to capture the crowd dynamics with a reduction of the dimensions of the input data. The self-organizing map technique is employed for this purpose and the results have been generated for different types of crowded scenes. Section 5 discusses the obtained results and sheds some light on the future directions of the work on crowd analysis.

2 Background

The steady population growth, along with the worldwide urbanization, has made the crowd phenomenon more frequent. It is not surprising; therefore, that crowd analysis has received attention from technical and social research disciplines. The crowd phenomenon is of great interest in a large number of applications:

Crowd Management: Crowd analysis can be used for developing crowd management strategies, especially for increasingly more frequent and popular events such as sport matches, large concerts, public demonstrations and so on, to avoid crowd related disasters and insure public safety.

Public Space Design: Crowd analysis can provide guidelines for the design of public spaces, e.g. to make the layout of shopping malls more convenient to costumers or to optimize the space usage of an office.

Virtual Environments: Mathematical models of crowds can be employed in virtual environments to enhance the simulation of crowd phenomena, to enrich the human life experience.

Visual Surveillance: Crowd analysis can be used for automatic detection of anomalies and alarms. Furthermore, the ability to track individuals in a crowd could help the police to catch suspects.

Intelligent Environments: In some intelligent environments which involve large groups of people, crowd analysis is a pre-requisite for assisting the crowd or an individual in the crowd. For example, in a museum deciding how to divert the crowd based on to the patterns of crowd.

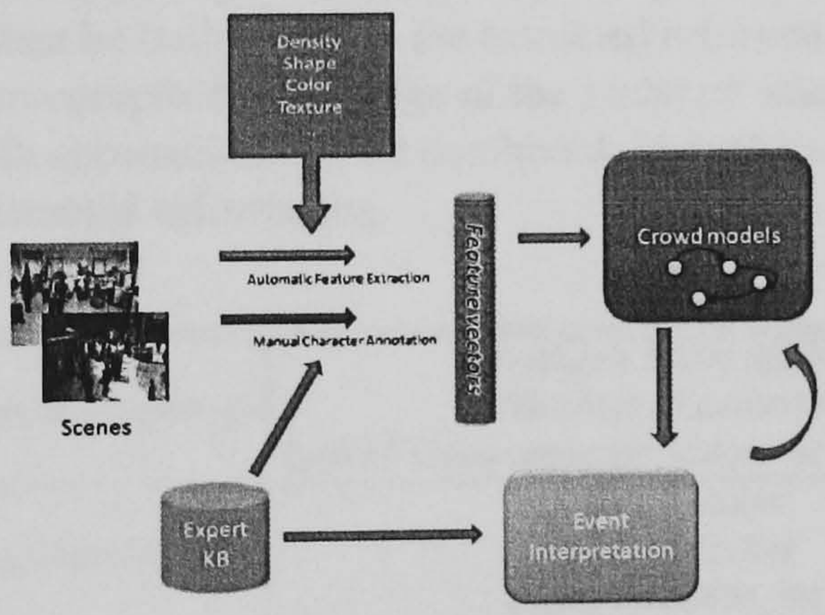


Fig. 2 A framework for Crowd analysis.

Crowd management and public space design are studied by sociologists, psychologists and civil engineers; virtual environments are studied by computer graphic researchers; visual surveillance and intelligent environments are of interest to computer vision researchers. The approach favored by psychology, sociology, civil engineer and computer graphic research is an approach based on human observation

and analysis. Sociologists, for instance, study the characters of a crowd as a social phenomenon, exploring human factors. For example, the computational model developed by Seed Projects at Stanford University [86], incorporated human behaviour in environments with emergency exits. The Crowd - MAGS Project, which is funded by GEOIDE and the Canadian Network of Centers of Excellence in Geomatics, aims to develop micro-simulations of crowd behaviours and the impact of police or military groups [22]. The Police Academy of the Netherlands and School of Psychology of University of Liverpool are cooperating on a project funded by the UK Home Office: "A European study of the interaction between police and crowds of foreign nationals considered to pose a risk to public order" [1].

On the other hand, computational methods such as those employed in computer graphics and vision methods focus on extracting quantitative features and detecting events in crowds, synthesizing the pehnomenon with mathematical and statistical models. For example, early project funded by the EPSRC in the UK were concerned with measuring crowd motion and density and hence potentially dangerous situations [25] [87] [93]. The EU funded project PRISMATICA [75] and ADVISOR [2], completed in 2003, were concerned with the management of public transport networks through CCTV cameras. The UK EPSRC funded project BEHAVE, was concerned with pre - screening of video sequences for the detection of abnormal or crime-oriented behaviour [12]. ISCAPS [42] started in 2005, a consortium of 10 European ICT companies and academic organizations, aims to provide automated surveillance of crowded areas. SERKET, a recently started EU project aims to develop methods to prevent terrorism [40].

Figure 2 illustrates the processes involved in crowd analysis. In a crowd scene the attributes of importance are crowd density, location, speed, etc. This information can be extracted either manually or automatically using computer vision techniques. Crowd models can then be built based on the extracted information. Event discovery is achieved using pre-compiled knowledge of the scene or using the computational model, although both approaches can be combined. In both cases the model is updated with newly extracted information.

Sensor typology and topology	Moving or Static platform Number of cameras Type of video sequence: colour or gray scale, etc.	
Environmental conditions	Indoor/outdoor Level of clutter Light condition, etc.	
Scene typology	Individual characters Appearance, etc.	location/velocity/etc.
	Collective Average speed, etc.	Crowd density

Table 1 Features in crowd analysis by computer vision methods.

2.1 Crowd Information Extraction

The components of crowd analysis from a computer vision perspective are described in Table 1. Essentially, the typology of sensors and their topology influence the scene capture processes; environmental conditions, such as natural and artificial illumination changes often introduce noise; the scene typology affects the type of process one requires to extract the most accurate information of a dynamics scene.

Visual surveillance methods have been developed to estimate motion of objects and people in the scene, in isolation or in groups; a review can be found in [37]. When video is analysed for very crowded scenes, conventional computer vision methods are not appropriate, in these cases methods must be designed to cope with extreme clutter. Features from conventional image processing are still employed, such as colour, shape and texture etc. However, sophisticated methods have been developed to retrieve crowd information. In the following sections review the existing state of the art will be reviewed.

2.1.1 Density Measurement

An important crowd feature is crowd density and it is natural to think that crowd of different density should receive a different level of attention.

Research methods have been proposed for crowd analysis which employs background removal techniques such like [93] [61] and [26] makes use of examples to map the global shape feature to configurations of humans directly. These works have a typical assumption that the number of foreground pixels are proportional to the number of people, which is only true when there are not serious occlusions between people.

Image processing and pattern recognition techniques are also used for the analysis of the scene to estimate the crowd density. Marana et al. [64] assume that images of low-density crowds tend to present coarse texture, while images of dense crowds tend to present fine textures. Self-organizing neural maps [65] combined with Minkowski fractal dimensions [63] are employed to deduce the crowd density from the texture of the image. The work by Marana is compared in [76] with another method that uses Chebyshev moments. An optimization of performance under different illumination conditions is discussed. Lin et al. [59] present a system that estimates the crowd size through the recognition of the head contour using Haar wavelet transform (HWT) and support vector machines (SVM).

Alternative methods combine several techniques, to achieve more accurate and reliable measurements. For example, in [87], an edge-based technique is integrated with background removal using a Kalman filter. Marana et al. [62] use different methods including Fourier and Fractal analysis and classifiers to estimate the crowd density level. Kong et al. in [52][53] employ background subtraction and edge detection is employed; the work defined the extracted edge orientation and blob size histograms as features. The relationship between the feature histograms and the

number of pedestrian is learned from labelled training data. Obvious more cues may indicate a more accurate solution.

2.1.2 Recognition

Conventional visual surveillance focuses on object detection and tracking. In essence, image processing techniques are employed to extract the chromatic and shape information of the moving objects and the background for detecting and tracking purposes.

For crowd dynamics modelling, detecting and tracking are also important as they provide the location and velocity features of the dynamics. Crowded scenes add a degree of complexity to the conventional detection and tracking problem of single individuals. In the following sections the focus will be on methodologies for crowded situations.

Face is the most discriminating feature of the human body and many researchers try to detect pedestrian through face detection. Majority of the existing research employs supervised learning methods to detect face in crowded situation, for example [85] [58] [43][38].

Pedestrian detection and tracking is a well studied problem in computer vision. Many methods have been proposed, such as using the afore mentioned background removal technique, or combining chromatic and shape information of the tracked pedestrians. The following sections discuss the methods that try to provide a solution for pedestrian detection in crowded scenes.

Occlusion caused by the high clutter of the pedestrian in crowd scene is the major challenge for crowd detection problem. Research is carried out to addresses the problem by using human body parts like [91] [28] [57]. Besides conventional cues of pedestrian appearance, space-temporal cues are used for detection. Brostow et al. [17] tackle the problem by tracking simple image features and probabilistically grouping them into clusters representing independently moving entities. In extremely cluttered scenes, individual pedestrian cannot be properly segmented in the image. However sometimes the *crowd* within which the pedestrians share a similar purpose can be recognized. Reisman et al. [79] propose a scheme that uses slices in the spatial-temporal domain to detect inward motion as well as intersections between multiple moving objects. The system calculates a probability distribution function for left and right inward motion and uses these probability distribution functions to infer a decision for crowd detection.

2.1.3 Tracking

Tracking has been proposed to localize the interested object in time-space. Also the velocity feature can be derived afterwards. Though as a natural extension of detection, tracking has its own problem to recognize and identify pedestrians in the consecutive frames. Tracking could be regarded as the most popular topic in

visual surveillance, however currently for crowd analysis, most of the techniques are validated only for multiple (e.g. up to 10) people.

As discussed in the last subsection, occlusions could occur very frequently when there are many objects and people in the scene. Tracking techniques have to overcome the problem in order to continuously track before, during and after the occurrence of occlusions. A comprehensive review on occlusion handling can be found in [30]. A formulation of the occlusion problem is provided, and the techniques are divided in two groups: merge-split approach, which addresses the problem to re-establish object identities following a split, and straight-through approaches, which maintains object identities at all times.

Crowd scenes increase the complexity of tracking because there are multiple moving objects in the scene. Different techniques r are developed to improve the continuous tracking of an individual in a crowd.

- **Likelihood.**Colour, edge etc. are the most popular features in tracking. In crowd salient traceable image features are particular interested for tracking. For example, as one of the good candidates, interest points (IPs) are employed in [30] and [67].
- **Human body model.**Methods using models of human bodies or human body parts have been developed for tracking in complex crowded scenes, which are usually completed with probabilistic frameworks, examples like Zhao [101][102] [92] [91].
- **Tracking inference strategies.** Tracking inference strategies have been developed for the problem of tracking multiple objects. For non-linear and non-Gaussian dynamic models, particle filter technique ,also known as CONDENSATION [41],is one of the most popular among those. Particle filters are sequential Monte Carlo methods based upon a point mass (or 'particle') representations of probability densities [27]. Large portion of multiple object tracking work have employed this technique, for example [88][73] [80] [18][51] [44].
- **Data assoication.**To address data association problem, there are Multiple Hypotheses Tracker (MHT) and Joint Probabilistic Data Association Filter (JPDAF). MHT tries to keep the track of all the possible hypotheses over time [78].A details summary and a discussion of MHT for multiple target tracking is included in [13].JPDAF computes a Bayesian estimation of correspondence between the different features and the different objects, e.g. [77] [45].

In certain cases, interaction happens frequently in crowded scene. Researchers have shown great interest in studying these interactions to get the new perspectives on tracking techniques. For example both Smith et al. [81] and Khan et al. [47] propose to use Markov Chain Monte Carlo (MCMC) and the particle filter. Some researchers interpret interactions as relationships between pedestrians and a group (pedestrian merging/splitting into groups) [66] [69].

Furthermore, for large public areas the use of a multi-camera system is required to cover most of the monitored areas, for exmaple [70] [20][46] [48].

2.2 Crowd Modelling and Events Inference

Dynamics in public spaces can indeed be recurrent. Crowd information can be better exploited to indicate the status of the crowd so that crowd events can be inferred. Crowd models have been built to represent these statuses, either implicitly or explicitly. On the other hand, some research makes direct use of crowd information instead of building models. In such cases, the events are usually inferred based on some prior knowledge of the properties of the particular scene and the crowd. In this section, crowd models and events inference in computer vision will be presented as well as some crowd models from non vision areas.

2.2.1 Crowd models and crowd events inference in computer vision

In computer vision crowd modelling is achieved based on the extracted information from visual data and normally can be employed in crowd events inference. Meanwhile there are also some approaches attempt to infer events without construction of models.

- **Crowd model as representations of recurrent behaviours.** Zhan et al. [94] [97] [96] propose a crowd model using accumulated motion and foreground (moving objects) information of a crowded scene. A preliminary data mining of the PDF models is given to find the major (most frequent) path of the crowd. Andrade et al. [6][5][7] characterize crowd behaviour by observing the crowd optical flow associated with the crowd and use unsupervised feature extraction to encode normal crowd behaviour.
- **Event inference.** Early work on crowd monitoring and crowd event inference using image processing is reviewed by Davies et al. [25].More recent work like in [14] [68] [24] [23] [19].In these methods especially assumptions of crowd are usually involved, indicating that some prior knowledge is required for events inference.

2.2.2 Crowd models from non vision approach

Computational models aim at describing and predicting the collective effects of crowd behaviour by identifying the relationship between crowd features.

- **Physics inspired models.** Several quantitative factors of crowds and pedestrians are measurable. This fact encourages researchers to look for the mathematical models of crowd dynamics. For example Helbing [34][35][33] proposes social force model based on the social field theory. Hughes [39] describes the crowd by "types" where pedestrians in each type have the same walking habits.
- **Agent based models.** These are qualitative models include employing fuzzy methods to describe the relations of factors and crowd motion instead of pure

mathematical methods. Agent-based models use agents to represent the pedestrian or the crowd, examples like [72][74] [16]. Some work of the agent-based models have already been commercialised, such as the work of Keith Still at Crowd Dynamics Ltd [21] and LEGION international LTD [56], both provide pedestrian simulations for space design and planning, based on agent technology.

- **Cellular automation models.** Another research approach employs the construction of local models, where active area has been virtually divided into cell such as [3] [54].
- **Nature based models.** Some of the models take their inspiration from nature. The emotional ant model [10] extends the psychological information using biologically inspired ant agent as a crowd and Kirchner et al. [49] applying a bionics approach to the cellular automation model.

2.3 Examples of Bridging the Research

Computer simulation can be used to evaluate the developed system's performance. Considering that real visual evidences for abnormal scenarios are rare or unsafe to reproduce in a controllable way, Andrade et al. [4] have developed an approach generating simulations to allow training and validation of computer vision systems applied to crowd monitoring. The simulation is generated by a pedestrian path model and a pedestrian body model. Vu et al. [90] conceive a test framework that generates 3D animations corresponding to behaviours recognised by an interpretation system. In other words, this is a test system for a given interpretation system by generating test animations. Non-vision models can be borrowed for computer vision analysis. Anotonini et al. [8][9] propose a framework using discrete choice model, which is widely used in traffic simulations, for pedestrian dynamics modelling.

The works of non-vision analysis show that all of the factors or information extracted from the real world using computer vision techniques are inter-related. Moreover, they have proposed the probable relationships in their works, which represent the human understanding of crowd dynamics. On the other hand, computer vision techniques have the ability of exploiting the special environmental constraints, which could be applied to calibrate the proposed models. We can claim that it is possible that to develop intelligent systems combining these works with computer vision approaches. The system would be capable of automatically understanding and modelling the crowd behaviours which works at both instantaneous and recurrent level.

3 Measuring Crowd Motion

Algorithms exist to analyze simple scenes, where a few people enter and exit the field of view of the deployed cameras. In such scenes, people and objects are identified and tracked throughout the network of cameras. People and objects, such as vehicles are tracked between frames [1] and their trajectories are also predicted using conventional Kalman filters, or more sophisticated particle filter techniques. We studied algorithms that use refined matching methods exploiting local descriptors to derive the dynamics features instead of providing a conventional *tracking* of pedestrians. The problem with tracking in very cluttered and complex scenes is that matching is not always possible and tracks are frequently lost, creating fragmentation in the tracking process. What we propose is the tracking for short periods of time and we provide two algorithms to provide robust matching between frames for use in short-time tracking. The extracted and matched dynamics features can then be directly used in the process of crowd understanding and dynamics modeling.

3.1 Method 1: Pyramid-based Interest Points Topological Matching

In order to devise algorithms to automatically derive complex crowd dynamics, local descriptors, classified as interest points, have been extracted using color gradient information at scale space. Furthermore, besides the use of the extracted descriptors, an advanced matching improved by incorporating topological constraints has been developed.

3.1.1 Extraction of Local descriptor: Harris Detector

The first method employs a modified version of the Harris interest point detector [31]). The Harris interest point detector provides a repeatable and distinctive descriptor of the image features and it is view-point and illumination invariant. The Harris interest point detector provides a repeatable and distinctive descriptor of the image features and it is view-point and illumination invariant. This detector extracts feature points making use of the three chromatic channels defined as M matrix:

$$M = G(\sigma) \otimes \begin{pmatrix} C_x \cdot C_x & C_x \cdot C_y \\ C_y \cdot C_x & C_y \cdot C_y \end{pmatrix} \tag{1}$$

In the operation the image is firstly smoothed using a standard Gaussian operator (of deviation σ). C_x and C_y are respectively the gradient in x and y directions of the pixel chromatic triplet. They are estimated by applying the Gaussian derivative operator $G(\sigma)$ of (deviation σ) to the smoothed image, this is efficiently implemented by

[1] Tracking refers to matching and predicting position and form of extracted features between time frames.

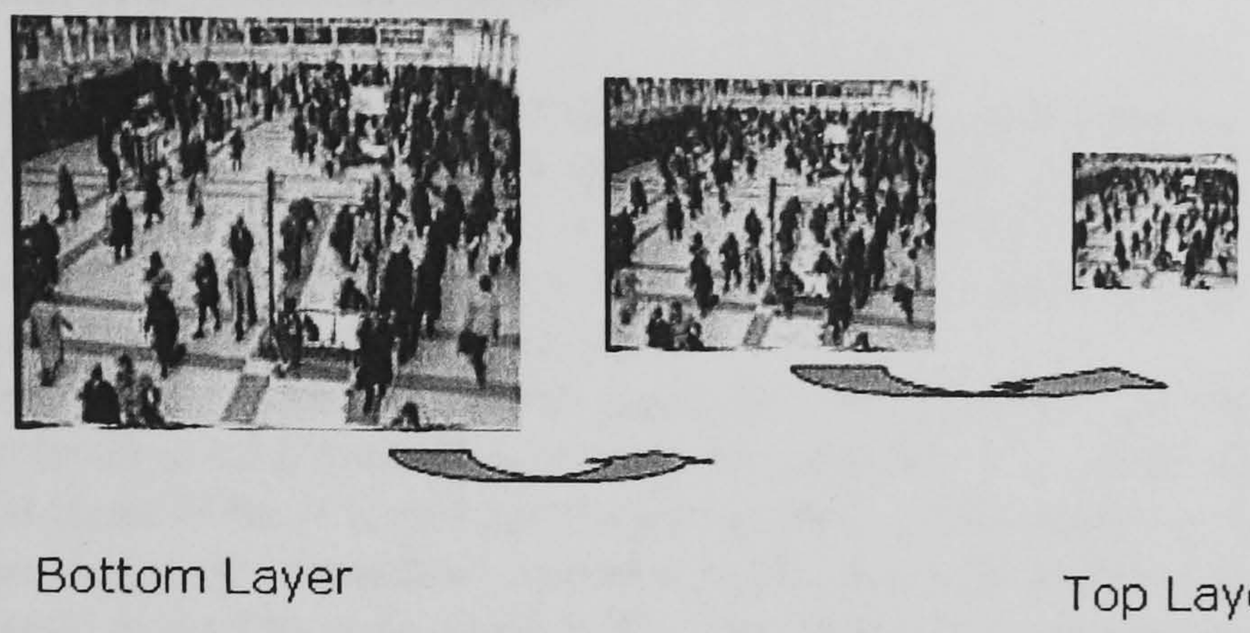


Fig. 3 Interest Point Generation, from bottom layer to top layer

using the method from [89]. The interest points are then extracted using term R, which is calculated as a combination of the Eigen values of the M matrix:

$$R = det(M) + \kappa trace^2(M) \tag{2}$$

Where κ is a constant where $00.4 \leq \kappa \leq 0.06$. The points with local maximum are selected as interest points. A multi-scale approach is used, generating the interest points at the lowest (finest scale) layer and then projecting them up to the top (coarsest scale) layer of the generated pyramid.

3.1.2 Point Matching

The matching is carried out in two steps: searching for the candidate matching points by similarity and then applying the topological constrains described later. Frequent occlusions reduce the probability of identifying correct matches, as a result without local support, similar gradient local regions might be found as plausible matches generating false positives. In this implementation proposes a topological constraint to make the search for correspondences more robust. Gabriel [29] proposed a similar method using topological information, however in his algorithm the area (object) of interest was predefined and the topological information was evaluated by the already know center of the object. In this approach, instead of detail tracking a particular object over long period, the motion of two consecutive frames is more desirable. Therefore the necessary local support is derived from local windows centered at the interest point and the relative location of the interest points in such windows is used. Support is estimated for the matched interest point pair inside the support window.

3.1.3 Temporal pyramidal analysis

Temporal smoothing and matching is also carried out by comparing a number of N spatial pyramids, corresponding to a specific time window. Thus a spatial-temporal pyramidal analysis of the sequence is generated for a number of frames. Temporal smoothing is employed to enforce time consistency on matches, reducing false alarms generated by unstable interest points.

So matching is carried out in both space and time, starting at the highest level (coarsest level) of each pyramid, searching interest point correspondences between the initial frame of the N frames and each other frame within the given time period (corresponding to $N-1$ matches). Spatial matching works from the top (finest scale) of a pyramid to the bottom (coarsest level). Then temporal integration of pyramidal matches of interest point j in 0^{th} frame can then be applied by combining the N matches.

3.2 Method 2: using Edge Continuity Constrains of Interest Points

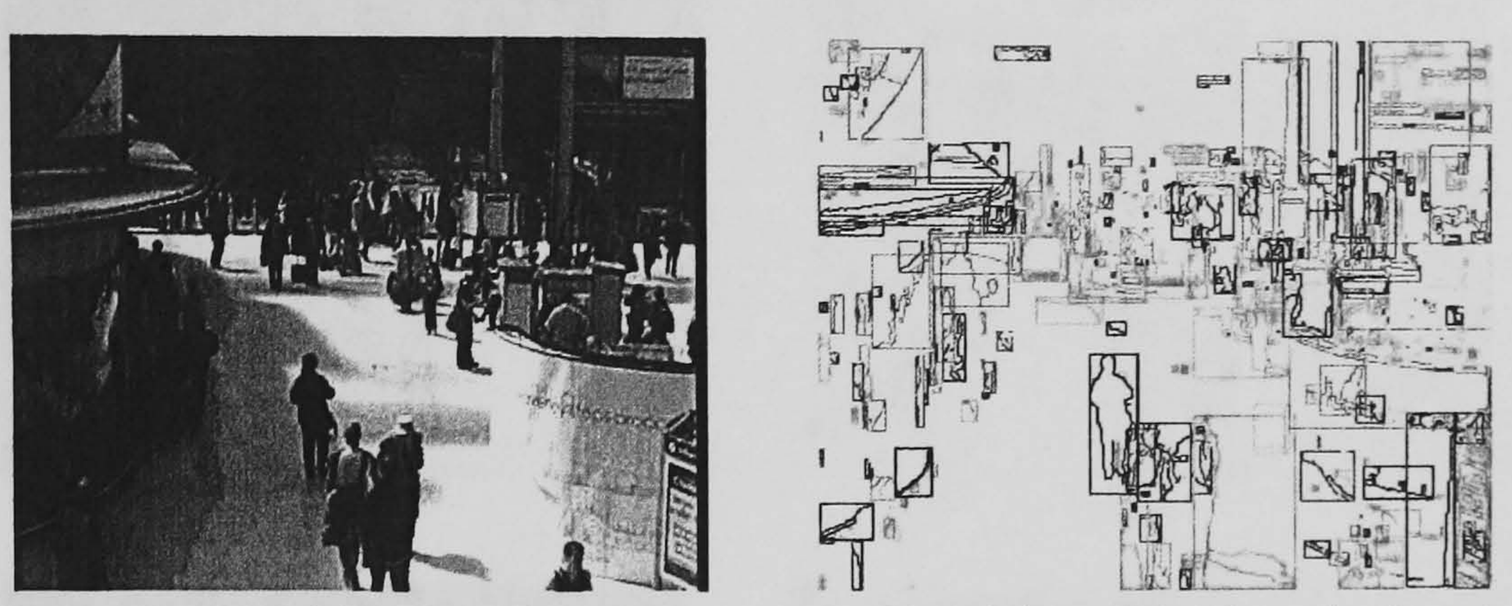
The second method is developed using local descriptors, but also incorporating shape information. Inspired by the methodology using in deformable object tracking, edge information is extracted and descriptor points are extracted as points along an edge with local maximum curvature. The information about an edge is maintained and used to impose the *edgelet constraint* and refine the estimate. Thus the advantage of using point features which are flexible to track and the advantage of using edge features which maintain structural information are combined here.

3.2.1 Edge Retrieval

The Canny edge detector is employed to extract the edge information of a given frame. Each Canny edge is a chain of point, and all the edges are stored in an edge list. Figures. 4 show an example image frame and the extracted edge chains with associated bounding boxes respectively. It can be observed that even in a scene which depicts a crowd of moderate density, edge chains can occlude each other, increasing the descriptor matching complexity.

Canny edge detector is an approach which is optimal for step edge corrupted by white noise. The optimality of the detector is related to three criteria. The detection criterion is about low error rate. It is important that edges occuring in images should not be missed and that there be no responses to non-edges. The second criterion is that the edge points be well localized. The distance between the edge pixels as found by the detector and the actual edge is to

be at a minimum. A third criterion is to minimize multiple responses to a single edge. Thus based on these criteria, Canny edge detector is proposed and become one of the most popular edge detectors [82].



(a) Original frame. (b) Edge chains and their bounding boxes).

Fig. 4 Edge Chain

3.2.2 Curvature Estimation and Interest Point Extraction

Interest points, can be quickly extracted for a sequence of frames, for instance with the Harris corner operator used in last section. However Harris interest points can only represent the local characteristics of an image in isolation, while the shape information of the moving person/people is lost. In this implementation the interest points are from the edges and then the constraint is imposed that they lie on a specific edge. Each edge can be represented by a parameterized curve:

$$x = x(t), \tag{3}$$

$$y = y(t). \tag{4}$$

The curve is smoothed with a Gaussian filter, as follows

$$X(t) = G(t) \otimes x(t), \tag{5}$$

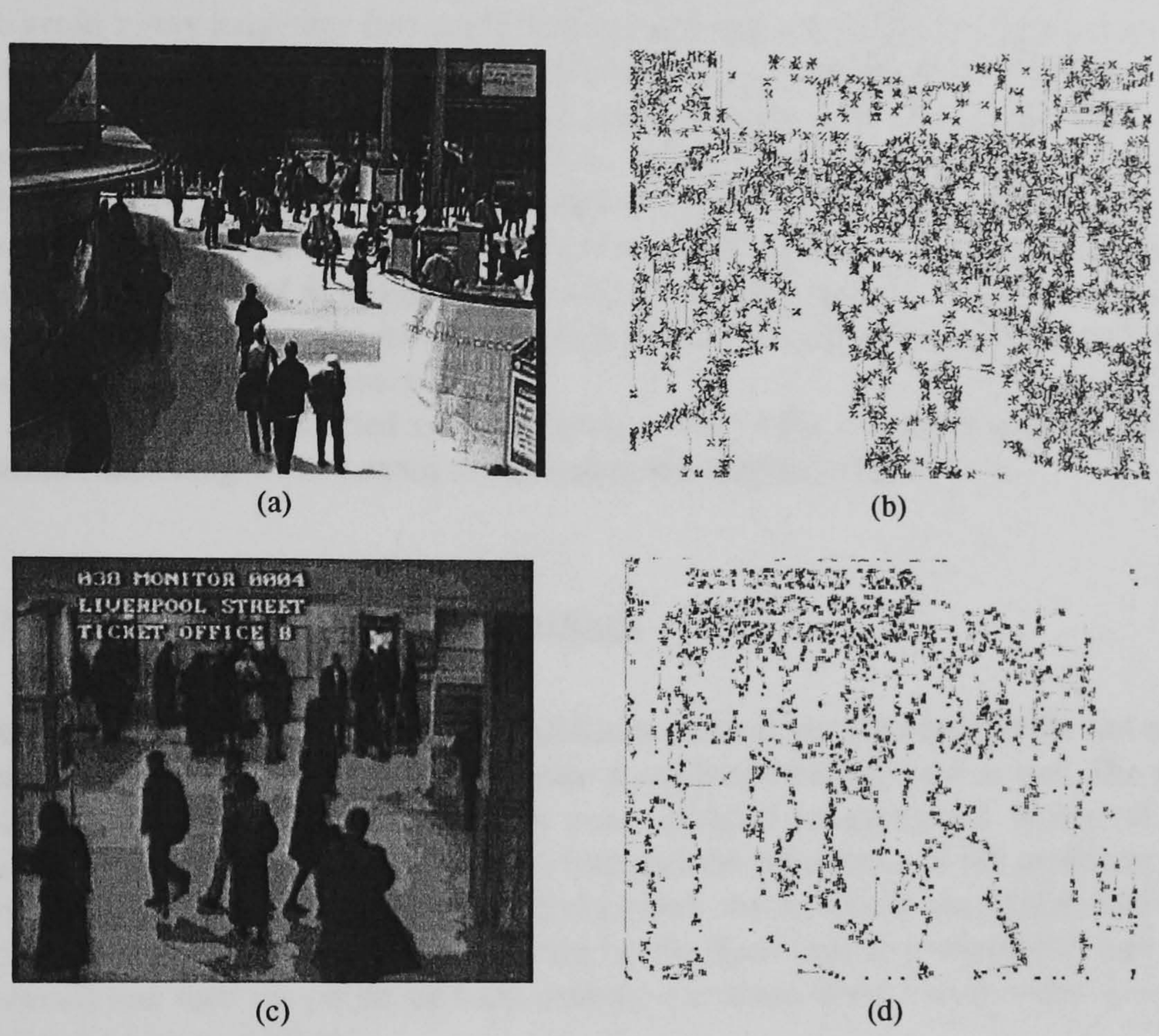
$$X'(t) = G'(t) \otimes x(t), \tag{6}$$

$$X''(t) = G''(t) \otimes x(t). \tag{7}$$

The curvature of each edgelet can then be given by [71] :

$$\kappa = \frac{X'Y'' - Y'X''}{(X'^2 + Y'^2)^{\frac{3}{2}}} \tag{8}$$

Corner points are defined and extracted as the local maxima of the absolute value of curvature on each edge. Thus the edge representation is changed from a point sequence to a corner point sequence, resulting in a list of corner point sequences for all the edges of the image.



(a) (b) (c) (d)

Fig. 5 Two scenes of different complexity levels are illustrated. The original frames (left) and the extracted corner points (right) which are marked with red crosses on grey edges.

3.2.3 Point Matching and the Edgelet Constraint

Given two consecutive frames I_t and I_{t+1}, the motion is estimated for each extracted point of interest. For each corner point with coordinate (x,y) in I_t a rectangular search window is defined centering at (x,y)in I_{t+1}. A look-up table (LUT) contains corner point and edge information is generated to enhance the matching. The correspondence is matched by using curvature information of corner points in the search window in LUT against the reference point. The error is calculated by the curvature.

Complex dynamics and frequent occlusions generated in crowd scenes make the estimation of motion a very complex task. Point matching in isolation is too fragile

and prone to errors to provide a good motion estimator. If the interest points are extracted on edge chains, then the edge constraint can be imposed and used.

For an image frame I_t, every edgelet is split to a uniform length edgelets represented by sub-sequences (so called edgelet). There are two reasons for doing this: to avoid a very long edge that could be generated by several different objects, and to enhance the matching of the edge fragments that are generated by occlusions. For each corner point there are n candidate matching points. Each candidate point belongs to an edgelet, thus there are $m(m <= n)$ candidate matching edgelets. To find the best match, three parameters are used: energy cost, variation of displacements and the match length for each candidate and combine them into a single matching score. The length of th edgelet is assumed to be small enough so that it would not split again to two or more matches. This is so that their candidate points correspond to the same candidate sequence.

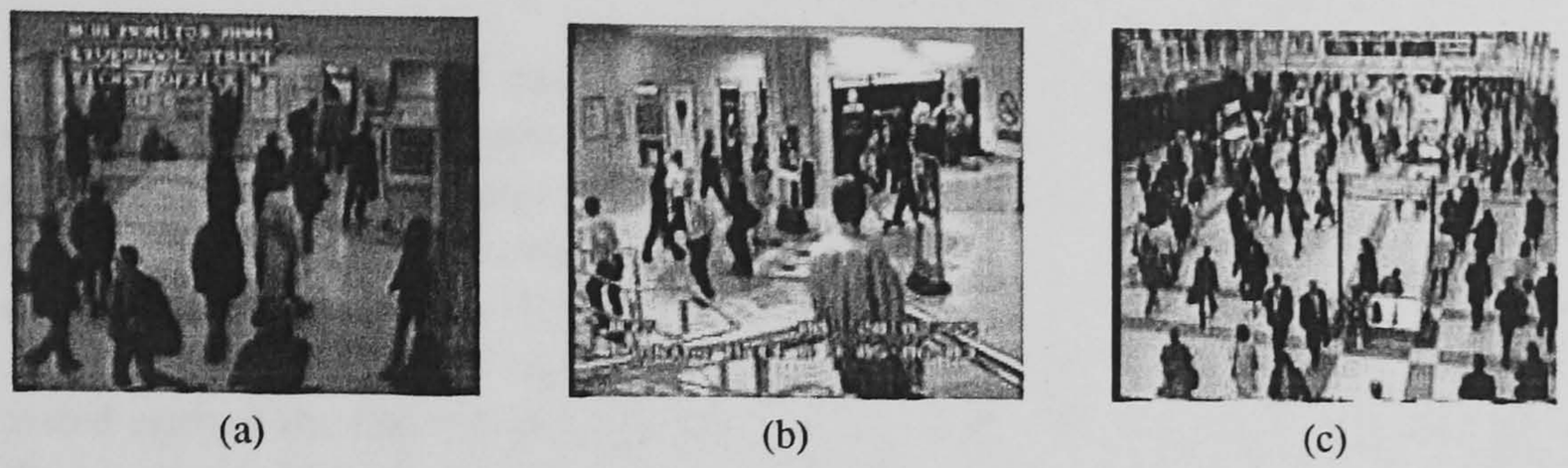
The matching is carried out over every point of the interested edgelet and an overall matching will be exam to determine the matched edgelet.

3.3 Comparison of the two methods

When the scene is very complex, occlusions make it virtually impossible not only to track individuals but also to estimate a stochastic background model. The two described motion estimation methods were validated and compared. In both of the algorithms, constraints are applied to improve the robustness of the matching between individual descriptors. The first algorithm checks locally the spatial temporal consistency of color gradient supported by the local topology constraints and the second one uses the points of local extreme curvature along Canny edges and applies contour constraints.

3.3.1 Testing Data

The two motion estimation algorithms are tested using three sequences taken from crowded public space and quantitative results are generated. In the following a brief description of the test dataset used in the experiments is given. Then the details of the testing methods adopted and explain the results generated from the tests are introduced. Again additional visual results are included at the end of the section. Sample frames from the three sequences are shown in Figure 6: sequence 1 (left) is a mid field scene with people scattered across the field of view; sequence 2 (middle) is a mid field scene with major motions taking place in certain areas; sequence 3 (right) is a far field scene with pedestrians present in all parts of the field of view, with some predominant trajectories.



(a) (b) (c)

Fig. 6 Sample frames from 3 testing sequences

3.3.2 Testing based on local descriptors

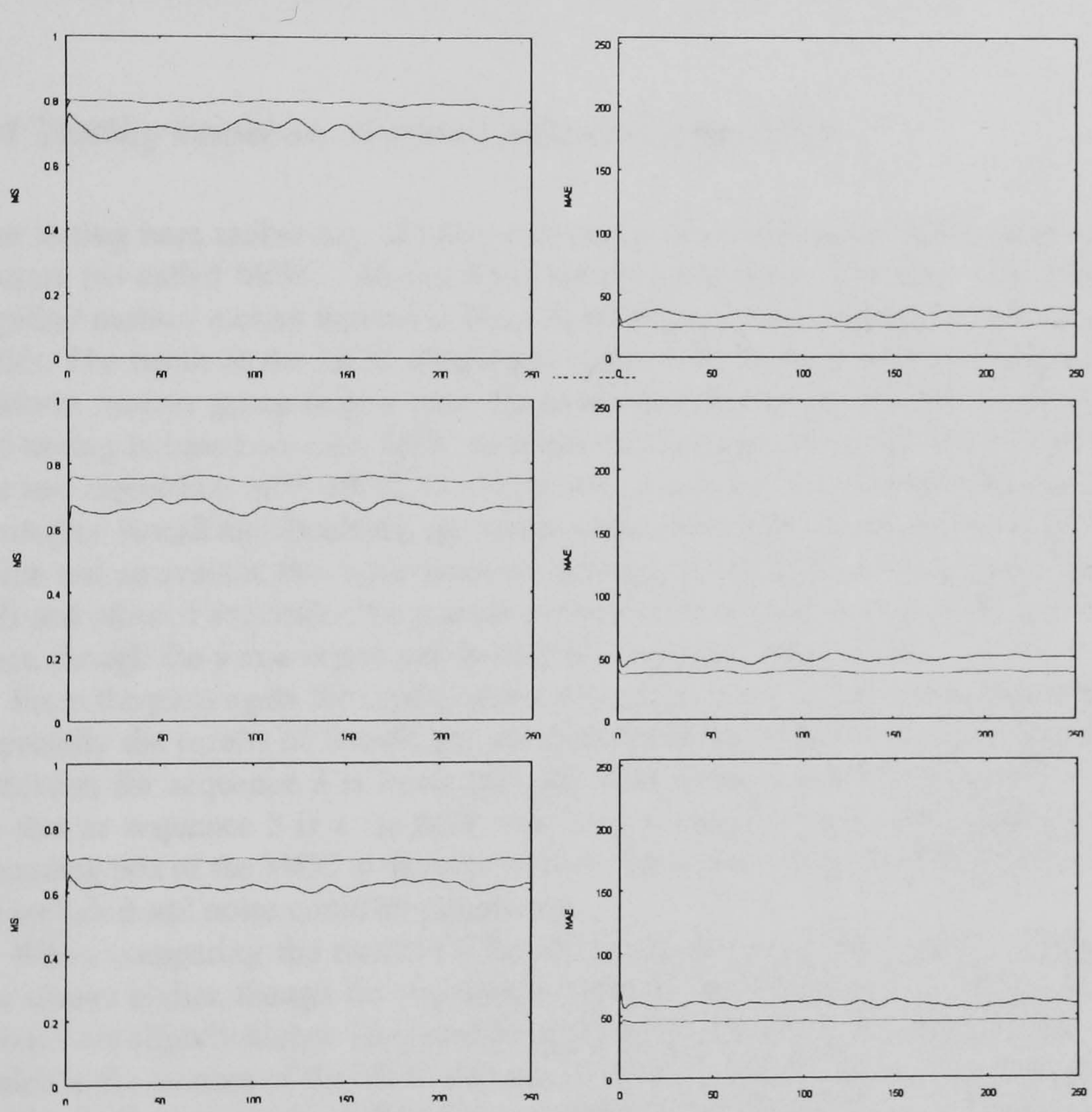


Fig. 7 MS (left column) and MAE (right column) along time for the 3 testing sequence (From top to the bottom: sequence 1, sequence 2 and sequence 3), red lines for Algorithm 1; green lines for Algorithm 2. Algorithm 2 keeps higher in MS and lower in MAE.

In this testing only the quality of matching of individual local descriptors is considered. For each pair of consecutive frames, local descriptors in the initial frame are compared with their corresponding local descriptors, found by the two presented algorithms, in the target/second frame, respectively. Two measures, Mean Similarity (MS) and Mean Absolute Error (MAE), are used here.

The images in Figure 7 represent the plots of MS and MAE for the two algorithms tested against the three sequences. MS and MAE are calculated every frame along the sequence. In each plot the x axis represents time (the number of the frame) and the y axis represents the values of MS and MAE, respectively. Hence, for the two algorithms the MS and MAE for the three testing sequences are both good, though in most of the cases the second algorithm has a higher MS and a lower MAE. Also, along the time scale the MS and the MAE produced from the first algorithm fluctuate a lot while the second one produces more stable results. It can be concluded that the second algorithm has a more desirable performances than the first one.

3.4 Testing based on Motion Connect Component

The testing here makes use of connected components algorithm based on motion vectors (so called MCC – Motion Connected Component). The algorithm groups together motion vectors that are in close proximity and have common motion properties. The result of the MCC algorithm segments the motion field into clusters of uniform motion group (e.g. a (part of) pedestrian or a group of pedestrians), and the testing is based on each MCC to assess the two algorithms. In order to assess the two algorithms with MCC, two measures, which are from in evaluating search strategies: Recall and Precision, are adapted here. For every frame an average Recall value and an average Precision value are calculated. Figure 8 gives the plots of Recall and plots of Precision; the layouts of these plots remain similar to the previous ones, though the y axis represents Recall and Precision, respectively.

From the plots again the results of Recall and Precision of both of the algorithms, especially the results of Recall, are satisfied. It can be observed that the results of Precision for sequence 3 is lower than the other three, one possible reason could be that as sequence 3 is a far field view for a crowded scene, when mapping the bounding box of the MCC to the second frame local descriptors of other MCC could be included and noise could be introduced.

When comparing the results of Recall, it can be seen values for Algorithm 2 are always higher, though for sequence 2 and sequence 3 Precision values for Algorithm 1 are slightly higher. Here another measure should be taken into consideration, which is the number of the MCC detected by each algorithm. According to the plots in Figure 9, in sequence 1 the average number of MCC detected by Algorithm 1 is around 20, while by Algorithm 2 the number is around 100; in sequence 2, the numbers are around 20 and 200, respectively; in sequence 3 the numbers are around 40 and 280, respectively. Algorithm 2 detects much more MCC, especially for sequence 2 and 3. Due to the above fact and the fact Algorithm 2 produces higher

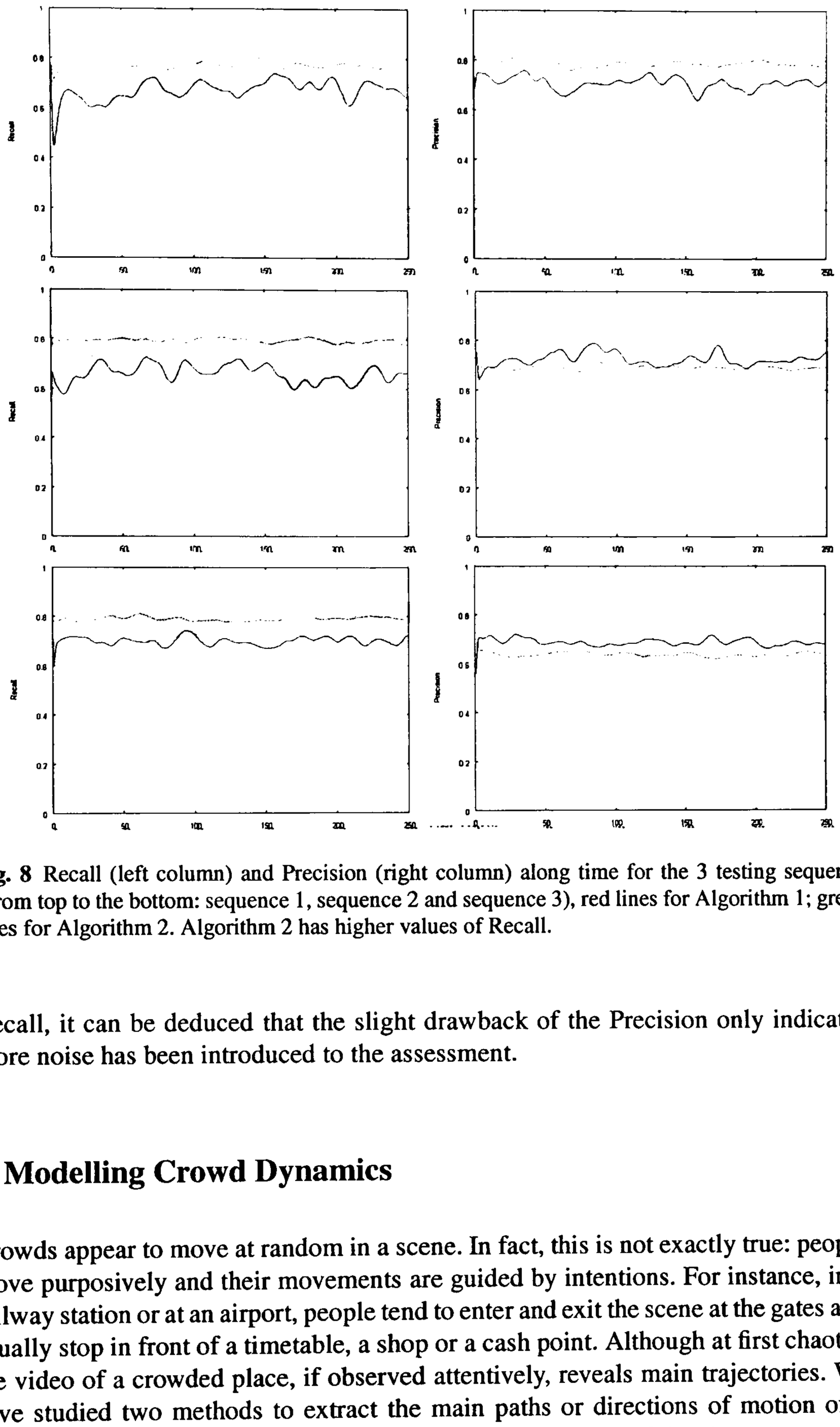


Fig. 8 Recall (left column) and Precision (right column) along time for the 3 testing sequence (From top to the bottom: sequence 1, sequence 2 and sequence 3), red lines for Algorithm 1; green lines for Algorithm 2. Algorithm 2 has higher values of Recall.

Recall, it can be deduced that the slight drawback of the Precision only indicates more noise has been introduced to the assessment.

4 Modelling Crowd Dynamics

Crowds appear to move at random in a scene. In fact, this is not exactly true: people move purposively and their movements are guided by intentions. For instance, in a railway station or at an airport, people tend to enter and exit the scene at the gates and usually stop in front of a timetable, a shop or a cash point. Although at first chaotic, the video of a crowded place, if observed attentively, reveals main trajectories. We have studied two methods to extract the main paths or directions of motion of a crowded scene. They are described in the following sections of the chapter.

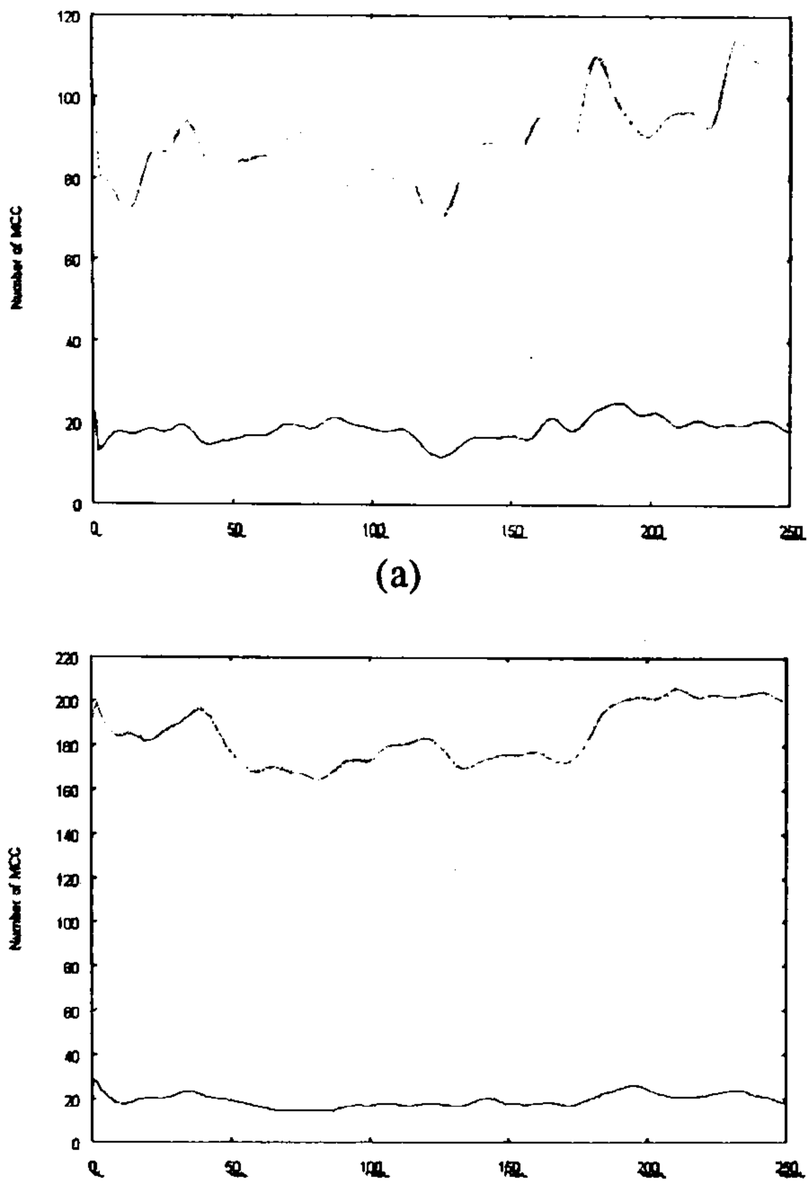


Fig. 9 Number of MCCs along time for the 3 testing sequence, red lines for Algorithm 1; green lines for Algorithm 2 (From top to bottom: sequence 1, sequence 2 and sequence 3). Algorithm 3 detects much more MCCs for all of the three video sequences.

4.1 Statistical Analysis

The proposed method can be summarized in the following steps:

- Occurrence PDF: foreground detection, connected components, accumulator,
- Orientation PDF: correlation matrix, accumulator of block matching,

- Path discovery: previous orientation, probability calculation, path split.

4.1.1 Occurrence PDF

It is unrealistic to precompile a background model of a complex real world scene, such as those video recorded by security cameras in public spaces. This is because of sudden or continuous changes in illumination, shadows and noise in the video signals. This method assumes that the scene is not too crowded and the Gaussian mixture model [83] is used to build a robust model of the background of the scene. The foreground data is further processed to reduce noise. In particular, connected components have been implemented. Connectivity of foreground pixels gives more robustness to the foreground data and assures that only large foreground blobs are accepted for further analysis, while smaller blobs are rejected as likely noise.

> Background model is essential for video analysis to separate foreground data from the scene. There is a standard background adaptation carried out by averaging images over time, creating a background approximation which is similar to the current static scene except where motion occurs. While this is effective in situations where objects move continuously and the background is visible a significant portion of the time, it is not robust to scenes with many moving objects particularly if they move slowly. Gaussian mixture model is proposed in [83]. It is an adaptive tracking system that is flexible enough to handle variations in lighting, moving scene clutter, multiple moving objects and other kinds of changes to the observed scene. Rather than explicitly modeling the values of all the pixels as one particular type of distribution, the values of a particular pixel are simply modelled as a mixture of Gaussians. Based on the persistence and the variance of each of the Gaussians of the mixture, the model determines which Gaussians may correspond to background colors. Pixel values that do not fit the background distributions are considered foreground until there is a Gaussian that includes them with sufficient, consistent evidence supporting it.

For each frame foreground, features are accumulated for every pixel, so that after a relatively long video sequence the accumulator of the foreground occurrence throughout the whole image will have some information.

4.1.2 Orientation PDF

The image plane is segmented into a regular grid of cells ($N \times M$). The dimension of each cell is a multiple of 2 and each cell is square-shaped ($K \times K$). The idea is to speed up the matching process employed as a coarse estimator of motion between frames. Motion is estimated between consecutive frames, using the

foreground blocks of the first frame as a reference/template and searching for an optimal match in the second frame. In the current implementation, block matching is carried out in a $3x3$ neighborhood, around the selected foreground cell. A cell is labeled as foreground if the majority of its pixels are indeed foreground. Matching performance is improved by matching only between foreground cells, ignoring background cells.

Each cell is therefore associated with a histogram representing the eight possible directions of motion. The intention here is to build a local representation of motion, similar to a discrete reinforcement learning technique [84], where each cell of the table has associated a quality array, indicating the likelihood of transition from the current cell to a neighboring cell. The final outcome is an orientation PDF, which could be interpreted as the global optical flow of the scene.

4.2 Path Discovery

The work described in the previous sections provides two PDFs: one for the occurrence and one for the orientation of a scene. To discover the main paths, the information and extract those corresponding to higher likelihood/probability need to be combined. Ideally the paths are identified corresponding to the modes of a probability density function that combines both occurrence and orientation information.

In order to estimate the main paths make a number of assumptions was made.

Path origin: The assumption is that all paths originate from the boundaries of the scene. Consequently path discovery is started from a cell the boundary of the scene and having high occurrence probability. This assumption would not work if the scene had an entrance or exit in the middle of the image, but this can be overcome relatively easily by using user-defined boundaries.

Graceful continuation/Smooth trajectory: As observed, the paths have a high probability to maintain their orientation (e.g. people are more likely to go on a straight line, and seldom go backwards.) So the expected direction of motion is modeled with a Poisson distribution with its maximum in the neighboring cell along the current direction of motion.

The idea is to spread the likelihood of change in direction unevenly, maintaining the previous orientation as the one at highest probability and forcing the other directions (change in direction) to have a lower likelihood. From the start point, the probability is calculated for each neighboring block using the occurrence PDF (PDFocc), the block matching accumulator (P_b) and the orientation probability (PDF_{or}). Furthermore, to avoid repeating calculations from the same block, the visited cells is marked, their probability is set to to 0 each time the path discovery process has to deal with them.

The process will follow the highest probability block. Also a way of deciding when to split a trajectory in two or more sub-trajectories is devised. This technique works on a threshold that estimates whether two or more paths are viable given their

associated likelihood. However, as not to generate too many branches, only a single split along a trajectory is admitted.

Once all paths are identified, a fitting process takes place. This serves two purposes: (i) to have a compact representation of the path, (ii) to have a faster way of estimating the distance between a blob/bounding rectangle, identified by new foreground data, and the spline, and consequently estimating an error. The final path is represented as a curve by fitting a uniform Cubic B-spline.

4.3 Self-Organizing Map for Learning Crowd dynamics

The previous approach is based on background modeling, which can not work properly under extremely crowded situations. The crowd PDFs derived by the described method are not global statistics. Also, the number of dimensions of the model is relatively high, especially for the orientation PDF. Those are disadvantages can be overcome by the method described in this section.

Here we describe some work carried out applying self-organized maps to learn the dominant crowd dynamics. The self-organized map (SOM) model [32] is a well known dimensionality reduction method proved to bear resemblance with some features of the human brain, which represent different sensory input by topologically ordered computational maps. SOMs are widely used in mapping multidimensional data onto a low-dimensional map, example of applications include the analysis of banking data, linguistic data [50] and image classification [55]. This section proposes a system learning the crowd dynamics with the SOM. The system uses dynamics information as input; and it generates SOM which captures the dominant recurrent dynamics.

4.3.1 Building SOM for a Crowded Scene

The most common SOMs have neurons organized as nodes in a one- or two-dimensional lattice. The neurons of a SOM are activated by input patterns in the course of a competitive learning process. At any moment in time only one output neuron is active, the so called winning neuron. Input patterns are from a n-dimensional input space and are then mapped to the one- or two- dimensional output space of the SOM. Every neuron has a weight vector which belongs to the input space.

The desirable SOM in this application should capture the two major components of the crowd dynamics: occurrence and orientation. Thus a four dimensional input space is chosen to be the weight space of the SOM, which can be represented as $f : (x, y, \theta, \rho)$. Each data from the input space can be explained as the location where crowd moves and the motion vectors in the form of angle (θ) and magnitude (ρ). The SOM used in this experiment is organized in a two-dimensional space and represented by a square lattice.

There are two phases for tuning the SOM with an input pattern I, competing and updating. In the competing phase every neuron is compared with I; the similarly of I and the weights of all of the neurons are computed; and the neuron $N(i_w, j_w)$(denoted by the neuron's coordinates of the lattice) with highest similarity is selected as the winning neuron. In the update phase, for each neuron $N(i, j)$, a distance is calculated as:

$$d^2 = (i - i_w)^2 + (j - j_w)^2 \tag{9}$$

the topological neighborhood function is then defined as:

$$h(n) = \exp(-\frac{d^2}{2\sigma^2(n)}) \tag{10}$$

where n denotes the time, which can also be explained as the number of iterations. and $\sigma^2(n)$ decreases with the time. The weight of each neuron $N(i, j)$ at time $n+1$ is then defined by:

$$w(n+1) = w(n) + \eta(n)h(n)(x - w(n)) \tag{11}$$

where $w(n)$ and w(n+1) is the weight of the neuron at time n and $n+1$. $\eta(n)$ is the function of learning rate, which always decreases with time.

4.3.2 Visualization

Figure 10 illustrates three different video sequences with different dynamics. These video sequences have been input into the system, and Figure 11 shows the output SOMs. In the figure SOMs are visualized in the input space, i.e. showing the weight vector of each neuron. In the visualization, the color arrows and their locations are from the weight vector of neurons; the location of the arrows are from the first two components of the weight vectors (x,y), and the arrows show the second two components - the components of motion (θ,ρ). The different colors of the arrows are also indicating the different orientation of the motion.

In the first video (the left column in Figure 10) the major crowd is moving from bottom left to top right of the scene. There is another crowd flow from bottom right of the scene which joins the major flow. In its SOM (the first one in Figure 11) the neurons with green arrows are clearly from the major flow and the ones with red and purple arrows are from the minor flow. In the second video (the middle column in Figure 10 it is an area of an entrance to a public space. So most of the people move from top to bottom of the scene. The crowd in the upper part of the scene is more sparse and moves faster when compared to the crowd in the lower part of the scene. There is also a minor flow, which joins the major flow from right of the scene. In the built SOM (the second SOM in Figure 11), again the flows are clearly indicated. Furthermore the SOM takes an umbrella shape, which represents the shape of the flow constrained by the obstacles in the scene. In the third video (the right column in

Figure 10) the scene is a large open area with multiple crowd flows. The major flow is moving from right to left; however there are several minor flows, most of which are in the lower part of the scene. Again the SOM (the third in Figure 11)captures the major dynamics and also some minor flows. From the three examples, it can be concluded that the SOMs not only preserves the dominant motion vector, but also represents the shape of the regions with dominant motion of the scenes.

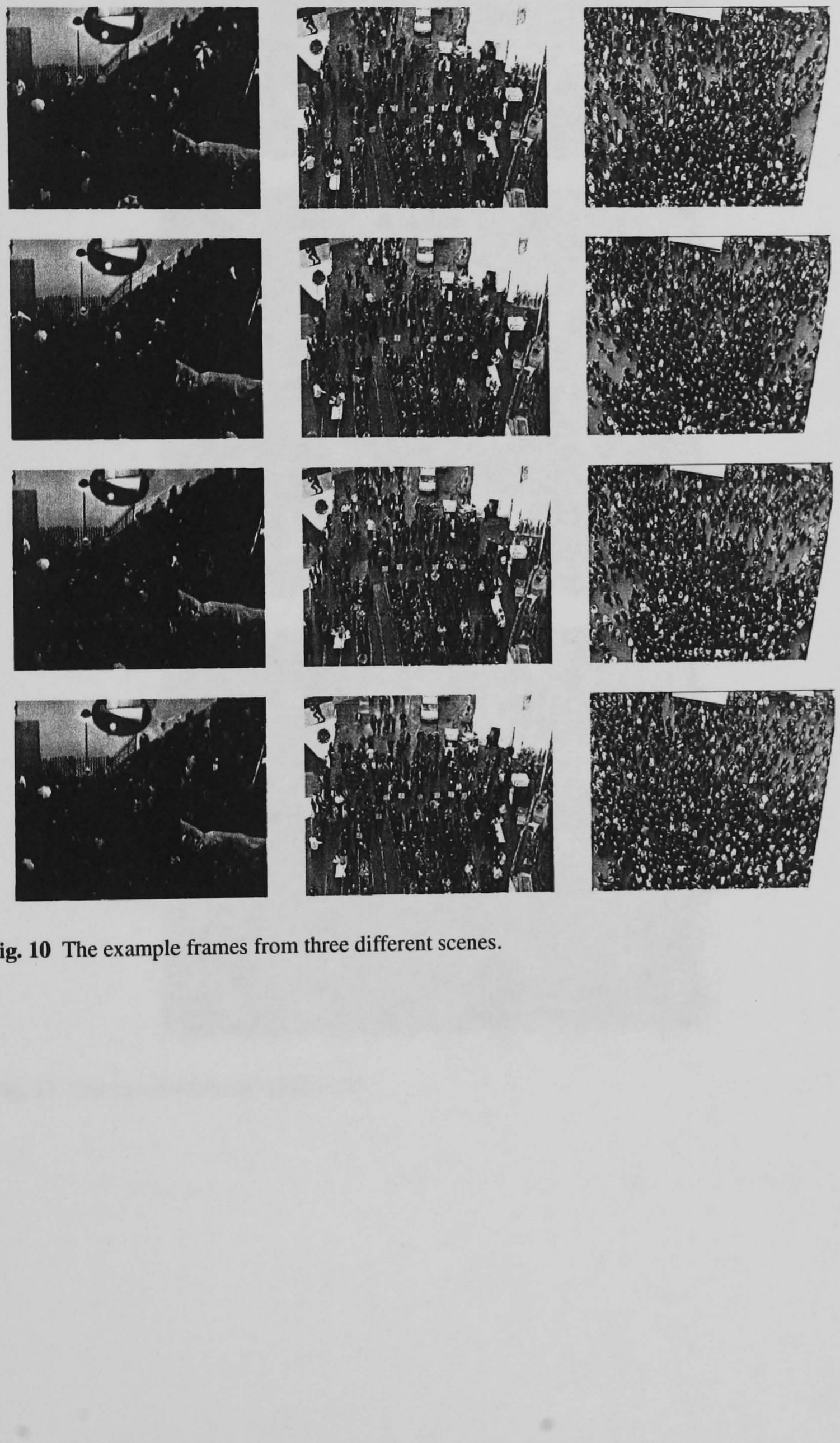


Fig. 10 The example frames from three different scenes.

Fig. 11 The visualization of built SOMs

5 Discussion

This chapter has described novel methods for the automatic analysis of the crowd phenomenon. They are based on computer vision techniques. In particular local descriptors matching with refined constraints are proposed to tackle the problem of crowd motion measurements. Statistical methods using Probability Density Functions are employed to learn the crowd dynamics by mining the main path of the crowded scene. Another approach of crowd dynamics learning adapts Self Organizing Maps to capture the main recurrent dynamics. There are a couple of possible extensions of the work. Especially for latter approach, analyzing the organization of the SOM would make it possible to understand the characters of the dynamics. Also the development of a metric of comparing SOMs could be very useful to enhance the automatic classification of crowded scenes.

References

1. Adang, O.M., Stott, C.: A European study of the interaction between police and crowds of foreign nationals considered to pose a risk to public order. Http://policestudies.homestead.com/Euro2004.html
2. ADVISOR: Http://advisor.matrasi-tls.fr/
3. AEA, Techology: A technical summary of the aea egress code. Technical Report **1** (2002)
4. Andrade, E., Fisher, R.: Simulation of crowd problems for computer vision. In: First International Workshop on Crowd Simulation, vol. 3, pp. 71–80 (2005)
5. Andrade, E., Fisher, R.: Hidden Markov models for optical flow analysis in crowds. In: Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)-Volume 01, pp. 460–463. IEEE Computer Society Washington, DC, USA (2006)
6. Andrade, E., Fisher, R.: Modelling crowd scenes for event detection. In: Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)-Volume 01, pp. 175–178. IEEE Computer Society Washington, DC, USA (2006)
7. Andrade, E.L., Blunsden, S., Fisher, R.B.: Performance analysis of event detection models in crowded scenes. In: Proc. Workshop on Towards Robust Visual Surveillance Techniques and Systems at Visual Information Engineering 2006, pp. 427–432. Bangalore, India (2006)
8. Antonini, G., Bierlaire, M., Weber, M.: Simulation of pedestrian behaviour using a discrete choice model calibrated on actual motion data. In: 4th STRC Swiss Transport Research Conference. Ascona (2004)
9. Antonini, G., Venegas, S., Thiran J.P.and Bierlaire, M.: A discrete choice pedestrian behaviour model in visual tracking systems. In: Advanced Concepts for Intelligent Vision Systems, pp. 273–280. Brussels, Belgium (2004)
10. Banarjee, S., Grosan, C., Abarha, A.: Emotional ant based modeling of crowd dynamics. In: Seventh International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC'05), pp. 279–286 (2005)
11. Beauchemin, S., Barron, J.: The computation of optical flow. ACM Computing Surveys (CSUR) **27**(3), 433–466 (1995)
12. BEHAVE: Http://www.homepages. informatics. ed.ac.uk /rbf /BEHAVE/
13. Blackman, S.: Multiple hypothesis tracking for multiple target tracking. Aerospace and Electronic Systems Magazine, IEEE **19**(1), 5–18 (2004)
14. Boghossian, B., Velastin, S.: Motion-based machine vision techniques for the management of large crowds. In: the 6th IEEE International Conference on Electronics, Circuits and Systems, vol. 2 (1999)

15. Bouguet, J.: Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm. Intel Corporation, Microprocessor Research Labs (2000)
16. Brenner, M., Wijermans, N., Nussle, T., de Boer, B.: Simulating and controlling civilian crowds in robocup rescue. In: inproceedings of RoboCup 2005: Robot Soccer World Cup IX. Osaka (2005)
17. Brostow, G., Cipolla, R.: Unsupervised Bayesian Detection of Independent Motion in Crowds. In: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Volume 1, pp. 594–601. IEEE Computer Society Washington, DC, USA (2006)
18. Cai, Y., de Freitas, N., Little, J.J.: Robust visual tracking for multiple targets. In: European Conference on Computer Vision, *LNCS*, vol. 3954, pp. 107–118. Springer (2006)
19. Chan, M.T., Hoogs, A., Bhotika, R., Perera, A., Schmiederer, J., Doretto, G.: Joint recognition of complex events and track matching. In: CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1615–1622. IEEE Computer Society, Washington, DC, USA (2006). DOI http://dx.doi.org /10.1109 /CVPR. 2006.160
20. Chang, T., Gong, S., Ong, E.: Tracking multiple people under occlusion using multiple cameras. In: British Machine Vision Conference, pp. 566–575 (2000)
21. Crowd, Dynamics: Http://www.crowddynamics.com/
22. Crowd, MAGS: Http://www2.ift.ulaval.ca/ muscamags/Dnd - crowdmags - project.htm
23. Cupillard, F., Bremond, F., Thonnat, M.: Behaviour recognition for individuals, groups of people and crowd. IEE Seminar Digests **7** (2003)
24. Cupillard, F., Bremond, F., Thonnat, M., INRIA, F.: Group behavior recognition with multiple cameras. Applications of Computer Vision, 2002.(WACV 2002). Proceedings. Sixth IEEE Workshop on pp. 177–183 (2002)
25. Davies, A., Yin, J., Velastin, S.: Crowd monitoring using image processing. Electronics & Communication Engineering Journal **7**(1), 37–47 (1995)
26. Dong, L., Parameswaran, V., Ramesh, V., Zoghlami., I.: Fast Crowd Segmentation Using Shape Indexing. Rio de Janeiro, Brazil (2007)
27. Doucet, A., Godsill, S., Andrieu, C.: On sequential Monte Carlo sampling methods for Bayesian filtering (2000)
28. Elgammal, A., Davis, L.: Probabilistic framework for segmenting people under occlusion. In: Eighth IEEE International Conference on Computer Vision, 2001. ICCV 2001. Proceedings., vol. 2, pp. 145–152 (2001)
29. Gabriel, P., Hayet, J., Piater, J., Verly, J.: Object tracking using color interest points. Proceedings. IEEE Conference on Advanced Video and Signal Based Surveillance pp. 159–164 (2005)
30. Gabriel, P., Verly, J., Piater, J., Genon, A.: The state of the art in multiple object tracking under occlusion in video sequences. Advanced Concepts for Intelligent Vision Systems pp. 166–173 (2003)
31. Gouet, V., Boujemaa, N.: About optimal use of color points of interest for content-based image retrieval. Technical Report pp. RP–4439 (2002)
32. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall PTR Upper Saddle River, NJ, USA (1994)
33. Helbing, D., Farkas, I., Vicsek, T.: Simulating Dynamical Features of Escape Panic. Letters to Nature **407**, 487–490 (2000)
34. Helbing, D., Molnár, P.: Social force model for pedestrian dynamics. Physical Review E **51**(5), 4282–4286 (1995)
35. Helbing, D., Molnar, P.: Self-organization phenomena in pedestrian crowds (1997). URL http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/9806152
36. Horn, B., Schunck, B.: Determining Optical Flow. Artificial Intelligence **17**(1-3), 185–203 (1981)
37. Hu, W., Tan, T., Wang, L., Maybank, S.: A survey on visual surveillance of object motion and behaviors. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews **34**(3), 334–352 (2004)

38. Huang, C., Ai, H., Li, Y., Lao, S.: Vector boosting for rotation invariant multi-view face detection. In: Tenth IEEE International Conference on Computer Vision, vol. 1, pp. 446–453 (2005)
39. Hughes, R.: A continuum theory for the flow of pedestrians. Transportation Research Part B: Methodological **36**(6), 507–535 (2002)
40. INRIA: Http://www.inria.fr /rapportsactivite /RA2005 /orion /uid1.html
41. Isard, M., Blake, A.: A mixed-state CONDENSATION tracker with automatic model-switching. In: IEEE International Conference on Computer Vision, pp. 107–112 (1998). Url: citeseer.ist.psu.edu/isard98mixedstate.html
42. ISCAPS: Http://www.iscaps.reading.ac.uk/home.htm
43. Jones, M., Viola, P.: Fast multi-view face detection. Mitsubishi Electric Research Lab TR-20003-96 (2003)
44. Kang, H., Kim, D., Bang, S.: Real-time multiple people tracking using competitive condensation. Proc. of the Intl. Conference on Pattern Recognition **1**, 413–416 (2002)
45. Karlsson, R., Gustafsson, F.: Monte Carlo data association for multiple target tracking. Target Tracking: Algorithms and Applications (Ref. No. 2001/174), IEE **1** (2001)
46. Khan, S.M., Shah, M.: A multiview approach to tracking people in crowded scenes using a planar homography constraint. In: 9th European Conference on Computer Vision, *LNCS*, vol. 3954, pp. 133–146. Springer (2006)
47. Khan, Z., Balch, T., Dellaert, F.: MCMC-based particle filtering for tracking a variable number of interacting targets. IEEE Transactions on Pattern Analysis and Machine Intelligence **27**(11), 1805–1819 (2005)
48. Kim, K., Davis, L.S.: Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering. In: European Conference on Computer Vision, *LNCS*, vol. 3953, pp. 98–109. Springer (2006)
49. Kirchner, A., Schadschneider, A.: Simulation of evacuation processes using a bionics-inspired cellular automaton model for pedestrian dynamics. Physica A: Statistical Mechanics and its Applications **312**(1-2), 260–276 (2002)
50. Kirt, T., Vainik, E., Võhandu, L.: A method for comparing self-organizing maps: case studies of banking and linguistic data. In: Eleventh East-European Conference on Advances in Databases and Information Systems ADBIS, p. 107C115. Varna, Bulgaria: Technical University of Varna (2007)
51. Koller-Meier, E., Ade, F.: Tracking multiple objects using the Condensation algorithm. Robotics and Autonomous Systems **34**(2-3), 93–105 (2001)
52. Kong, D., Gray, D., Tao, H.: Counting Pedestrians in Crowds Using Viewpoint Invariant Training. British Machine Vision Conference (2005)
53. Kong, D., Gray, D., Tao, H.: A viewpoint invariant approach for crowd counting. Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)-Volume 03 pp. 1187–1190 (2006)
54. Kretz, T., Schreckenberg, M.: F.a.s.t. - floor field- and agent-based simulation tool (2006)
55. Lefebvre, G., Laurent, C., Ros, J., Garcia, C.: Supervised Image Classification by SOM Activity Map Comparison. Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)-Volume 02 pp. 728–731 (2006)
56. Legion: Http://www.legion.biz/about /index.html
57. Leibe, B., Seemann, E., Schiele, B.: Pedestrian detection in crowded scenes. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005. **1** (2005)
58. Li, S.Z., Zhu, L., Zhang, Z., Blake, A., Zhang, H., Shum, H.: Statistical learning of multi-view face detection. In: Proceedings of the 7th European Conference on Computer Vision-Part IV, pp. 67–81. Springer-Verlag, London, UK (2002)
59. Lin, S., Chen, J., Chao, H.: Estimation of number of people in crowded scenes using perspective transformation. IEEE Transactions on Systems, Man and Cybernetics, Part A, **31**(6), 645–654 (2001)
60. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. International Joint Conference on Artificial Intelligence **81**, 674–679 (1981)

61. Ma, R., Li, L., Huang, W., Tian, Q.: On pixel count based crowd density estimation for visual surveillance. IEEE Conference on Cybernetics and Intelligent Systems **1** (2004)
62. Marana, A., da Costa, L., Lotufo, R., Velastin, S.: On the Efficacy of Texture Analysis for Crowd Monitoring. Proceedings of the International Symposium on Computer Graphics, Image Processing, and Vision-Volume 00 p. 354 (1998)
63. Marana, A., Da Fontoura Costa, L., Lotufo, R., Velastin, S.: Estimating crowd density with Minkowski fractal dimension. IEEE International Conference on Acoustics, Speech, and Signal Processing, 1999. ICASSP'99. Proceedings., **6**, 3521–3524 (1999)
64. Marana, A., Velastin, S., Costa, L., Lotufo, R.: Estimation of crowd density using image processing. IEE Colloquium on Image Processing for Security Applications (Digest No: 1997/074), p. 11 (1997)
65. Marana, A., Velastin, S., Costa, L., Lotufo, R.: Automatic estimation of crowd density using texture. Safety Science **28**(3), 165–175 (1998)
66. Marques, J., Jorge, P., Abrantes, A., Lemos, J.: Tracking Groups of Pedestrians in Video Sequences. IEEE 2003 Conference on Computer Vision and Pattern Recognition Workshop **9**, 101 (2003)
67. Mathes, T., Piater, J.: Robust non-rigid object tracking using point distribution models. Proc. of British Machine Vision Conference (BMVC) **2** (2005)
68. Maurin, B., Masoud, O., Papanikolopoulos, N.: Monitoring crowded traffic scenes. The IEEE 5th International Conference on Intelligent Transportation Systems, 2002. Proceedings. pp. 19–24 (2002)
69. McKenna, S., Jabri, S., Duric, Z., Rosenfeld, A., Wechsler, H.: Tracking groups of people. Computer Vision and Image Understanding **80**(1), 42–56 (2000)
70. Mittal, A., Davis, L.: M 2 Tracker: A Multi-View Approach to Segmenting and Tracking People in a Cluttered Scene. International Journal of Computer Vision **51**(3), 189–203 (2003)
71. Mokhtarian, F., Abbasi, S., Kittler, J.: Robust and efficient shape indexing through curvature scale space. Proc. British Machine Vision Conference **62** (1996)
72. Musse, S., Thalmann, D.: A Model of Human Crowd Behavior: Group Inter-Relationship and Collision Detection Analysis. Proc. Workshop of Computer Animation and Simulation of Eurographics **97**, 39–51 (1997)
73. Okuma, K., Taleghani, A., de Freitas, N., Little, J., Lowe, D.: A boosted particle filter: Multitarget detection and tracking. European Conference on Computer Vision **1**, 28–39 (2004)
74. Pan, X., Han, C., Dauber, K., Law, K.: Human and social behavior in computational modeling and analysis of egress. Automation in Construction **15**(4), 448–461 (2006)
75. PRISMATICA: Http://prismatica.king.ac.uk/
76. Rahmalan, H., Nixon, M., Carter, J.: On Crowd Density Estimation for Surveillance. The Institution of Engineering and Technology Conference on Crime and Security pp. 540C–545 (2006)
77. Rasmussen, C., Hager, G.: Joint probabilistic techniques for tracking multi-part objects. Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on pp. 16–21 (1998)
78. Reid, D.: An algorithm for tracking multiple targets. Automatic Control, IEEE Transactions on **24**(6), 843–854 (1979)
79. Reisman, P., Mano, O., Avidan, S., Shashua, A., Ltd, M., Jerusalem, I.: Crowd detection in video sequences. Intelligent Vehicles Symposium, 2004 IEEE pp. 66–71 (2004)
80. Sidenbladh, H., Wirkander, S.: Tracking random sets of vehicles in terrain. Proc. 2003 IEEE Workshop on Multi-Object Tracking **9**, 98 (2003)
81. Smith, K., Gatica-Perez, D., Odobez, J.: Using particles to track varying numbers of interacting people. Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Volume 1-Volume 01 pp. 962–969 (2005)
82. Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis, and Machine Vision, 1998. Tech. rep., ISBN 0-534-95393-X (1998)
83. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (1999)

84. Sutton, R., Barto, A.: Reinforcement Learning: An Introduction. MIT Press (1998)
85. Swets, D., Punch, B.: Genetic algorithms for object localization in a complex scene. IEEE International Conference on Image Processing pp. 595–598 (1995)
86. University, S.: CIFE Seed Project 2004-2005, 2005-2006, http://eil.stanford.edu/egress/
87. Velastin, S., Yin, J., Davies, A., Vicencio-Silva, M., Allsop, R., Penn, A.: Automated measurement of crowd density and motion using imageprocessing. Road Traffic Monitoring and Control, 1994., Seventh International Conference on pp. 127–132 (1994)
88. Venegas, S., Knebel, S., Thiran, J.: Multi-object tracking using particle filter algorithm on the top-view plan. Technical report,LTS-REPORT-2004-003, EPFL (2004). Http :// infoscience.epfl.ch / getfile.py? mode= best & recid=87041
89. van Vliet, L., Young, I., beek, P.: Recursive gaussian derivative filters. In: In Proc. l4th International Conference on Pattern Recognition (ICPR'98), volume 1, IEEE Computer Society Press, Aug. 1998., pp. 509–514 (1998). URL citeseer.comp.nus.edu.sg/565386.html
90. Vu, V., Bremond, F., Thonnat, M.: Human Behaviour Visualisation and Simulation for Automatic Video Understanding. Proc. of the 10th Int. Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG–2002), Plzen–Bory, Czech Republic pp. 485–492 (2002)
91. Wu, B., Nevatia, R.: Detection of Multiple, Partially Occluded Humans in a Single Image by Bayesian Combination of Edgelet Part Detectors. Tenth IEEE International Conference on Computer Vision, 2005. ICCV 2005 **1**, 90–97 (2005)
92. Wu, B., Nevatia, R.: Tracking of multiple, partially occluded humans based on static body part detection. In: CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 951–958 (2006)
93. Yin, J., Velastin, S., Davies, A.: Image Processing Techniques for Crowd Density Estimation Using a Reference Image. Proc. 2nd Asia-Pacific Conf. Comput. Vision **3**, 6–10 (1995)
94. Zhan, B., Remagnino, P., Velastin, S.: Analysing Crowd Intelligence. Second AIxIA Workshop on Ambient Intelligence (2005)
95. Zhan, B., Remagnino, P., Velastin, S.: Mining paths of complex crowd scenes. Advances in Visual Computing: First International Symposium pp. 126–133 (2005)
96. Zhan, B., Remagnino, P., Velastin, S.: Mining paths of complex crowd scenes. Lecture notes in computer science pp. 126–133 (2005). ISBN/ISSN 3-540-30750-8
97. Zhan, B., Remagnino, P., Velastin, S.: Visual analysis of crowded pedestrain scenes. XLIII Congresso Annuale AICA pp. 549–555 (2005)
98. Zhan, B., Remagnino, P., Velastin, S., Bremond, F., Thonnat, M.: Matching gradient descriptors with topological constraints to characterise the crowd dynamics. Visual Information Engineering, 2006. VIE 2006. IET International Conference on pp. 441–446 (2006). ISSN: 0537-9989, ISBN: 978-0-86341-671-2
99. Zhan, B., Remagnino, P., Velastin, S., Monekosso, N., Xu, L.: A Quantitative Comparison of Two New Motion Estimation Algorithms. LECTURE NOTES IN COMPUTER SCIENCE **4841**, 424 (2007)
100. Zhan, B., Remagnino, P., Velastin, S.A., Monekosso, N., Xu, L.Q.: Motion estimation with edge continuity constraint for crowd scene analysis. In: G. Bebis, R. Boyle, B. Parvin, D. Koracin, P. Remagnino, A.V. Nefian, M. Gopi, V. Pascucci, J. Zara, J. Molineros, H. Theisel, T. Malzbender (eds.) ISVC (2), *Lecture Notes in Computer Science*, vol. 4292, pp. 861–869. Springer (2006)
101. Zhao, T., Nevatia, R.: Tracking multiple humans in complex situations. Pattern Analysis and Machine Intelligence, IEEE Transactions on **26**(9), 1208–1221 (2004)
102. Zhao, T., Nevatia, R.: Tracking multiple humans in crowded environment. Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition **2**, II–406–II–413 (2004)

Augmenting Professional Training, an Ambient Intelligence approach

B. Zhan, D.N. Monekosso, , S. Rush, P. Remagnino, S.A. Velastin

1 Introduction

This chapter summarizes research developed for an inter-disciplinary project on computer vision methods applied to enhance and automate the professional training of nurses [1]. The project has engaged the computer vision team in the Faculty of Computing, Information Systems and Mathematics and the School of Nursing at Kingston University.

The inter-faculty project is the first attempt at Kingston University to design an Ambient Intelligence system, for use in the training of professionals. Ambient Intelligence is a paradigm introduced by the European community in 2000 [7], to describe a user centric intelligent system, capable of serving the generic or specific user, responding to the needs of the individual and the group. In the context of our project, the paradigm is interpreted as a set of guidelines to develop algorithms capable of interpreting behavior in a very complex environment monitored by an array of cameras.

The School of Nursing at Kingston Hill campus trains student nurses, paramedic and medical students (St. George's Medical school). The training consists of individual and group practical exercises based on taught techniques, entailing both medical and managerial skills. Group skills are tested in large simulations. During term time, practice skills training is organized in series of morning and afternoon sessions. Simulations entail a preliminary preparatory round table, the actual sim-

B.Zhan, D.N.Monekosso, P.Remagnino and S.A.Velastin
Faculty of Computing, Information Systems and Mathematics, Kingston University, e-mail: B.Zhan@kingston.ac.uk

S. Rush
School of Nursing, Kingston University e-mail: srush@hscs.sgul.ac.uk

[1] The research was partially funded by the *European Office of Aerospace Research and Development* (EOARD) project FA8655-06-1-3013.

ulation where skills are tested at individual and in groups and a final round table discussion, where strengths and weaknesses of the assessed student are discussed.

All scenes are extremely complex and cluttered, including crowded situation with clutter and frequent occlusions. The scenes are very crowded and occlusions are very frequent. In order to simplify the computer vision processing all individuals in the scene are asked to wear a colored tabard. Four colors are used to distinguish among instructors (blue), student nurses (yellow), medical and paramedic students (green) and patients (red). Tabards were to be worn for the entire duration of the exercise.

Conventional training of nurses and medical students is very time consuming and when large numbers of students are involved, it is very hard for an instructor to assess correctly the performance of a student or a group of students. The School of Nursing runs state of the art training methodology, engaging students in individual and team work. Assessment is usually carried out on the fly and by recording footage of students' performance and illustrating to individuals and classes best practice, encouraging less capable students, praising best practice of better students. The Skills' laboratory situated at Kingston Hill campus at Kingston University can host up to 30 students at a time with instructors and role players engaged in large simulations. The lab is currently endowed with a variety of medical equipment and mobile and fixed cameras. The following two figures illustrate a round table example and one of the installed cameras, used to acquire video footage. In our experiments, we have

Table 1 figure:

employed four cameras (pan-tilt-zoom used as fixed cameras). A preliminary study was carried out by analyzing the four views independently, attempting at generating the automatic understanding of an evolving scene.

Section 2 describes the algorithm used to track people in the environment, Section 3 describes the algorithm designed to deliver an automatic reasoning about the scene. Section 4 illustrates some results and Section 5 summarizes the proposed method and introduces some future work.

2 Color tracking of people

As already shown in other research, a skin color model can be estimated by acquiring video data of a given color using template patches and via the training of a color model using the expectation maximization algorithm [4]. In order to optimize the model, the skin color data can be studied and an optimal initialization defined in terms of number of clusters and initial positions and approximating functions.

A color model is fairly robust to changes in illumination but it has the weakness of being specific to the camera used to acquire the training data. In all our tests, each new video camera we have used to acquire video footage had its own color model. As the training can be performed off line, the limitation is not prohibitive. Color models were trained for the four different colors used to recognize the categories of people. these include the student nurse (yellow), the instructor (blue), the patient (red) and the medical student (green).

In order to track color patches, identifying body parts, we have implemented the CAMSHIFT algorithm. The CAMSHIFT algorithm was originally proposed in [1], as an evolution of the MEANSHIFT algorithm [6, 3]. CAMSHIFT adapts to evolving a probability density function (PDF) by alternating cycles of the MEANSHIFT algorithm with a resizing of the search window. The window size is a function of the center of mass of the probability density map (0^{th} moment).

Tracking color patches entails running the CAMSHIFT algorithm for each patch. However, this is not sufficient to maintain hypotheses in a rapidly evolving scene. That is why our method keeps track of a list of alive patches, by tracking them throughout the scene with the CAMSHIFT algorithm, removing those which have too low a probability associated for a number of frames and introducing new patches, whenever sufficiently large new patches appear in the scene with a sufficiently high probability.

More details of the developed algorithm can be found in [2].

3 Counting people by spatial relationship analysis

Colour segmentation generates fragmentation, by identifying one person with more blobs. This is mainly due to occlusions and self-occlusions, but also by the reflections on the person. In our algorithm spatial relationships is employed to group the blobs split from a single person. At first, for each frame a graph is created with links between all identified blobs. Each link is then evaluated to judge whether the linked blobs should be merged into a cluster to recover an individual or they should be kept separate, making the assumption both blobs are disjoint, likely to be part of different people in the scene.

3.1 Links Between Blobs

Links are built between each pair of blobs to estimate the spatial relations. For real scenario with people entering and exiting at random in the scene. The creation, deletion and updating of the links are required to be automatic according to the changing of the situations. In this system a specified framework has been designed for tackling the problem. The system requires that a blob A has links with all the other blobs in the scene during its life cycle. However during the life cycle of A, blob B could entered and the leave the scene. Under such circumstance, A should then be linked to B onces B has entered and the link should be released right after B has left the scene. The complexity of the problem increases when the number of people involved increases.

As a result, the crucial phases of this process are creation, deletion and updating. The solution has been designed as creation of the links are triggered by the entering of blobs, deletions are triggered by the exiting of blobs and for each time segment Δt links are updated by the situation at that moment in time.

Using the above example, a link is created between A and B, when B enters the scene. The link should be kept updated while B is staying in the scene. The link should then be released when B is no longer in the scene. *In the real world situation it is difficult and unreliable to predict which blob is A and which is B, that is, to predict among two blobs in the scene which one is going to exit earlier.* for algorithmic simplicity, a link is bi-directional, so each link between blob A, for instance, and any other blob, also implies that all linked blobs keep track of the existence of A. *Creation of the link has been proposed to be: when a blob A enters the scene, links are created between A and all the other blobs in the scene, the access of a link are added to A and to the other end of the link.* When a blob leaves the scene, it sends a signal to all the links connected to it, to release and delete them. At each frame, sampled at a given Δt, the system checks the blobs to create, delete or update the existing links. Algorithm 1 illustrates this process.

3.2 Distance Calculation

Distance information is held by the built links to represent the spatial relations between different blobs. The distance between blobs is calculated as the Euclidean distance between the blob's centers. Because of the perspective distortion, the absolute value of the Euclidean distance cannot be used to estimating the spatial relation between the blobs. For instance, two blobs at an absolute distance of 50 pixels, could be close to each other when they are in front of the camera while they could be far from each other when they are distant from the camera. Hence, a method of calculating relative distance by comparing the absolute distance with the size of connected blob has been proposed here, i.e. the ratio of the absolute distance and the blob size was used. In this method the variation of sizes of blobs in different locations needs is considered. So the size of blob which has upper bound is used in this calculation,

Algorithm 1 The creation, deletion and update of the links in a frame

if objects: $O^-_{0...m}$ are leaving the scene **then**

  for $i = 0$ to m **do**
    Object O^-_i send signals to all the links connected with it
    Delete O^-_i
  end for
end if
Delete links with signals
if objects: $O^+_{0...n}$ are entering the scene **then**

  for $j = 0$ to n **do**
    Build links between object O^+_j and all objects existing objects in the scene
  end for
end if
Update all the existing links

as in theory that the blob is further from the camera and in it should has a smaller size. The Euclidean distance is used as the absolute distance between blob i and j is as below:

$$D_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{1}$$

Where (x_i, y_i) and (x_j, y_j) are the coordinates of center points of blob i, j, respectively. And the temporal relative distance of blob i and j is calculated as:

$$d_{ij} = \frac{D_{ij}}{\sqrt{w_k^2 + h_k^2}}, \; k = \begin{cases} j, \; if \; y_i - 0.5h_i < y_j - 0.5h_j \\ i, \; otherwise. \end{cases} \tag{2}$$

where w_k and h_k are the dimensions of the blob. *insight.*

The above calculations are carried out in a single frame. A temporal average operator has been applied over every Δt frames for each distance calculation. This operation can reduce the instability caused by the tracking algorithm, thus the video sequence has been divided into fixed length time segments, i.e. each time segment contains distance information for Δt frames. Equation 2) describes the calculation of this distance,

$$\bar{d}_{ij}(T) = \frac{1}{\Delta t} \sum_{\Delta t} d_{ij}(T - \Delta t) \tag{3}$$

so the distance between blobs i and j at time T is the average of the distances over the previous Δt frames. The major purpose of this operation is to stabilize the distance and Δt is a short time segment, for example in our case we use a 8 frame Δt, which is equivalent to 0.5 seconds.

3.3 Temporal Pyramid of distance

Short term spatial relations are not sufficient for clustering blobs. A temporal pyramid of distance scheme has been introduced to maintain the long term distance information. Essentially, two blobs belong to the same cluster if they are close to each other from the moment they both appear in the scene to the moment they are clustered together. For each pair of blobs, the scheme will take into account the distance information from each level of the pyramid and calculate the overall probability that they belong to the same cluster. This scheme is based on an assumption that two persons are not likely to stay next to one another during a very long time period.

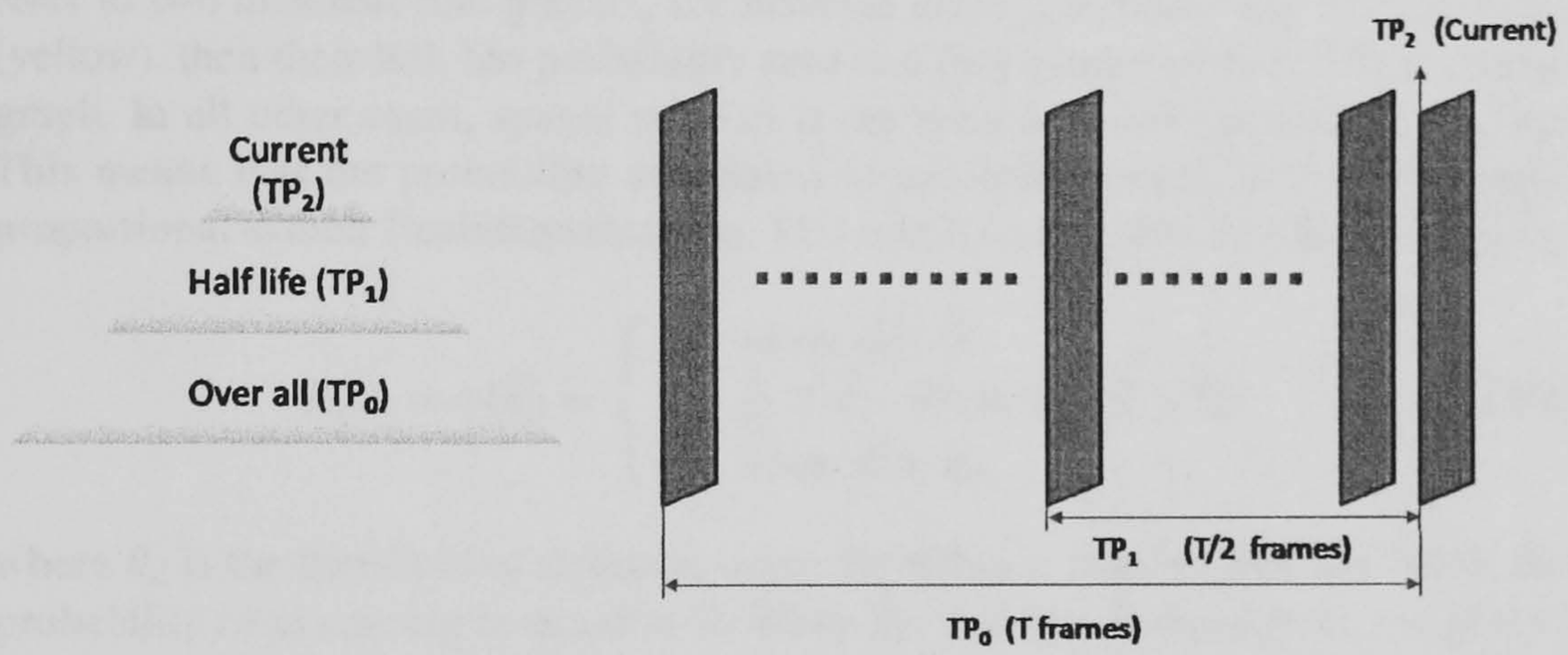


Fig. 1 Temporal Distance Pyramid: The bottom layer holds the overall distance information from time 0 to time T; the middle layer holds the distance information from time $T/2$ to T and the top layer holds the distance information for the current time slice T.

The temporal pyramid consists of three levels : the bottom layer holds the overall distance information between two blobs from their appearance in the scene to the present time, the top layer holds the present distance information and the middle layer holds the information from the half time to the present; this is illustrated in Fig. 1.The generation of the temporal distance pyramid is:

$$TP_0(T) = \bar{d}(0 \rightarrow T) = \frac{1}{T}\sum_{t=1}^{T}\bar{d}(t) \tag{4}$$

$$TP_1(T) = \bar{d}(T/2 \rightarrow T) = \frac{1}{\frac{T}{2}}\sum_{t=\frac{T}{2}}^{T}\bar{d}(t) \tag{5}$$

$$TP_2(T) = \bar{d}(T) = \bar{d}(T) \tag{6}$$

where $TP_0(T)$ to $TP_2(T)$ represents the distance information held from the bottom layer to the top layer at time T. In practice to reduce the redundant calculations of top layer($TP_0(T)$) and middle layer($TP_1(T)$), a recursive method has been employed and the equations are modified as follows:

$$TP_0(T) = \frac{1}{T}(TP_0(T-1) \times (T-1) + \overline{d}(T)) \tag{7}$$

$$TP_1(T) = \frac{1}{\frac{T}{2}}(TP_1(T-1) \times \frac{T-1}{2} - \overline{d}(\frac{T}{2}-1) + \overline{d}(T)) \tag{8}$$

$$TP_2(T) = \overline{d(T)} \tag{9}$$

3.4 Probability of clustering

Clustering is carried out for each category, so, if two blobs belong to colors that refer to two different role players, for instance instructor (blue) and student nurse (yellow), then their link has probability zero and they cannot be linked to the same graph. In all other cases, spatial relation is the main criterion used for clustering. This means that the probability associated to the link between blobs is inversely proportional to their Euclidean distance. This rule is represented by a function $\varphi(\overline{d})$:

$$P(\overline{d}) = \varphi(\overline{d}) = \begin{cases} 1, & when\ \overline{d} = 0; \\ 1 - \frac{1}{\theta_d} \times \overline{d}, & when\ 0 \leq \overline{d} \leq \theta_d; \\ 0, & when\ \overline{d} > \theta_d. \end{cases} \tag{10}$$

where θ_d is the threshold of distance, when the distance falls beyond this value, the probability of clustering is equal to 0. When the distance is equal to 0, the probability is equal to 1. The probability of clustering two blobs with a distance that falls between 0 and θ_d is interpolated with a linear function. Each layer of the temporal distance pyramid provides a probability of clustering and the outcome of the three layers has been averaged as follows:

$$P_{dis} = \frac{1}{3}(P(TP_0) + P(TP_1) + P(TP_2)) \tag{11}$$

The overall size of the blobs is also used to bias the probability of clustering blobs. A linear approximation of the blob size at different locations of the scene has been used as reference. The size of the overall bounding box between blobs is compared against the estimated reference, according to their locations. This comparison is represented by the ratio:

$$\overline{s} = \frac{S_o}{S_r} \tag{12}$$

where S_o is the size of the blobs and S_r is the reference size from the linear approximation. The probability of clustering by area is calculated by $\varphi(\overline{s})$:

$$P_{size} = P(\overline{s}) = \varphi(\overline{s}) = \begin{cases} 1, & when\ \overline{s} = 0; \\ 1 - \frac{1}{\theta_s} \times \overline{s}, & when\ 0 \leq \overline{s} \leq \theta_s; \\ 0, & when\ \overline{s} > \theta_s. \end{cases} \tag{13}$$

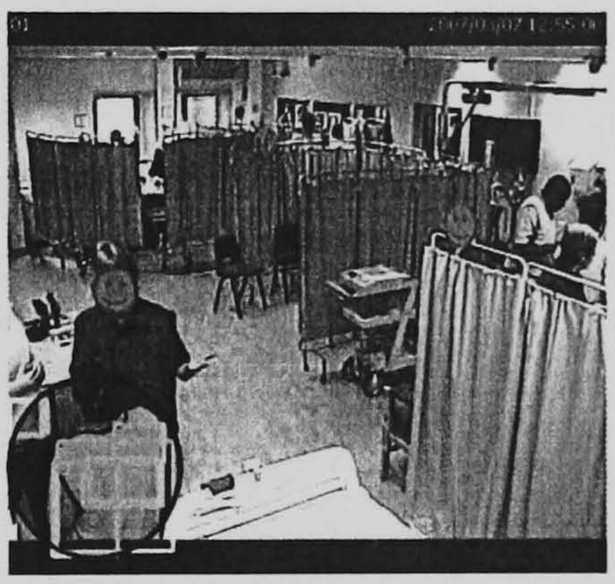
where θ_s is the threshold of the ratio of the size ($\bar{s}$). $\varphi(\bar{s})$ is employed for the reason that smaller fragments should increase the probability to cluster. The overall probability of clustering is:

$$P = P_{dis} \times P_{size} = P(\bar{d}) \times P(\bar{s}) \tag{14}$$

3.5 Clustering

For each frame, the clustering takes place in two steps: pair clustering and sub clustering. Pair clustering checks all pairs of blobs, clustering together all the pairs with high probability. This rule ensures that all the blobs which potentially belong to the same person are clustered together. If two blobs are selected to be clustered and they already belong to two clusters, then the clusters can be merged (Fig.2(a)). Pair clustering may generate bad clustering. In fact, blobs which belong to different persons could be clustered together (Fig.2(b)). The second step - sub clustering is used to get the scores of different number n ($1 \leq n \leq N$)of sub clusters of a cluster C which contains N blobs. To achieve the best number of sub clusters, a process of sub clustering has been carried out. In a cluster generated from the pair clustering step, each pair of blobs is associated with a probability of clustering which is generated by the method described in 3.4. The strength Γ of a cluster is defined as:

(a) A frame in which multiple blobs (illustrated with a black oval) should be clustered together.



(b) A frame in which blobs belong to different persons and could be clustered together (illustrated with a black oval).

Fig. 2 Two frames of problems in clustering

$$\Gamma = \frac{1}{C_N^2} \sum_{i=0}^{C_N^2} P_i \tag{15}$$

where N is the total number of blobs, so there are C_N^2 pairs of blobs.The pairs of blobs that with high probability of clustering which make them connected are called *Connections*; in contrast, the pairs of blobs without *Connections* between them are called *Unconnections*. The basic rule of sub clustering is that every time the weakest *Connection* is removed, the blobs are reclustered by the remaining *Connections*. The score of the operation equals the energy cost E of removing the *Connection* and the related *Unconnection*.

$$\Lambda = \frac{1}{n}\sum E + \frac{1}{m}\sum \Gamma \tag{16}$$

where the energy cost of removing a *Connection* of probability of clustering P is

$$E = 1 - P \tag{17}$$

This operation is kept to carry on until all the *Connections* are removed, meanwhile

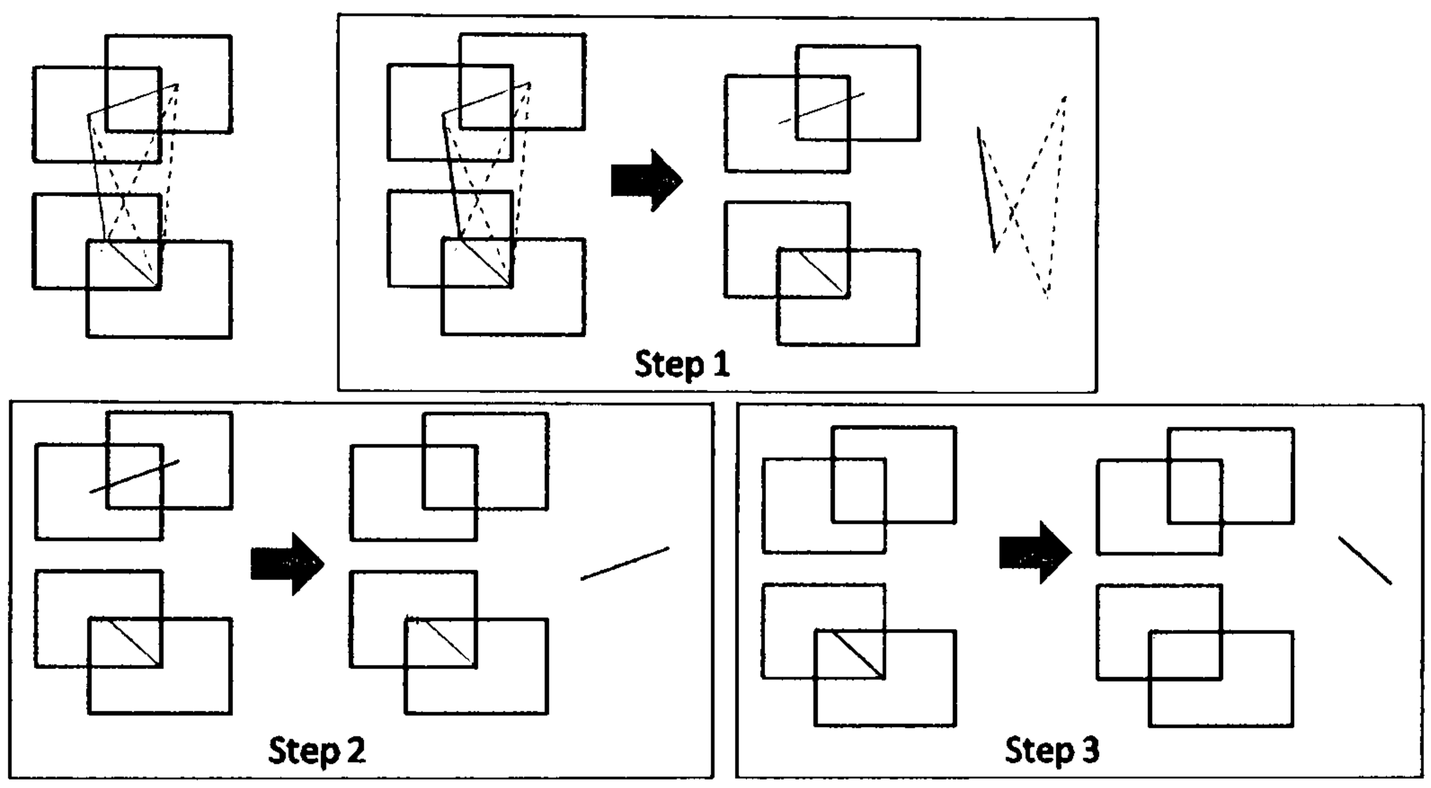


Fig. 3 An example of subclustering. Solid lines between blobs show the *Connections* and the dash lines are the *Unconnections*. In each step, the black *Connections* is removed, and the related *Unconnections* are removed. This operation is carried out until all the *Connections* are moved and all the blobs are isolated.

all the blobs are isolated . Fig.3. shows an example of the subclustering process of a cluster containing 4 blobs.

During the operation, the scores are accumulated for different number of sub-clusters. In this case the number of sub clusters with highest score is selected to be added to the number of the people and the subclusters are regarded as individuals. The total number of people is the sum of the selected numbers of sub clusters of all the clusters in the frame.

4 Experimental Results

For a video sequence, each frame a number of people as well as locations of people are retrieved. To access the system performance, ground truth is manually marked up by the The ViPER Ground Truth Authoring Tool(ViPER-GT tool), which is a part of The Video Performance Evaluation Resource (ViPER) developed by Language and Media Processing Laboratory, University of Maryland [2]. The ground truthing is carried out frame by frame, and each person is selected by a bounding box (Fig.4). In our work, performance was evaluated using measures borrowed from the information retrieval literature. Recall and Precision, which have been used in evaluating search strategies, are used here to test the results of our algorithm against ground truth information. Recall is the ratio of the number of relevant records retrieved to the total number of relevant records in the database. Precision is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved. The Precision-Recall curve is employed to give a informative picture of system performance[5]. For a video sequence, each time the bounding boxes of the ground truth (GT) represent the relevant records; the bounding boxes generated by the system (RE) are considered the retrieved records. Therefore the Recall and Precision of each frame are calculated by comparing GT and RE.

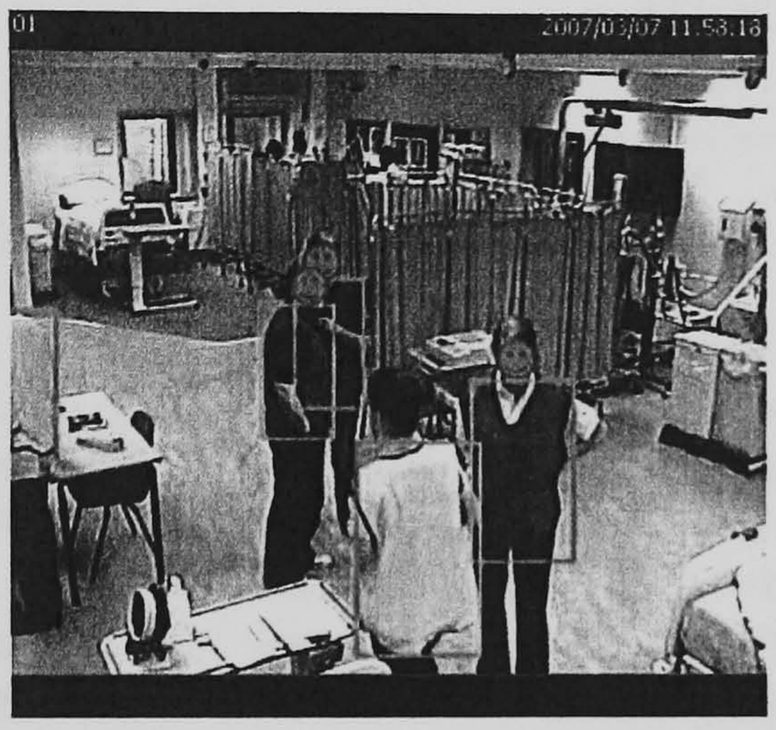


Fig. 4 A ground truth example from ViPER-GT

$$Recall = \frac{GT \bigcap RE}{GT} \tag{18}$$

$$Precision = \frac{RE \bigcap GT}{RE} \tag{19}$$

[2] The detail of ViPER and ViPER Ground Truth Authoring Tool are available online at http://viper-toolkit.sourceforge.net/.

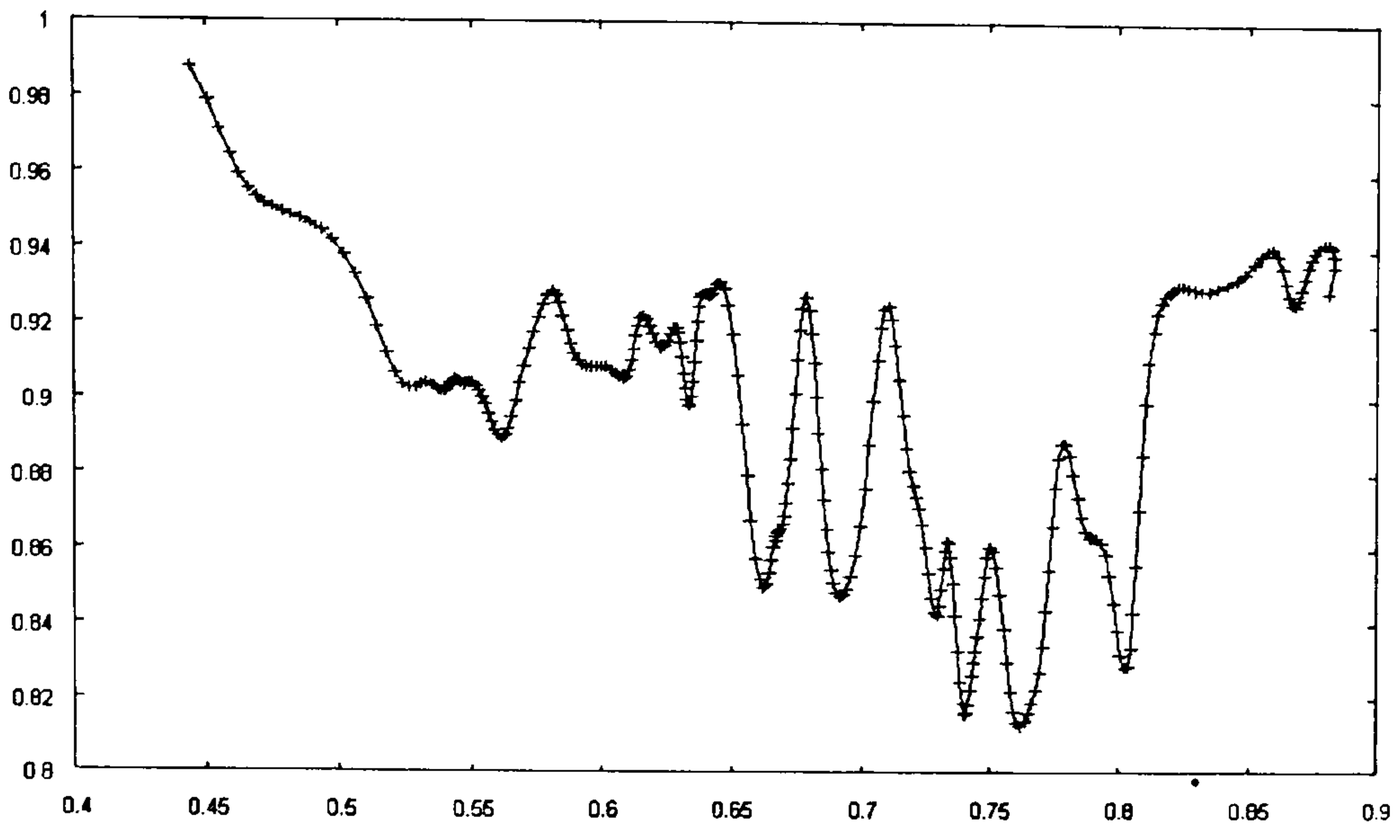


Fig. 5 Precision-Recall curve

Profession information is also considered, i.e. intersections of *GT*s and *RE*s with different colors do not count. The Recall and Precision estimates have been recorded along time scale for the video whole sequences, and each pair of the measures contributes as a point on the Precision-Recall curve. Fig. 5 shows the Precision-Recall curve for a video sequence. The curve shows that Precision keeps above 0.8 and Recall increases to up to 0.8 without a large drop on Precision. Fig. 6 illustrates the counting results from different situations with different number of people and different number of professions. These results show that the system has a stable performance under different circumstance.

5 Conclusions

This chapter has described an intelligent system that follows the guidelines of the Ambient Intelligence paradigm. At present, only cameras are used to recognize behavior and estimate the category and number of people in the scene. Color models are used to track people in the scene and provide sufficient information to the system to generate graphs of detected and tracked color patches. The color patches are then used to generate a graph that is automatically analyzed by an algorithm, which can cluster blobs and estimate the number of people in the scene.

The contribution of this chapter is mainly the design of a robust algorithm for the interpretation of a complex scene. Future work will include the combination of evidence from all the cameras, the use of stereo cameras for occlusion disambiguation and the introduction of radio frequency technology following the *zigbee* standard to help with the recognition of positional information of scene actors and a better description of the scene.

Fig. 6 Counting People: In these figures, ellipses represent the original blobs; thick outlines of shapes (rectangles, ellipses) show the existences of individuals; thin outlines of ellipses show the existences of clustered blobs.

References

1. Bradski, G.: Computer vision face tracking for use in a perceptual user interface. Intel Technology Journal **2**
2. B.Zhan, N.D.Monekosso, T.Rukhsana, P.Remagnino, Y.Kuno, A.Mansur: Skin patches trajectories as scene dynamics descriptors. In: International Association of Pattern Recognition Conference on Machine Vision Applications 2007, p. pp. (2007)
3. Cheng, Y.: Mean shift, mode seeking, and clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence **17**(8), 790–799 (1995)
4. C.M.Bishop: Pattern Recognition and Machine Learning. Springer (2006)

5. Davis, J., Goadrich, M.: The relationship between precision-recall and roc curves. In: Proceedings of the 23^{rd} international conference on Machine learning, pp. 233–240. ACM (2006)
6. Fukunaga, K., Hostetler, L.: The estimation of the gradient of a density function, with applications in pattern recognition. IEEE Transactions on Information Theory **21**, 32–40 (1975)
7. P.Remagnino, G.L.Foresti, T.Ellis (eds.): Ambient Intelligence: a Novel Paradigm. Springer (2004)

Self-Organizing Maps for the Automatic Interpretation of Crowd Dynamics

B.Zhan, P.Remagnino,N.Monekosso, S.A.Velastin
{B.Zhan, P.Remagnino}@kingston.ac.uk
Digital Imaging Research Centre, Faculty of Computing, Information Systems and Mathematics, Kingston University, UK

Abstract. This paper introduces the use of self-organizing maps for the visualization of crowd dynamics and to learn models of the dominant motions of crowds in complex scenes. The self-organizing map (SOM) model is a well known dimensionality reduction method proved to bear resemblance with characteristics of the human brain, representing sensory input by topologically ordered computational maps. This paper proposes algorithms to learn and compare crowd dynamics with the SOM model. Different information is employed as input to the used SOM. Qualitative and quantitative results are presented in the paper.

1 Introduction

We are interested in devising methods to automatically measure and model the crowd phenomenon. Crowded public places are increasingly monitored by security and safety operators. There are companies (for example LEGION [1]) that employed large resources to study the phenomenon and generate realistic simulations: for instance to optimize the flow of people of a public space.

Computer Vision research offers a large number of techniques to extract and combine information of a video sequence acquired to observe a complex scene. The life cycle of a computer vision system includes the acquisition of the monitored scene with one or more homogeneous or heterogeneous cameras, the extraction of features of interest and then the classification of objects, people and their dynamics. In simple scenes the background is extracted with statistical methods and then foreground data and related information are inferred to describe and model the scene. Background is usually defined as stationary data, for instance man made structure, such as buildings, in a typical video surveillance application, or the indoor structure of a building in a safety application, for instance deployed to monitor and safeguard elderly people in a home.

Unfortunately, background modeling becomes rapidly less effective in complex scenes and its usefulness seems to be inversely proportional to the clutter measured in the scene. Figure 1 shows a small experiment testing the effectiveness of background modeling with different types of scenes. Three frames per chosen sequence and the resulting background image built with roughly 1000 frames are illustrated. The background modeling works well with the first scene;

it fails to recover the background of some regions in the second scene because of the frequent occupancy over these regions; and in the third scene, due to the continuous clutter, the background model can be barely recovered. Although

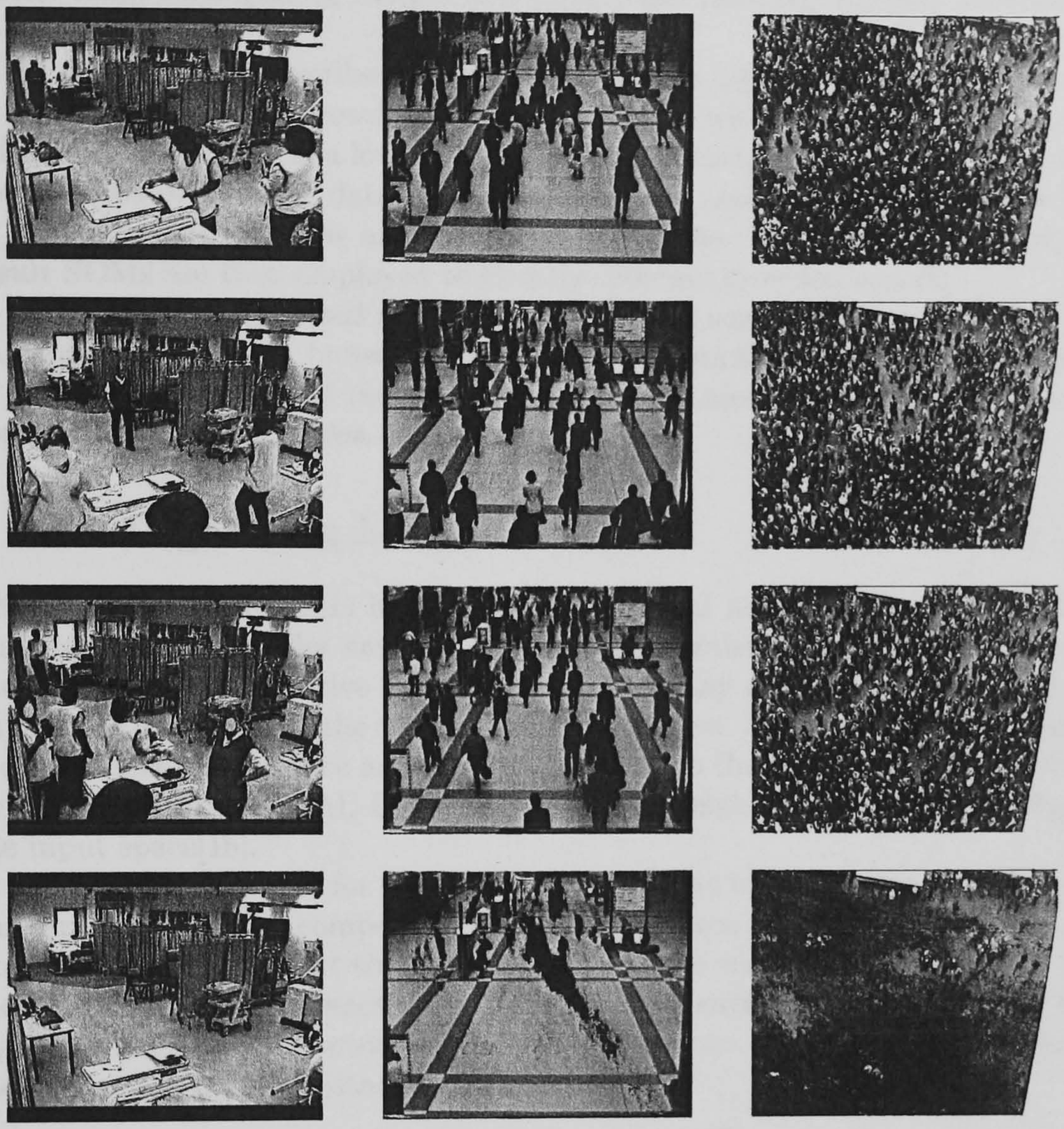


Fig. 1. The example frames and the built background images from three different scenes. Left to right: three different scenes; top to bottom, three example frames and the built background images, respectively.

sophisticated methods have been proposed for tracking crowded environments, such as the Particle Filter [2][3] and the Joint Probabilistic Data Association Filter (JPDAF) [4], the state of art describes only scenes with a limited number of people. In highly crowded scenes, tracking is not a viable option and it is more interesting and valuable to retrieve the global crowd motion instead of individual motion. Crowd motion estimation algorithms have been proposed using local descriptors [5] [6]. The overall objective of our research is to model

crowd dynamics. This is normally achieved based on the extracted information from visual data. Andrade [7][8][9] characterizes crowd behavior by observing the crowd optical flow associated with the crowd and use unsupervised feature extraction to encode normal crowd behavior. Zhan [10][11][12] proposes a crowd model using accumulated motion and foreground (moving objects) information of a crowded scene.

In this paper we describe some work carried out applying self-organized maps to learn the dominant crowd dynamics. SOMs are widely used in mapping multidimensional data onto a low-dimensional map, example of applications include the analysis of banking data, linguistic data [13] and image classification [14]. In this paper, optical flow and raw image have been used as input to SOM; the result SOMs are then employed to classify different crowded scenes.

This paper is organized as follow: In Section 2 some background of Self Organizing Map is given; in Section 3 the usage of optical flow input is introduced; in Section 4 the usage of raw image input is presented; and Section 5 gives the conclusion and a discussion on further work.

2 Self Organizing Map

The most common SOMs have neurons organized as nodes in a one- or two-dimensional lattice. The neurons of a SOM are activated by input patterns in the course of a competitive learning process. At any moment in time, only one output neuron is active, the so called winning neuron. Input patterns are from an n-dimensional input space and are then mapped to the one- or two- dimensional output space of the SOM. Every neuron has a weight vector which belongs to the input space[15].

There are two phases for tuning the SOM with an input pattern I , competing and updating. In the competing phase every neuron is compared with I ; the similarity between I and the weights of all of the neurons are computed; and the neuron $N(i_w, j_w)$ (denoted by the neuron's coordinates of the lattice) with highest similarity is selected as the winning neuron. In the update phase, for each neuron $N(i, j)$, a distance is calculated as

$$d^2 = (i - i_w)^2 + (j - j_w)^2 \tag{1}$$

the topological neighborhood function is then defined as:

$$h(n) = \exp(-\frac{d^2}{2\sigma^2(n)}) \tag{2}$$

where n denotes the time, which can also be explained as the number of iterations. and $\sigma^2(n)$ decreases with the time. The weight of each neuron $N(i, j)$ at time $n + 1$ is then defined by:

$$w(n + 1) = w(n) + \eta(n)h(n)(x - w(n)) \tag{3}$$

where $w(n)$ and $w(n+1)$ are the weights of a neuron at time n and $n+1$, while $\eta(n)$ is the learning parameter, which decreases with time.

3 Optical Flow as Input

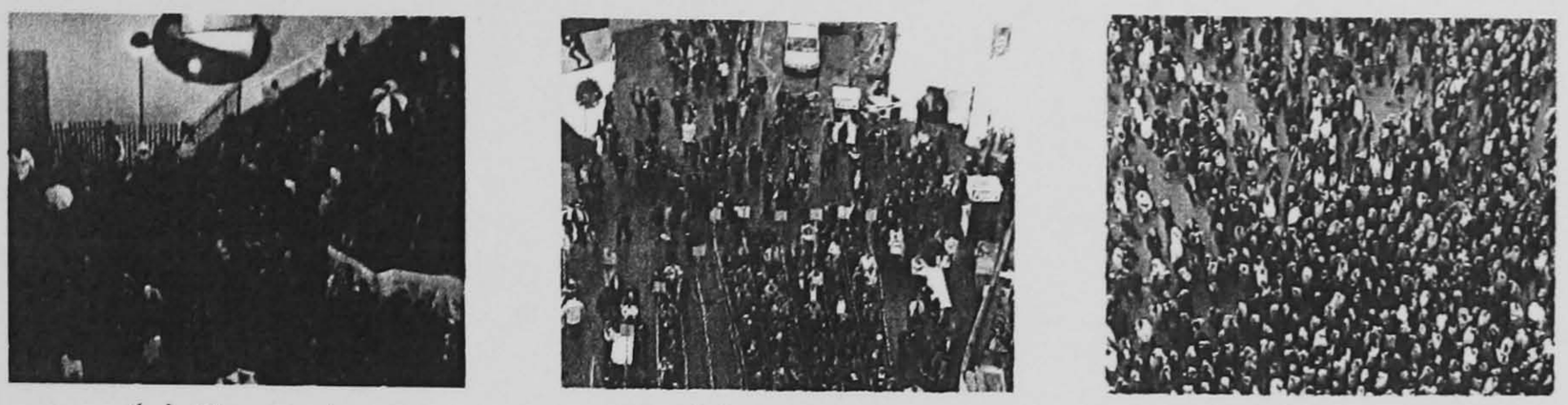
In this application, a SOM should capture the two major components of the crowd dynamics: spatial occurrence and orientation. Thus a four dimensional input space is chosen to be the weight space of the SOM, which can be represented as $f : (x, y, \theta, \rho)$. Each data from the input space can be explained as the location where crowd moves and the motion vectors in the form of angle θ and magnitude ρ. The SOM used in this experiment is organized in a two-dimensional space and represented by a regular square lattice.

3.1 Visualization

Figure 2 illustrates three different video sequences with different dynamics. These video sequences have been input into the system, and Figure 3 shows the output SOMs. In the figure SOMs are visualized in the input space, i.e. showing the weight vector of each neuron. In the visualization, the color arrows and their locations are from the weight vector of neurons; the location of the arrows are from the first two components of the weight vectors (x, y), and the arrows show the second two components - the components of motion (θ, ρ). The different colors of the arrows are also indicating the different orientation of the motion. In the first video (the left image in Figure 2) the major crowd is moving from bottom left to top right of the scene. There is another crowd flow from bottom right of the scene which joins the major flow. In its SOM (the first one in Figure 3) the neurons with green arrows are clearly from the major flow and the ones with red and purple arrows are from the minor flow. In the second video (the middle image in Figure 2) it is an area of an entrance to a public space. Most of the people move from top to bottom in the illustrated scene. The crowd in the upper part of the scene is sparser and moves faster when compared to the crowd in the lower part of the scene. There is also a minor flow, which joins the major flow from right of the scene. In the built SOM (the second SOM in Figure 3), again the flows are clearly indicated. Furthermore, the SOM takes an umbrella shape, which represents the shape of the flow constrained by the obstacles in the scene. In the third video (the right image in Figure 2) the scene is a large open area with multiple crowd flows. The major flow is moving from right to left; however there are several minor flows, most of which are in the lower part of the scene. Again the SOM (the third in Figure 3) captures the major dynamics and also some minor flows. From the three examples, it can be concluded that the SOMs not only preserves the dominant motion vector, but also represents the shape of the regions with dominant motion of the scenes.

3.2 Classification

Visualizations of the SOMs have already provided information of recurrent motion. Scene classification has been carried out using the characters captured by the SOMs. To achieve this, comparisons of the SOMs built for different scenes have been carried out. The classification is based on the similarities of the SOMs.



(a) Scene A (b) Scene B (c) Scene C

Fig. 2. The example frames from three different scenes.

The topological structures of the lattice of SOMs, as well as the weights of the neurons of the SOMs are used for the comparison. Topological structure is an important feature of SOM, a large number of methods have been proposed to measure it [16]. In this work a C- Measure is used, which is defined as: The similarity of the C- Measures of two different SOMs is calculated as

$$C = \sum_{(i,j \in_{i \neq j}) A \times A} F_A(i,j) F_V(w_i, w_j) \tag{4}$$

where F_A and F_V are the similarity on input space (i.e. weight space) and output space (i.e. SOM lattice), respectively. The i and j are the index of the neurons and w_i and w_j are the weight of the indexed neurons. The similarity of correspondent SOM neurons is calculated:

$$Sim_w = F_V(w_i, w_j) = \sum_{k=0}^{k<Dim_w} \frac{min(w_i^k, w_j^k)}{max(w_i^k, w_j^k)} \tag{5}$$

where Dim_w is the dimension of the weight space, w_i^k and w_j^k are the $k-th$ element of the weights w_i and w_j, respectively. An average over the lattice has been calculated. This equation is used for calculating the similarity of the weights of two neurons. The similarity of the structure is calculated as the similarity of the C-Measures:

$$Sim_c = \frac{min(C_i, C_j)}{max(C_i, C_j)} \tag{6}$$

A combination of the two similarities - weight similarity and structure similarity are calculated by:

$$Sim = \sqrt{Sim_c \times Sim_w} \tag{7}$$

The correspondence of the neurons is defined by the closeness of the values of their weights. Particularly for neuron (i) in SOM A, the correspondent neuron in SOM B is the one with the closest weight value of it. The matching could be asymmetrical, for example assuming for neuron (i) in SOM A, its correspondent neuron in SOM B is neuron (j); however for the neuron (j) its correspondent

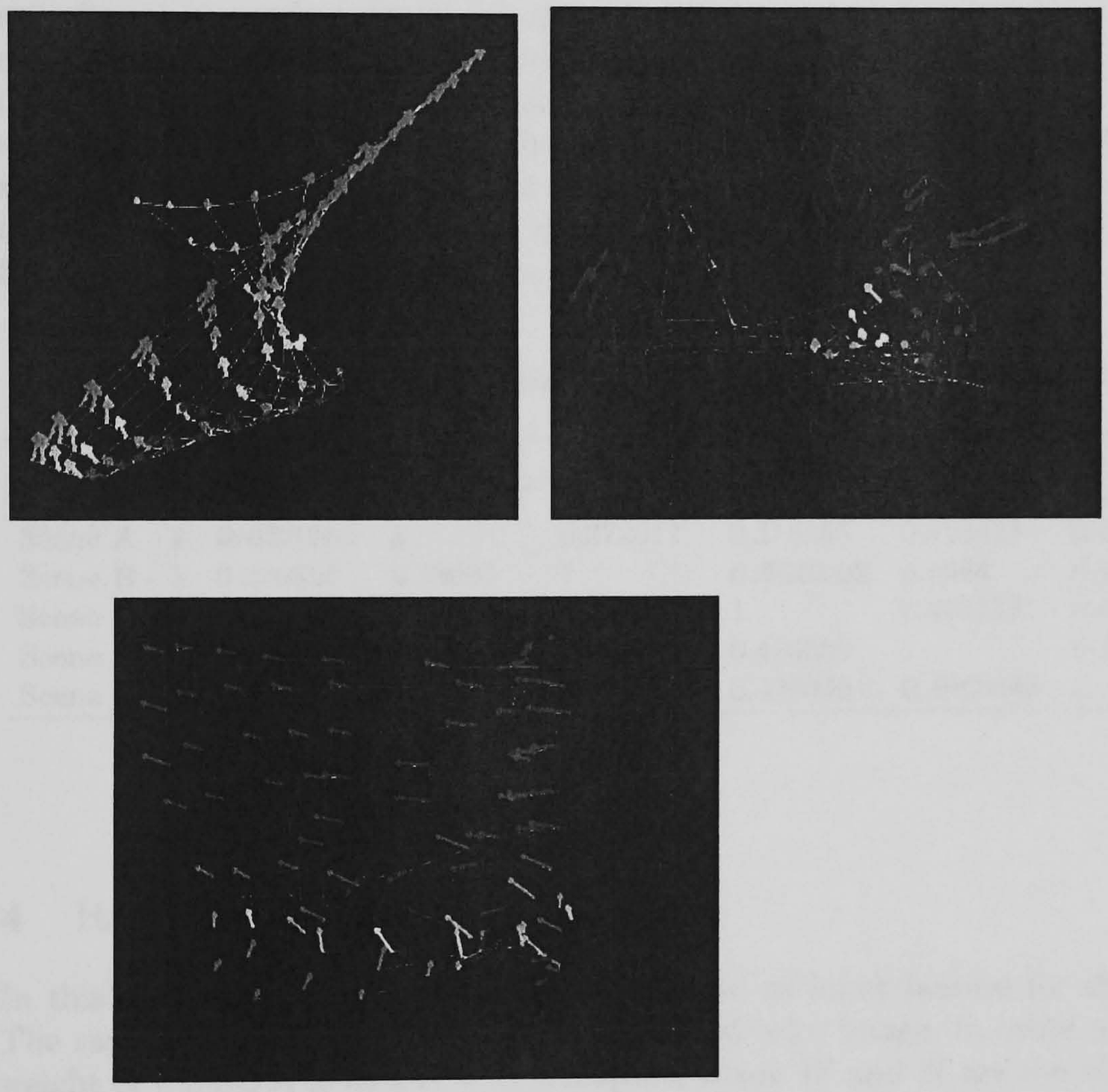


Fig. 3. Three different scenes and their visualization using SOM.

neuron in SOM A is not necessarily neuron (i). This can be caused by the situation that the neurons from SOM A and B are not in the same value scale. In some extreme case, all the neurons in one SOM could even be matched to the same neuron in the other SOM. As a result another interesting figure is the number of matched neurons in SOM B. The figure will indicate if the values of weights in SOM A are in the same scale of SOM B. It is combined with the last measure by:

$$S = \frac{Sim + N_{matched}/N_{total}}{2} \tag{8}$$

The comparison is not symmetric also means that if SOM A is comparing with SOM B, the result will be different from using SOM B to compare with SOM A. Consequently, two similarities are generated the comparison of two SOMs. This experiment basically takes three scenes, and two sequences are extracted from each scene, so that there are 6 sequences in total in the experiment. The following confusion matrix illustrates the relative results. In Table 3.2, each row has the similarity value of a SOM with the other sequences. There are two values: i.e. similarity of SOM A comparing to SOM B and similarity of SOM B comparing to SOM A. The values above 0.5 are in bold font in the table, and they are all from the video sequences from the same scenes.

Table 1. Confusion matrix of SOMs from different scenes (Scn abbreviates Scene)

	Scn A - 1	Scn A - 2	Scn B - 1	Scn B - 2	Scn C - 1	Scn C - 2
Scene A - 1	1	**0.653097**	0.484192	0.433234	0.468101	0.458993
Scene A - 2	**0.633261**	1	0.372017	0.315155	0.426438	0.400024
Scene B - 1	0.330897	0.33033	1	**0.645102**	0.4264	0.465297
Scene B - 2	0.35838	0.332804	**0.641464**	1	0.467114	0.455613
Scene C - 1	0.369259	0.400326	0.443745	0.426589	1	**0.715606**
Scene C - 2	0.366577	0.318921	0.414272	0.429349	**0.687943**	1

4 Raw image as Input

In this application the whole image is regarded as input feature for the SOM. The raw data has been used with three channel color image. In other word, the weight of the SOM is in a $W \times H \times 3$ space, where W and H are the width and the height of the image, respectively. Dimensional of the input video data are reduced from $W \times H \times 3$ (Image space) to $(n \times n)$ (lattice space).

The neurons of the SOMs retain the different status of the particular scene. Some selected neurons from SOM constructed by raw images are illustrated in Figure 4. The neurons illustrated the different crowd status of the square, and also some trajectories of the crowd.

The above experiment is carried out over video sequence which contains only one single crowded scene. In the following experiment, the SOM is built from

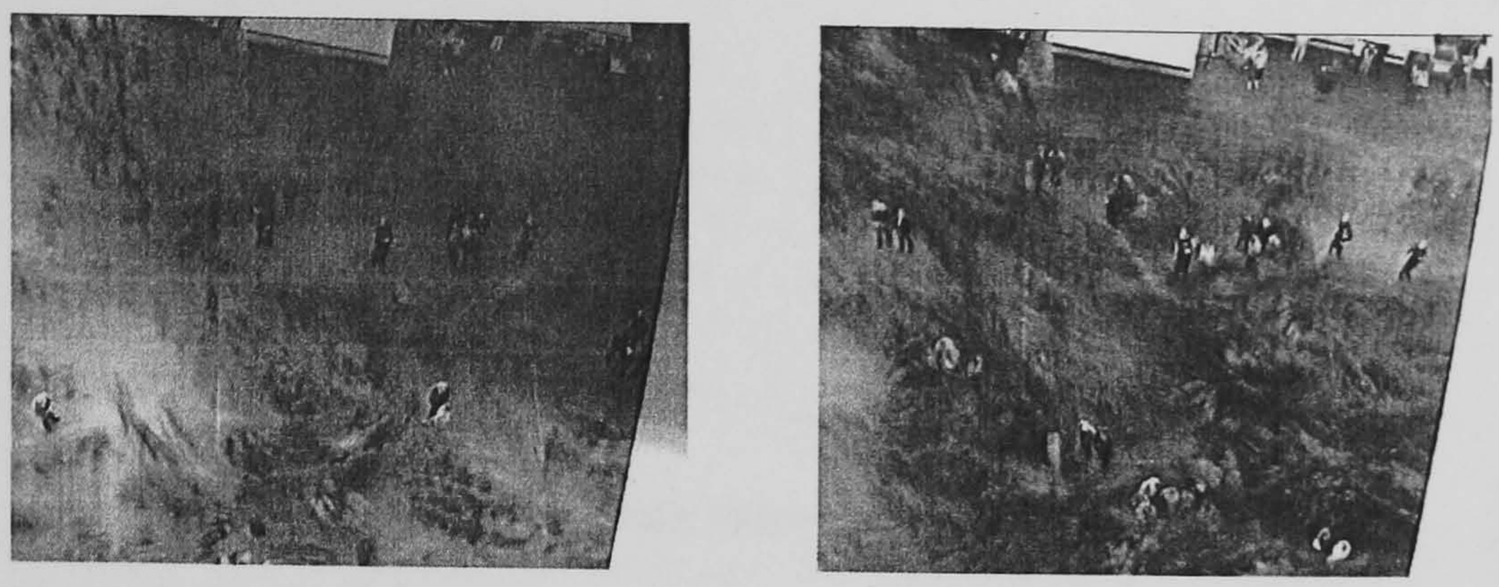


Fig. 4. Selected SOM neurons built from a single scene

video sequences consisting of more than one crowd scene. Figure 5 shows two neurons from the SOM built by a video sequence which contains two different crowded scenes. The built SOM has modeled the two different scenes (Example frames from the two scenes can be found in Figure 2(a) and 2(b)).

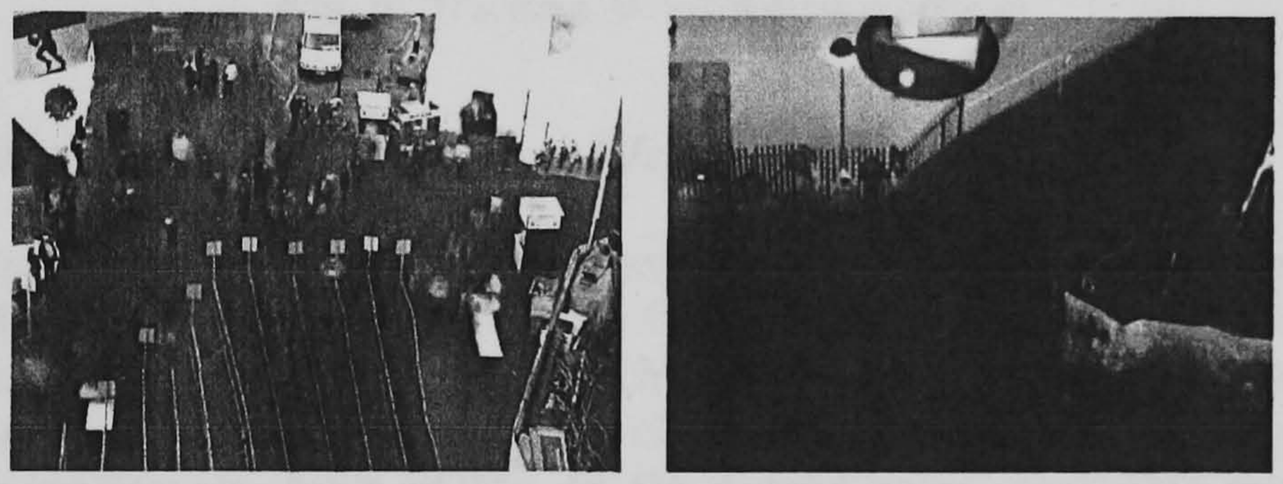
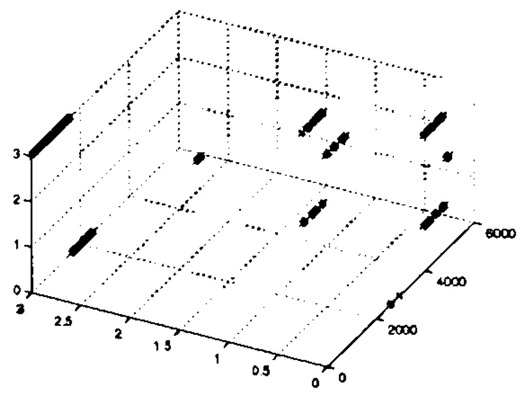
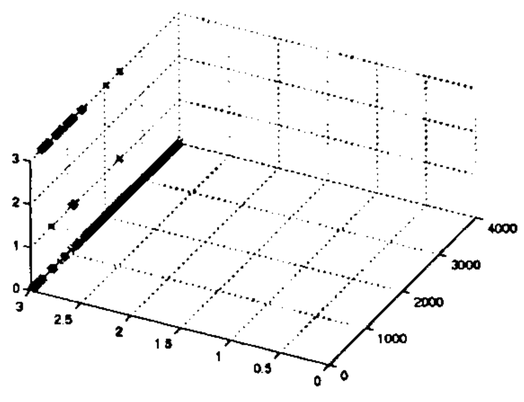


Fig. 5. Two neurons from SOM built by a video sequence which contains scene A and B

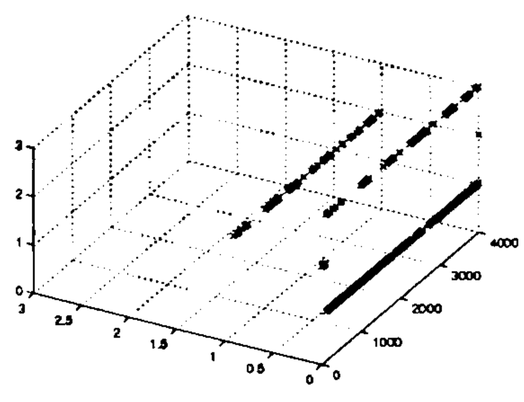
Different neurons of the SOM represent different dynamics in the video sequence. The tracking of the winning neurons indicates the transition between dynamics. Figure 6(a) shows the changes of the winning neurons on the SOM lattice when using the training video sequence (the coordinates are shown by the left side vertical plane. The axis with numbers from 0 to 5000 is the time line.) There is an obvious transition between the winning neurons in the middle of the time line where it represents the changing of the scenes. New image sequences from the two scenes are used as input to the SOM to test its ability of scene classification. Figure 6(b) and 6(c) are the result of tracking the winning neuron over time. The winning neurons of the first scene are on the same plane, and for the second scene the winning neurons never get to the previous plane.



(a) Train video sequence



(b) Test video sequence from scene A



(c) Test video sequence from scene B

Fig. 6. Tracking of the winning neuron

5 Conclusion and Future Works

This paper presented crowd analysis work using Self-organizing Maps. experiments were carried out testing optical flow and raw images to train the SOM. In the first case the built SOM of each crowded scene shows its capability of capturing the major dynamics. Scene classification is carried out by quantitatively comparing the built SOMs. In the second cases, the SOM can capture major dynamics from more than one scene. Experiment shows that the frames from different scenes activate the neurons from different locations of the lattice so that they can be labeled and classified. This work is, to our knowledge, the first attempt to employ SOM in crowd analysis applications. It reveals the great potential of SOM in handling this problem. More experiments, for example with different input features can be carried out. Also a deeper analysis of relationships between neurons can be involved to build a better model of the dynamics.

Acknowledgement

The authors wish to thank to Legion Group plc who provided the video data used the experiments in Section 2.

References

1. Legion: (Legion group plc) http://www.legion.biz/about /index.html.

2. Venegas, S., Knebel, S., Thiran, J.: Multi-object tracking using particle filter algorithm on the top-view plan. Technical report,LTS-REPORT-2004-003, EPFL (2004) http :// infoscience.epfl.ch / getfile.py? mode= best & recid=87041.
3. Cai, Y., de Freitas, N., Little, J.J.: Robust visual tracking for multiple targets. In: European Conference on Computer Vision. Volume 3954 of LNCS., Springer (2006) 107–118
4. Karlsson, R., Gustafsson, F.: Monte Carlo data association for multiple target tracking. Target Tracking: Algorithms and Applications (Ref. No. 2001/174), IEE **1** (2001)
5. Zhan, B., Remagnino, P., Velastin, S., Bremond, F., Thonnat, M.: Matching gradient descriptors with topological constraints to characterise the crowd dynamics. Visual Information Engineering, 2006. VIE 2006. IET International Conference on (2006) 441–446 ISSN: 0537-9989, ISBN: 978-0-86341-671-2.
6. Zhan, B., Remagnino, P., Velastin, S.A., Monekosso, N., Xu, L.Q.: Motion estimation with edge continuity constraint for crowd scene analysis. In Bebis, G., Boyle, R., Parvin, B., Koracin, D., Remagnino, P., Nefian, A.V., Gopi, M., Pascucci, V., Zara, J., Molineros, J., Theisel, H., Malzbender, T., eds.: ISVC (2). Volume 4292 of Lecture Notes in Computer Science., Springer (2006) 861–869
7. Andrade, E., Fisher, R.: Modelling crowd scenes for event detection. In: Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)-Volume 01, IEEE Computer Society Washington, DC, USA (2006) 175–178
8. Andrade, E., Fisher, R.: Hidden Markov models for optical flow analysis in crowds. In: Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)-Volume 01, IEEE Computer Society Washington, DC, USA (2006) 460–463
9. Andrade, E.L., Blunsden, S., Fisher, R.B.: Performance analysis of event detection models in crowded scenes. In: Proc. Workshop on Towards Robust Visual Surveillance Techniques and Systems at Visual Information Engineering 2006, Bangalore, India (2006) 427–432
10. Zhan, B., Remagnino, P., Velastin, S.: Analysing Crowd Intelligence. Second AIxIA Workshop on Ambient Intelligence (2005)
11. Zhan, B., Remagnino, P., Velastin, S.: Visual analysis of crowded pedestrain scenes. XLIII Congresso Annuale AICA (2005) 549–555
12. Zhan, B., Remagnino, P., Velastin, S.: Mining paths of complex crowd scenes. Lecture notes in computer science (2005) 126–133 ISBN/ISSN 3-540-30750-8.
13. Kirt, T., Vainik, E., Võhandu, L.: A method for comparing self-organizing maps: case studies of banking and linguistic data. In: Eleventh East-European Conference on Advances in Databases and Information Systems ADBIS, Varna, Bulgaria: Technical University of Varna (2007) 107C115
14. Lefebvre, G., Laurent, C., Ros, J., Garcia, C.: Supervised Image Classification by SOM Activity Map Comparison. Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)-Volume 02 (2006) 728–731
15. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall PTR Upper Saddle River, NJ, USA (1994)
16. Polani, D.: Measures for the organization of self-organizing maps. Self-Organizing neural networks: recent advances and applications (2002) 13–44

A quantitative comparison of two new motion estimation algorithms

B.Zhan, P.Remagnino, S.A.Velastin, N.Monekosso[1],L-Q.Xu[2]
{B.Zhan, P.Remagnino}@kingston.ac.uk
[1] Digital Imaging Research Centre, Kingston University, UK [2] British Telecom Research,Ipswich,UK

Abstract. This paper proposes a comparison of two motion estimation algorithms for crowd scene analysis in video sequences. The first method uses the local gradient supported by neighbouring topology constraints. The second method makes use of descriptors extracted from points lying at the maximum curvature along Canny edges. Performance is evaluated using real-world video sequences, providing the reader with a quantitative comparison of the two methods.

1 Introduction and Previous Work

A pedestrian scene normally introduces different levels of difficulty, e.g. for visual surveilance the number of pedestrians in the field of view could varies from one to several hundred. When the scene is very complex, occlusions make it virtually impossible not only to track individuals but also to estimate stochas-tic background model. Robust and reliable descriptors must be employed to describe atomic features of underlying dynamics. The descriptors must be able to work under different levels of difficulty to extract the motion information, so as to enchance the higher level analysis of the scene dynamics. Extensive literature exists on local descriptors that have been used in in deformable object tracking [1], image retrieval applications, e.g. SIFT[2], Harris Corner[3]. Recently years it has also been applied to analysis of image sequences in sport and monitoring applications [4] [5]. Two novel motion estimation methods extended the usage of such local descriptors to esimate crowd motion are validated and compared in this paper. In both of the algorithms, constraints are applied to improve the robustness of the matching between individual descriptors. The first algorithm checks locally spatial temporal consistency of color gradient supported by local topology constraints and the second one uses the points of local extreme curvature along Canny edges and applies contour constraints. Overall dynamics of the scene can then be leant from the short term motion estimated from these algorithms. Section 2 describes the two algorithms , Section 3 gives the experimental results of both algorithms, and Section 4 draws some conclusions.

2 The Proposed Algorithms

Video sequences involving crowds are very complex to analyze, mainly because conventional background subtraction algorithms and motion estimation methods might not deliver expected results. Given the problem of interpreting crowd motion, we choose to use refined matching of local image descriptors to derive the dynamics features. Our aim is to recognise points of interests which can be tracked for some periods of time, and then combine their tracks into meaningful crowd trajectories. The following two subsections summarize the two algorithms, published in [6] [7].

2.1 Algorithm 1

The first approach, proposes a topological matching of interest points extracted in the evolving scene. Points of interest are extracted using a color variant of the Harris detector as described in [3]. The matching between frames is carried out in two steps: (i) searching of the candidate matching points by similarity and then (ii) applying the topological constrains. Briefly, for each selected interest point in the reference image, corresponding candidate matching point is searched in the matching image inside a given search window using the gradient similarity measure given by $sim(a,b) = \frac{min(R_a,R_b)}{max(R_a,R_b)}$ (given a, b as two interest points). This basic matching does not provide a robust solution, due to the instability of interest points in a highly complex scene. Frequent occlusions reduce the probability of identifying correct matches and, without local support, similar gradient localities might be found as acceptable matches generating false positives. To tackle this problem, we introduced an effective topological constraint into the search process. The necessary local support is derived from local window centred at the interest point and we make use of the relative location of the interest points in such window. Support is estimated for the matched interest point pair inside the support windows. All interest points found in the support window are then matched. Support is then quantified in terms of the error by measuring the standard deviation of the ensemble of found correspondences (see [6] for more details on the algorithm).

2.2 Algorithm 2

The results of *Algorithm*1 for crowd dynamics measurement turns out to be good, however there is still a room for further improvement, as certain false positives still exist. The reason could be that the relationships between the interest points provided by topological constraints are still not very reliable, so we proposed another method [7] using local descriptors which provide additional information. We choose Canny edge information and extracted the curvature along each edge to retrieve descriptor points (those with local maximum curvature) as salient features along an edge. Besides the points, edge information is maintained by "edgelet constraint" to refine the estimate. Thus, we combine the

advantage of using point features that are flexible to track with the advantage of using edge features that maintain structural information. Each Canny edge is a chain of points S_p, and all the edges are stored in an edge list L_p. It can be observed that even in a scene which depicts a crowd of moderate density, edge chains can occlude one another, increasing the descriptor matching complexity. Given two consecutive frames, we estimate the motion of the extracted local descriptors (matching within a window of interest).The best n matches are then selected as candidate points and considered for the next step. For an image frame, we divide every chain S to a uniform length edgelets represented by sub sequences E. There are two reasons for doing this: to avoid a very long edge that could be generated by several different objects, and to enhance the matching of the edge fragments that are generated by occlusions. For each local descriptor in E we have as result from the first step, the n candidate matching points. Each candidate point belongs to a sub sequence S. To find the best match of E, we use three parameters: energy cost, variation of displacements and the match length for each candidate and combine them into a single matching score. Here we assume that the length of E is small enough so that it would not split again to two or more matches. This is so that their candidate points correspond to the same candidate sequence. Details on the second algorithm can be found in [7].

3 Comparison of the two methods

The two motion estimation algorithms are tested using three sequences taken from crowded public space and quantitative results are generated. As we are concerned about extraction of short term motion instead of tracking, measures are defined based on two consecutive images instead of whole sequence. In the following we first give a brief description of the test dataset used in our experiments, then go through the details of the testing methods adopted and explain the results generated from the tests; some visual results are also included at the end of the section.

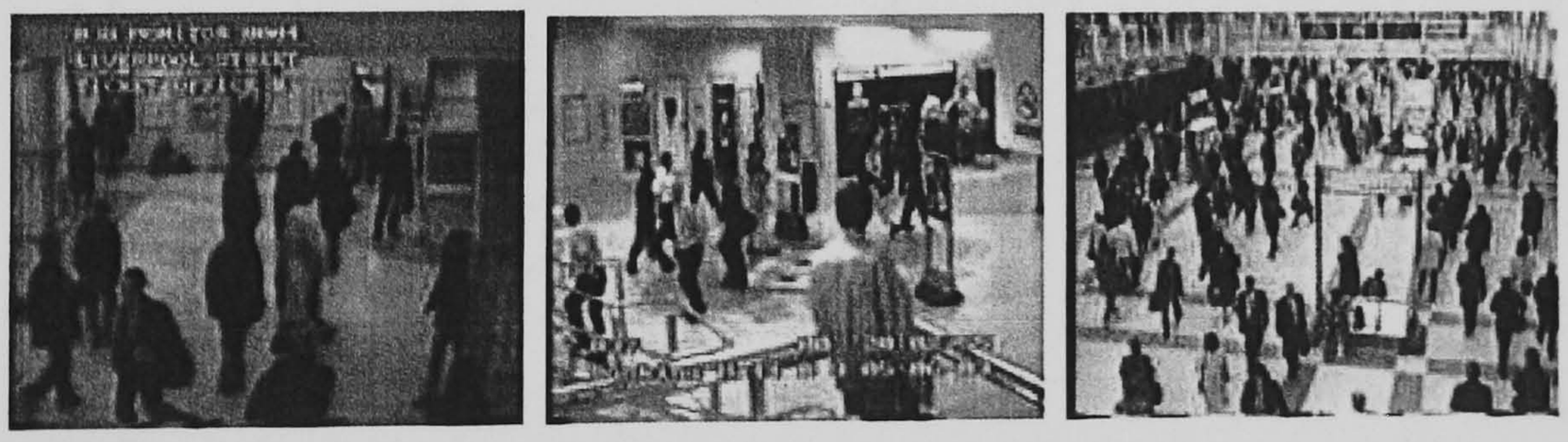


Fig. 1. Sample frames from 3 testing sequences

3.1 Testing data

Sample frames from the three sequences are shown in Figure 1: sequence 1 (left) is a mid field scene with people scattered across the field of view; sequence 2 (middle) is a mid field scene with major motions taking place in certain areas; sequence 3 (right) is a far field scene with pedestrians present in all parts of the field of view, with some predominant trajectories.

3.2 Testing based on local descriptors

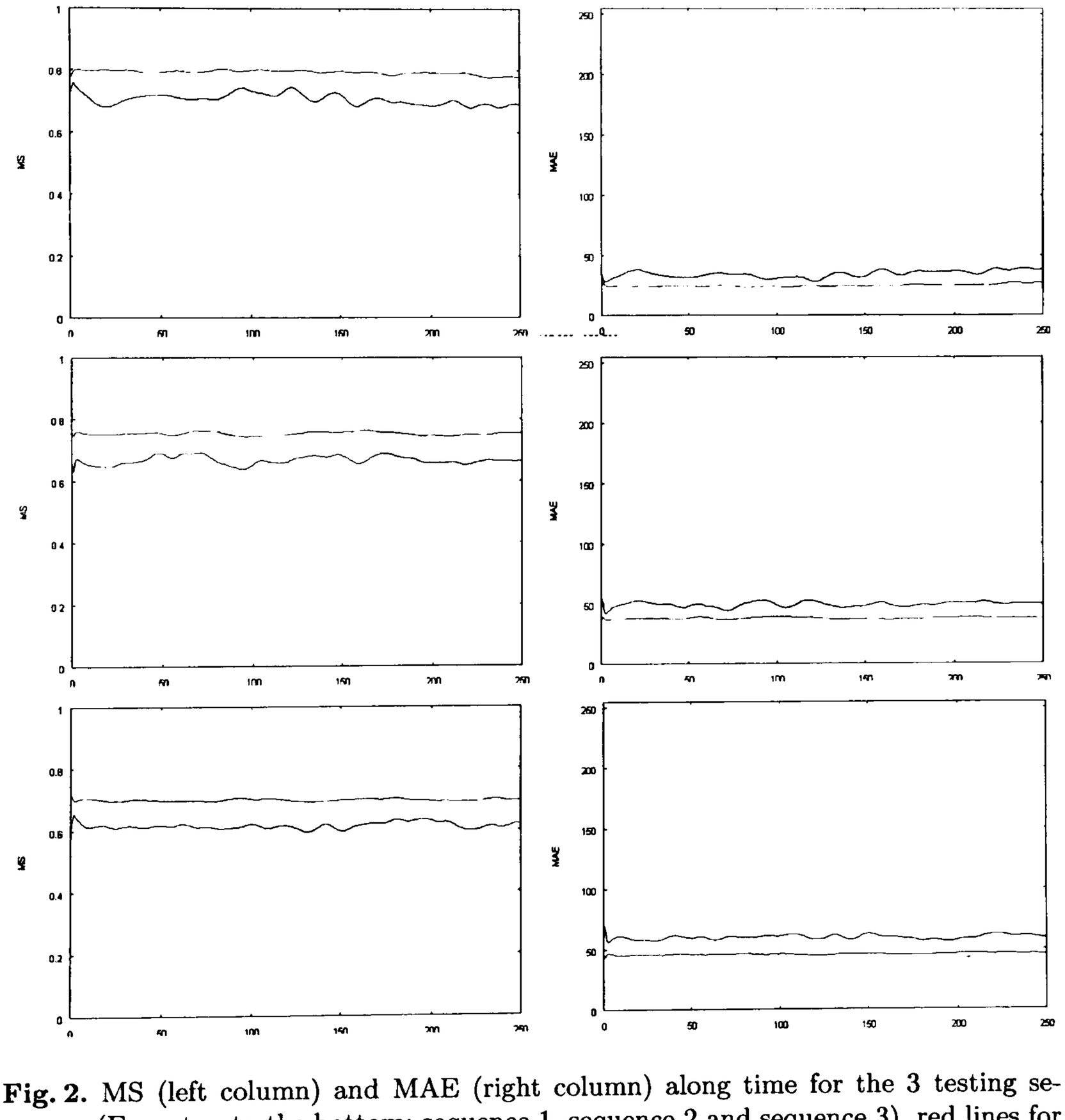


Fig. 2. MS (left column) and MAE (right column) along time for the 3 testing sequence (From top to the bottom: sequence 1, sequence 2 and sequence 3), red lines for Algorithm 1; green lines for Algorithm 2. Algorithm 2 keeps higher in MS and lower in MAE.

In this testing only the quality of matching of individual local descriptors is considered. For each pair of consecutive frames, local descriptors in the initial frame are compared with their corresponding local descriptors, found by the two presented algorithms, in the target/second frame, respectively. Two measures, Mean Similarity (MS) and Mean Absolute Error (MAE), are used here. MS is designed to assess the relative similarity of the matched local descriptors.

$$MS = \frac{1}{n}\sum_{i=0}^{n}\frac{min(X_i^{to}, X_i^t)}{max(X_i^{to}, X_i^t)} \quad (1)$$

Where n is the total number of the local descriptors in the initial frame. MS is defined as the average of the similarity, and the similarity is calculated by the minimum of the two matched local descriptors' pixel value divided by the maximum. The result is a value which falls in the $(0, 1)$ range. Another measure, MAE is commonly used for the testing of motion estimation algorithms [8] as it returns an error measure. MAE is defined as follows

$$MAE = \frac{1}{n}\sum_{i=0}^{n}\|X_i^{to} - X_i^t\| \quad (2)$$

Where X_i^{to} is the pixel value at the i^{th} corner in the first frame, and X_i^t is the cor-respond local descriptor in the next frame.

The images in Figure 2 represent the plots of MS and MAE for the two algorithms tested against the three sequences. MS and MAE are calculated every frame along the sequence. In each plot the x axis represents time (the number of the frame) and the y axis represents the values of MS and MAE, respectively. Hence, for the two algorithms the MS and MAE for the three testing sequences are both good, though in most of the cases the second algorithm has a higher MS and a lower MAE. Also, along the time scale the MS and the MAE produced from the first algorithm fluctuate a lot while the second one produces more stable results. It can be concluded that the second algorithm has a more desirable performances than the first one.

3.3 Testing based on Motion Connect Component

The testing here makes use of connected components algorithm based on motion vectors (so called MCC – Motion Connected Component). The algorithm groups together motion vectors that are in close proximity and have common motion properties. The result of the MCC algorithm segments the motion field into clusters of uniform motion group (e.g. a (part of) pedestrian or a group of pedestrians), and the testing is based on each MCC to assess the two algorithms. In order to assess the two algorithms with MCC, we adopted two measures which are used in evaluating search strategies: Recall and Precision. Recall is the ratio of the number of relevant records retrieved to the total number of relevant records in the database. Precision is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved [9].

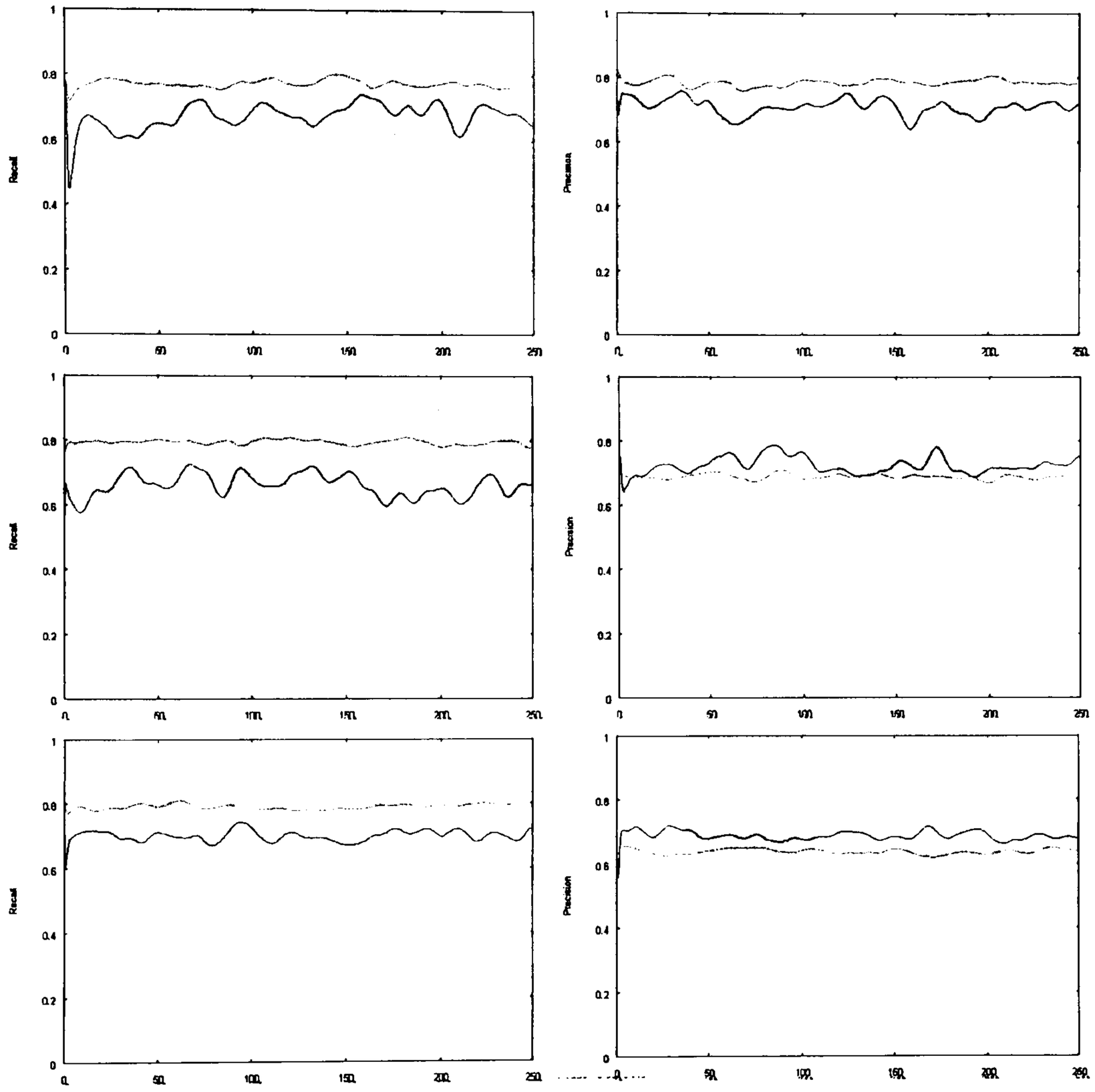


Fig. 3. Recall (left column) and Precision (right column) along time for the 3 testing sequence (From top to the bottom: sequence 1, sequence 2 and sequence 3), red lines for Algorithm 1; green lines for Algorithm 2. Algorithm 2 has higher values of Recall.

In the proposed implementation we take the bounding box of each MCC, and calculate the average motion of MCC, thus map the bounding box to the next frame. The number of "relevant records in the data base" should be the number of local descriptors of the MCC in the initial frame (N_{t_0}) while the number of "retrieved records" should be the number of local descriptors in the mapped bounding box in the second frame (N_t). The definitions of the two measures are given by

$$Recall = \frac{N_{t_0} \cap N_t}{N_{t_0}} \tag{3}$$

$$Precision = \frac{N_{t_0} \cap N_t}{N_t} \tag{4}$$

Both of the values are in the range $[0, 1]$. For every frame an average Recall value and an average Precision value are calculated. Figure 3 gives the plots of Recall and plots of Precision; the layouts of these plots remain similar to the previous ones, though the y axis represents Recall and Precision, respectively. From the plots again we can claim the results of Recall and Precision of both of the algorithms, especially the results of Recall, are satisfied. It can be observed that the results of Precision for sequence 3 is lower than the other three, one possible reason could be that as sequence 3 is a far field view for a crowded scene, when mapping the bounding box of the MCC to the second frame local descriptors of other MCC could be included and noise could be introduced. When comparing the results of Recall, it can be seen values for Algorithm 2 are always higher, though for sequence 2 and sequence 3 Precision values for Algorithm 1 are slightly higher. Here another measure should be taken into consideration, which is the number of the MCC detected by each algorithm. According to the plots in Figure 4, in sequence 1 the average number of MCC detected by Algorithm 1 is around 20, while by Algorithm 2 the number is around 100; in sequence 2, the numbers are around 20 and 200, respectively; in sequence 3 the numbers are around 40 and 280, respectively. Algorithm 2 detects much more MCC, especially for sequence 2 and 3. Due to the above fact and the fact Algorithm 2 produces higher Recall, it can be deduced that the slight drawback of the Precision only indicates more noise has been introduced to our assessment.

4 Conclusions

Two novel algorithms to estimate the motion of a crowd in complex scenes are presented, evaluated and compared in this paper. The two algorithms are compared using three surveillance video sequences and quantitative results are generated based on individual local descriptor and MCC (Motion Connected Component). MS and MAE are used as criteria for local descriptor based assessment. The values of MS generated by the two algorithms are all above 0.6 and for Algorithm 2 the values are all above 0.7. For the values of MAE, those generated by Algorithm 2 are always below those generated by Algorithm 1. In the MCC based assessment, for ratio of Recall almost all of the values generated by

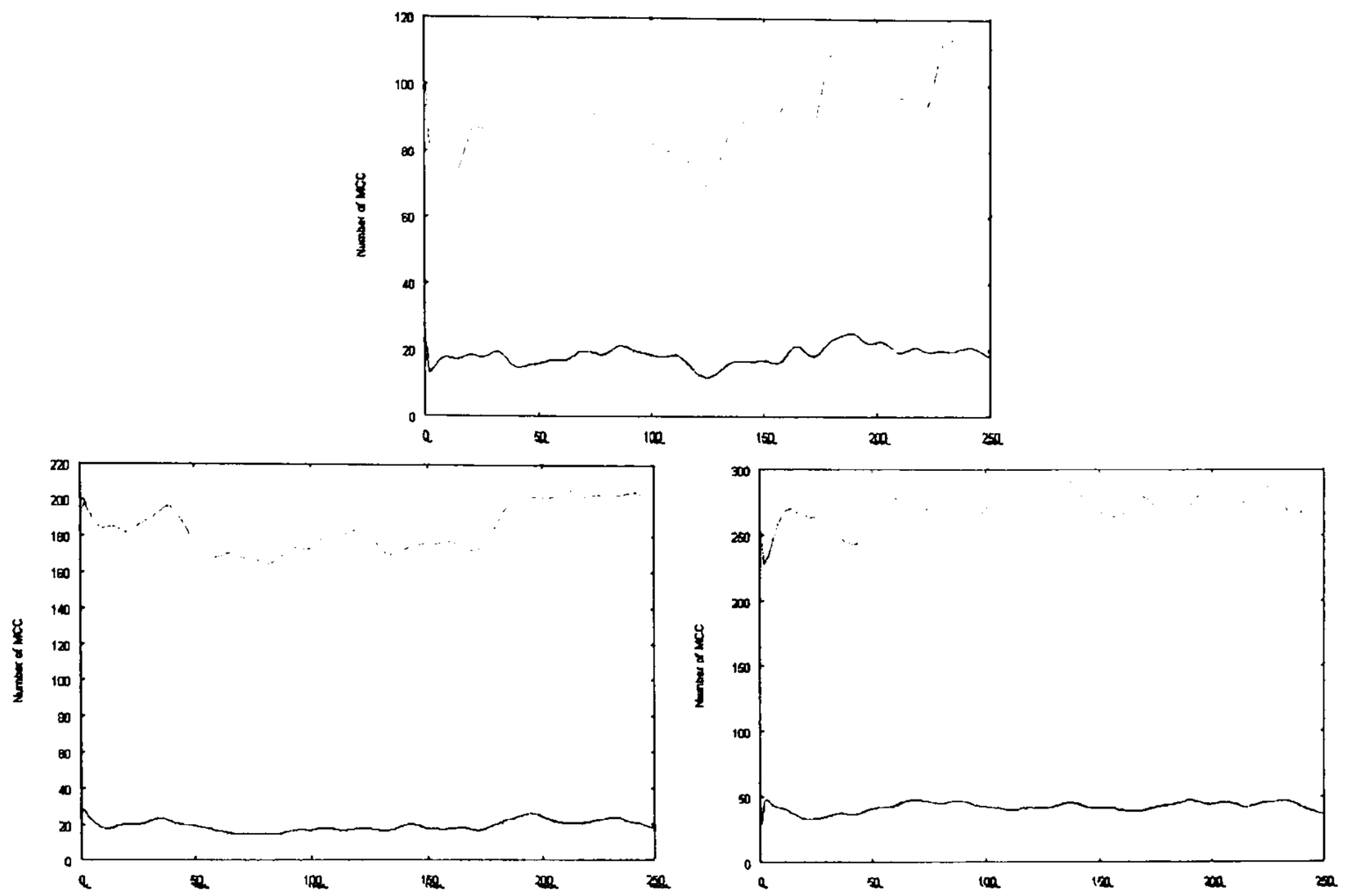


Fig. 4. Number of MCCs along time for the 3 testing sequence, red lines for Algorithm 1; green lines for Algorithm 2 (From top to bottom: sequence 1, sequence 2 and sequence 3). Algorithm 3 detects much more MCCs for all of the three video sequences.

Algorithm 1 are above 0.6 while those generated by Algorithm 2 are close to 0.8; and for ratio of Precision, the values generated by both two are above 0.6. We can conclude that the experimental results show the Algorithm 2 works better with most of the experimental sequences while both outcomes are acceptable. The crowd dynamics estimation provides a suitable precursor to processes for learning modes of complex dynamics, describing behaviour and supporting for work in high-level vision and socio-dynamics modelling. Based on the two algorithms presented, our future work will be focused on developing novel method of building mature and reliable crowd dynamics model through computer vision.

Acknowledgement

The research described in this paper is partially supported by the British Telecom Group PLC.

References

1. Cohen, I., Ayache, N., Sulger, P.: Tracking points on deformable objects using curvature information. Proceedings of the Second European Conference on Computer Vision (1992)

2. Lowe, D.G.: Object Recognition from Local Scale-Invariant Features(ICCV'99) . Seventh International Conference on Computer Vision **2** (1999)
3. Gouet, V., Boujemaa, N.: About optimal use of color points of interest for content-based image retrieval. Technical Report (2002) RP–4439
4. Gabriel, P., Hayet, J., Piater, J., Verly, J.: Object tracking using color interest points. Proceedings. IEEE Conference on Advanced Video and Signal Based Surveillance (2005) 159–164
5. Mathes, T., Piater, J.: Robust non-rigid object tracking using point distribution models. Proc. of British Machine Vision Conference (BMVC) **2** (2005)
6. Zhan, B., Remagnino, P., Velastin, S., Bremond, F., Thonnat, M.: Matching gradient descriptors with topological constraints to characterise the crowd dynamics. Visual Information Engineering, 2006. VIE 2006. IET International Conference on (2006) 441–446 ISSN: 0537-9989, ISBN: 978-0-86341-671-2.
7. Zhan, B., Remagnino, P., Velastin, S.A., Monekosso, N., Xu, L.Q.: Motion estimation with edge continuity constraint for crowd scene analysis. In Bebis, G., Boyle, R., Parvin, B., Koracin, D., Remagnino, P., Nefian, A.V., Gopi, M., Pascucci, V., Zara, J., Molineros, J., Theisel, H., Malzbender, T., eds.: ISVC (2). Volume 4292 of Lecture Notes in Computer Science., Springer (2006) 861–869
8. Subramanya, S.R., Patel, H., Ersoy, I.: Performance evaluation of block-based motion estimation algorithms and distortion measures. In: ITCC '04: Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04) Volume 2, Washington, DC, USA, IEEE Computer Society (2004) 2
9. http://www.hsl.creighton.edu/hsl/Searching/Recall-Precision.html.

Skin Patch Trajectories as Scene Dynamics Descriptors

B.Zhan, N.D.Monekosso and P.Remagnino
Kingston University
Surrey, UK
{B.Zhan,P.Remagnino}@kingston.ac.uk

T.Rukhsana, A.Mansur and Y.Kuno
Saitama University
Saitama, Japan
kuno@cv.ics.saitama-u.ac.jp

Abstract

There is an increasing interest in the concept of intelligent environments where a closed or delimited public space (shopping mall, station, museum, hospital etc) is endowed with some automatic ability to interpret human behavior. Intelligent environments interact with their users, aiding, serving and pre-emptying them. In a not too distant future, this paradigm - in Europe called ambient intelligence - will soon include robotic platforms. Both intelligent environment and robotic platforms will collaborate to better inform the inhabiting or visiting user. This paper presents some steps towards that direction, describing a study on some scene descriptors, which can be employed to provide an automatic interpretation of the clutter and dynamics of a complex scene.

1. Introduction

The main goal of this research is to estimate automatically the amount of clutter and the level of dynamics in a complex scene, frequented by an unspecified number of people.

This is important in applications where situation assessment is crucial to better inform people inhabiting a specific environment. For instance, in a shopping mall or in a museum, individuals and more or less large groups of people might pass or stop by to window shop or to observe an exhibit. In such cases information about merchandise or exhibit could be delivered in a more efficient manner, for instance with the aid of a robot.

An automatic estimation of clutter and dynamics is also important in crucial situations, where people must be informed of exits and escape routes.

What in Europe is now called ambient intelligence and in the United States goes under the name of smart or intelligent environments, is a paradigm which has been in the mind of artificial intelligence researchers for some time [9]. The idea is of a *living* environment, able to interact with the user to make their lives easier. The emerging phenomenon of robotic platforms, seen more as *companions* than mechanical machines, inspires the idea of a living environment, where both the surroundings and robots collaborate between them and with the user to improve productivity (factory or office), security (public space), safety (nursing home or hospital). This *symbiotic* existence actively assists the user, seen either as a casual passenger or pedestrian in the environment, or as a frequent visitor (station or shopping mall) or even the person inhabiting (home) the intelligent space.

This paper presents a method to estimate dynamics, offering a means to evaluate its amount and classify people behavior as interested or uninterested in the scene. One can them imagine the degree of interest in a scene being used to inform a robotic platform to deliver a specific message to the user.

The next sections describe the proposed method and illustrate some examples of how it could be employed to assess a situation.

2. Methodology

Conventional cameras, used in museums and shopping malls can capture full human figures and sometimes human faces. Video data of this kind can be employed to recognize and track people in a complex environment. Full figure chromatic and structure models can be built [7], and people physiognomy, gait and shape characteristics have indeed been used to suit this purpose. In this paper we use skin color to extract exposed patches of the human body figure and we show that those can be robustly tracked throughout a scene. The tracks are then employed to annotate the dynamics of individual patches and draw some qualitative and quantitative description of the global evolution of the scene.

The following sections describe in detail our method, employed to extract and track skin color patches, and estimate trajectory trends.

2.1. Robust skin color detection

This part of our method makes use of the already proved idea that skin color can be indeed modeled across races so long as a suitable color space is employed. In our experiments we convert the color frames from the RGB to the YUV space. The choice of the YUV color space is justified by a fast conversion and the factorization of chromatic and illumination features. The illumination Y component is easily factored out, and the chromatic UV plane is employed to estimate the skin color model (an example of color probability density function – PDF - of skin color is shown in Figure 1).

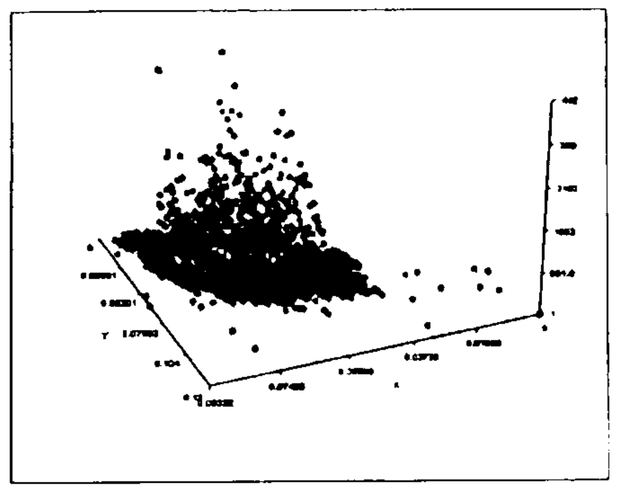


Figure 1: The PDF of skin color patch.

As already shown in other research, a skin color model can be estimated by acquiring video data of skin color patches and via the training of a color model using the expectation maximization algorithm [8]. In order to optimize the model, the skin color data can be studied and an optimal initialization defined in terms of number of clusters and initial positions and approximating functions.

Figure 1 illustrates the PDF of image data used to train the skin color model. All data is constrained in a small region of the $[u,v]$ plane and a mixture of Gaussians $\{\mu_i, \Sigma_i\}$ can serve as a good explicit approximation of the distribution.

Figure 2: PDF of data used to train the skin color model.

Figure 2 illustrates the probability masks of some of the data used to build the skin color model. The model is fairly robust to changes in illumination but it has the weakness of being specific to the camera used to acquire the training data. In all our tests, each new video camera we have used to acquire video footage had its own color model. As the training can be performed off line, the limitation is not prohibitive.

2.2. Color patch tracking

The MEANSHIFT method, based on an old idea of Fukunaga [1] and resurrected by Cheng [2] and Cominiciu [3], has been proven very robust for the tracking of objects and people in cluttered scenes. The MEANSHIFT algorithm tracks an object by estimating the drift of the underlying density function representing the evolving process. The limitation of the MEANSHIFT method stems from its inability to deal with time varying density functions. The CAMSHIFT algorithm proposed in [4] adapts to evolving PDFs by alternating cycles of the MEANSHIFT algorithm with a resizing of the search window. The window size is a function of the center of mass of the probability density map (zeroth moment).

Figure 3: LEFT: frame with bounding rectangles of recognized skin patches and RIGHT: related probability map.

Tracking color patches entails running the CAMSHIFT algorithm for each patch. However, this is not sufficient to maintain hypotheses in a rapidly evolving scene. That is why our method keeps track of a list of *alive* patches, by tracking them throughout the scene with the CAMSHIFT algorithm, removing those which have too low a probability associated for a number of frames and introducing new patches, whenever sufficiently large new patches appear in the scene with a sufficiently high probability.

Figures 4 and 5 illustrate four frames where skin color patches are identified and tracked throughout the scene.

Figure 4: The two above frames show the low curvature trends of uninterested behavior: when people pass by an exhibit.

Figure 5: The above frames show when people are interested in the shown exhibits and they stop by the exhibit. Trends of such trajectories have higher curvature.

The trajectories of Figure 4 and 5 also illustrate two occasional problems: (i) FALSE NEGATIVES: sometimes not all exposed skin is recognized (because of the small size of the patch and because of the limitation of the color model) and (ii) FALSE POSITIVES: at times patches not of skin color are detected, these are commonly stationary objects and their stationary position can be used to eliminate misrecognition cases.

2.3. Dynamics estimators

Trajectories of skin patches identify people trajectories and can be seen as signatures of people behavior.

For instance, people interested in exhibits or merchandise have a more irregular signature, distinguished by curvature that becomes higher and changing more frequently, when the patches represent people looking at an object.

The amount of time spent in the scene also plays an important role: the shorter the time the smaller the interest shown in the exhibit/object. Frames in Figure 4 illustrate two examples of uninterested behavior, well correlated with a smoother (low curvature) trajectory, while Frame 5 clearly illustrates how the interest in an object is correlated with a change in curvature

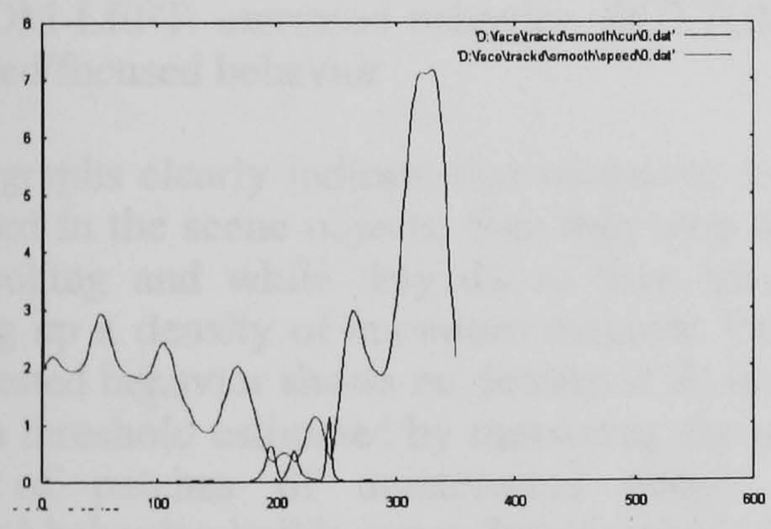
Dynamics can therefore be estimated by studying the trajectories of the tracked skin color patches and making use of their trends. An in depth study of the trajectories led us to the following conclusions, all based on the assumption that the extracted skin patches belong indeed to people in the scene:

- Average number of patches and their speed over a period of time can be used to estimate the entropy of the scene: the higher the number of patches the more people are in the scene and the histogram of speed values over time and its change illustrates the amount of movement in the scene (the flatter

the spread the higher the entropy),

- Fast patch movements indicate people in the scene are moving rapidly: the speed of each patch is estimated by the distance in pixels of a patch between frames,
- The curvature of a trajectory is a good indicator of how many twists and turns the trajectory trend has. Changes in curvature might occur more frequently in some moments then others: the frequency of change and the magnitude of curvature is an indicator of the person interest in some parts of the scene,
- A density signature of curvature peaks can therefore be estimated to describe people interest: the higher the density the higher the attention a person has for an object. Our study demonstrates that highly interested people will stop and move about in front of the object, uninterested or little interested people will move a lot in the scene and stop rarely and their curvature signature shows trends with small number of peaks of small number of high peaks. A suitable time window is defined to estimate the density: typically a number of seconds usually spent by a person to observe an object in the scene. This parameter depends on the application and can be learnt.

The figure below (Figure 6) illustrates the speed and curvature trends of a patch used to train the model of uninterested people. The speed becomes fairly high, however, the curvature remains lower than a low threshold; typically around 1.



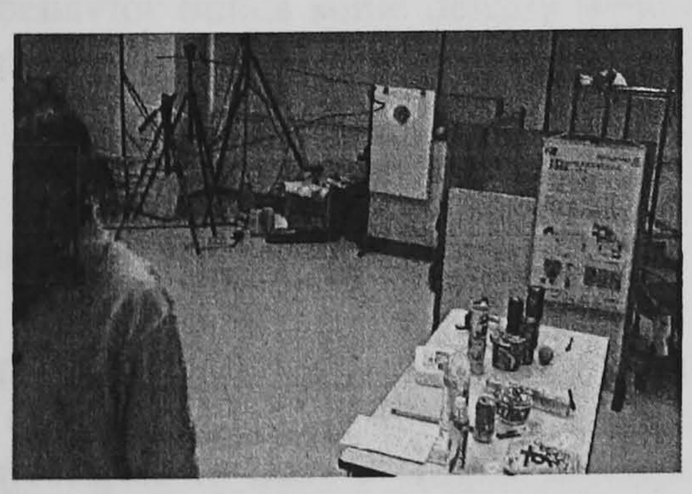
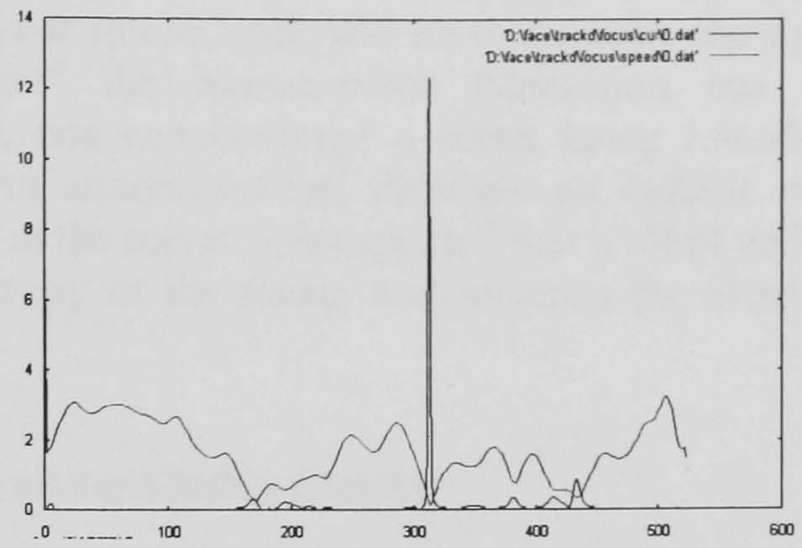


Figure 6: Speed (green) and curvature (red) of a patch of a person uninterested in the monitored scene.

The following graph (Figure 7) illustrates the signature of a patch related to a person who is interested in the scene. The speed is lower, indicating the person pays more attention to the scene. The frame in Figure 7 clearly shows the close occurrence of curvature peaks in two points of the scene, indicating that the person stopped, they looked around for a while, before moving to the next area of interest to stop again and observe, before leaving the scene. The yellow trajectory - associated with a hand – was picked up too late to illustrate the curvature phenomenon typical of an *interested* behavior.



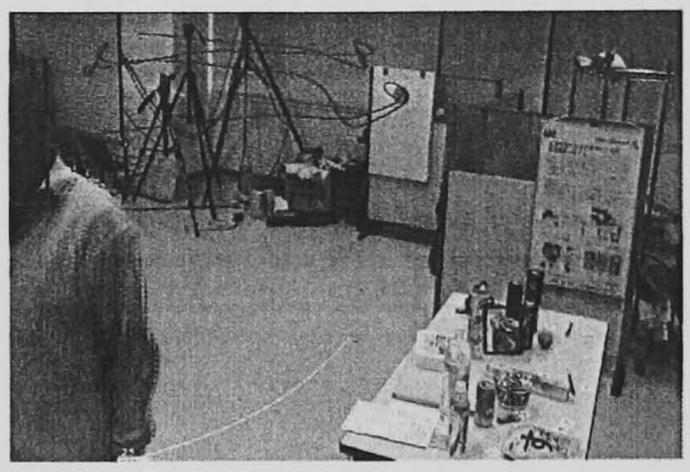
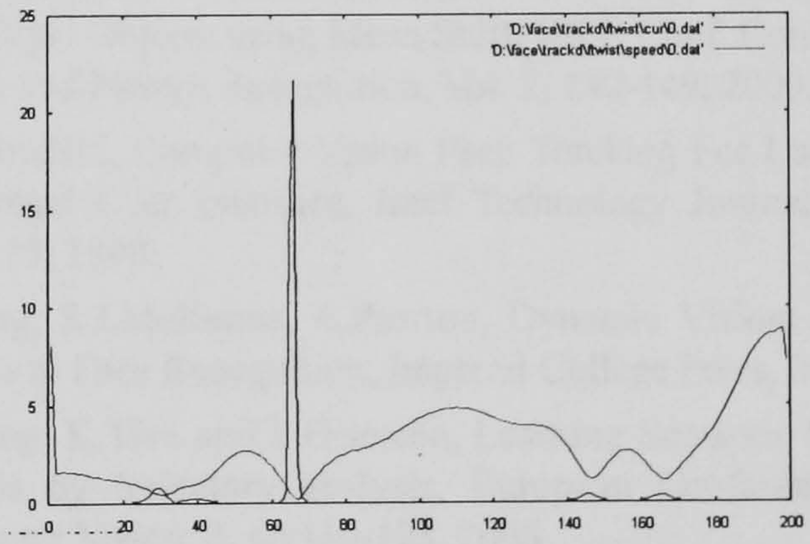


Figure 7: Speed (green) and curvature (red) of a path of a person interested in the object in the scene.

The following graph (Figure 8) illustrates what we call uninterested and animated behavior, characterized by patches of people uninterested in the scene objects, but where those people stay for longer in the scene and move about without really focusing on any object and they do not stand still in any particular position of the scene.



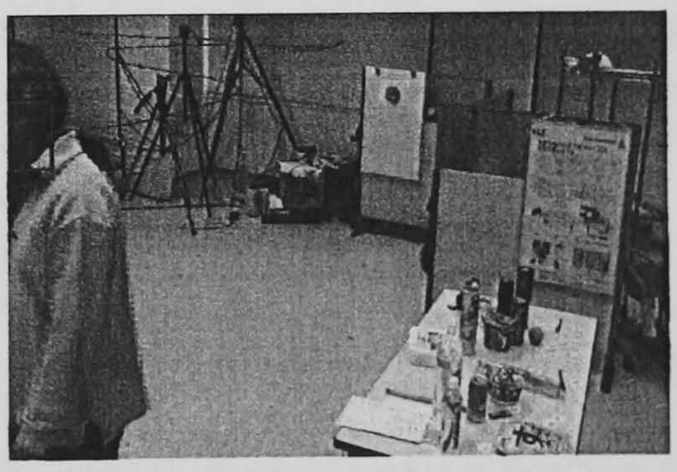


Figure 8: Speed (green) and curvature (red) of a patch of a person with animated behavior and uninterested in the scene.

As can be seen in the above graph, such behavior shows a large number of sparse high curvature peaks and it also correlates with a higher speed, indicating that the person did not stop for longer than the short period of time required to change direction in the scene a few times and

then leave the scene.

3. Trajectory classification

Experiments were run in a University laboratory and all scenes filmed from a single camera.

A number of experiments were run with individuals performing the same action repeatedly more times. Mixtures of actions were then recorded with more people in the scene performing either the same or different actions.

The following graphs (Figure 9) illustrate how the density of curvature maxima can be employed to disambiguate between an interested, uninterested and animated behavior.

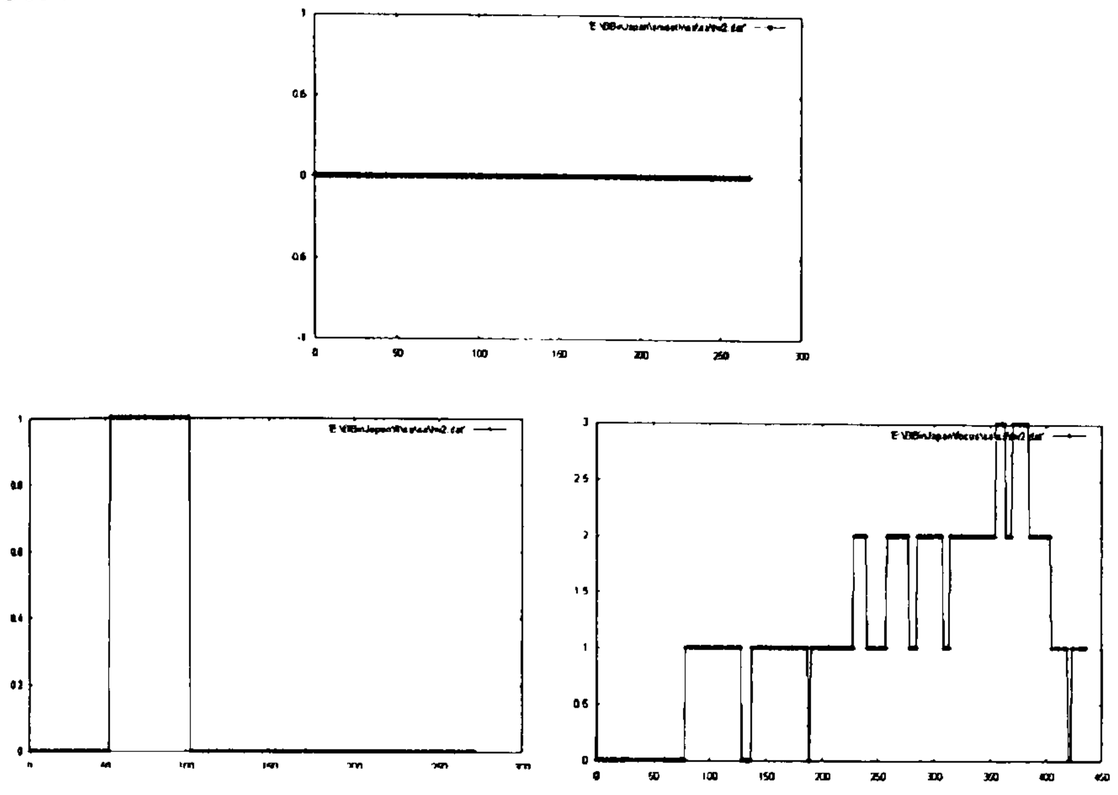


Figure 9: Examples of curvature density, estimated in a time windows of 100 frames. TOP: uninterested behavior; BOTTOM LEFT: animated behavior, BOTTOM RIGHT: interested/focused behavior

The graphs clearly indicate that whenever a person is interested in the scene objects, then they stop and spend time looking and while they do so they move about, building up a density of curvature maxima. Completely uninterested behavior shows no density at all for maxima above a threshold estimated by measuring the mean curvature of patches of uninterested people. Finally, animated behavior builds some density which, however, is not comparable with the density built for a focused behavior.

4. Conclusions

The paper has presented dynamics descriptors that make use of a skin color tracker and the trends of the tracked trajectories to infer a simple description of behavior in the scene. Preliminary experiments illustrate that curvature can be indeed employed to analyze trajectories and classify behavior. The amount of skin color patches in the scene and their life spans can shed some light on the clutter in the scene and their dynamics can be employed to assess a highly changing situation. The next step will be to further test our proposed method, provide a more automatic way to categorize scenes and the inclusion of robotic platforms, whose introduction in the scene is selected by the classification of dynamics. The introduction of robots and their interaction with the people present in the scene will then modify the dynamics and part of our future work will be to measure the dynamics "gradient" the human-robot interaction has caused. Briefly, one can envisage a robot being introduced to make an announcement, illustrate an exhibit or guide people in the scene. It is expected that a robot will reduce the entropy of the scene, and increase the interested of people.

5. Acknowledgements

Special thanks go to the Royal Academy of Engineering for funding Dr Remagnino's sabbatical in Japan, and to Professor Yoshinori Kuno for hosting both Remagnino's and Zhan's sabbatical at Saitama University. Dr Remagnino is also grateful to Dr Takashi Yoshimi, Humancentric Laboratory Corporate Research & Development Center Toshiba Corporation, and the Toshiba Science Museum for providing video data.

References

[1] K. Fukunaga and L.D. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," IEEE Transactions on Information Theory, vol. 21, pp. 32-40, 1975.

[2] Y. Cheng, Mean Shift, Mode Seeking, and Clustering, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 17, Issue 8, PP. 790-799, 1995.

[3] D. Comaniciu, V. Ramesh, P. Meer: Real-Time Tracking of Non-Rigid Objects using Mean Shift, IEEE Conf. Computer Vision and Pattern Recognition, Vol. 2, 142-149, 2000.

[4] G.R.Bradski, Computer Vision Face Tracking For Use in a Perceptual User Interface, Intel Technology Journal, Q2, Num. 15, 1998.

[5] S.Gong, S.J.McKenna, A.Psarrou, Dynamic Vision: From Images to Face Recognition, Imperial College Press, 2000.

[6] X.Wang, K.Tieu and E.Grimson, Learning Semantic Scene Models by trajectory analysis, European Conference in Computer Vision, 3, pp.110-123, 2006.

[7] Z.Zhang, K.Huang and T.Tan, Comparison of Similarity Measures for Trajectory Clustering in Outdoor Surveillance, International Conference on Pattern Recognition, Vol. 3, pp. 1135-1138, 2006.

[8] Arthur Dempster, Nan Laird, and Donald Rubin. "Maximum likelihood from incomplete data via the EM algorithm". *Journal of the Royal Statistical Society*, Series B, 39(1):1–38, 1977.

[9] P.Remagnino and G.L.Foresti, Ambient Intelligence: a New Multidisciplinary Approach, IEEE Transactions on Systems, Man and Cybernetics, Volume 35, Issue 1, pp. 1-6, 2005.

MATCHING GRADIENT DESCRIPTORS WITH TOPOLOGICAL CONSTRAINTS TO CHARACTERISE THE CROWD DYNAMICS

B.Zhan[1], P.Remagnino[1], S.Velastin[1], F.Bremond[2], M.Thonnat[2]

1 {B.Zhan, P.Remagnino,Sergio.Velastion}@kingston.ac.uk, Digital Imaging Research Centre, Kingston University, UK
2 {Francois.Bremond, Monique.Thonnat} @inria.fr, ORION, Sophia Antipolis, INRIA, France

Keywords: Local descriptors, sparse matching, visual surveillance, complex dynamics analysis.

Abstract

Understanding complex crowd scenes involves many dimensions: level of clutter, density of pedestrians, global/detail dynamics of the scene, etc. Socio-dynamics has tackled the problem by providing prototypes of crowd behaviour based on human observations and then explaining them by certain physical models. We are interested in automatic learning the factors involved in complex scenes for different dimensions and deriving a physical model from the real world via computer vision methods. In this paper we propose a solution to extract the motion of complex crowd which could be very difficult for conventional trackers. The method is based on a matching of local gradient descriptor supported by local topology constraints. Spatial pyramid and temporal smoothing are employed for optimizing the algorithm. Dynamics can then be accumulated over time so as to derive a higher level understanding of the scene.

1 Introduction

Visual scene dynamics can be analyzed at different resolutions e.g tracking of moving object, describing behaviour therein. The rationale of this paper is based on the assumption that people dynamics are purposive, and that the presence of many people has a clear conscious or unconscious influence over people's motion.

Our aim is to be able to identify the main motion patterns in a very complex scene automatically, making use of computer vision methods and based on automatic machine observation, rather than human observation and inference. We can infer these important characteristics by modelling the flow of people and objects in the scene.

Socio-Dynamics researchers have made extensive use of physical models for mathematical descriptions of interactions between people. Perhaps the most interesting concept is captured by the so-called "Master Equation" [6] (Equation (1)):

$$\dot{P}(x,t) = \sum_{x' \neq x} w_t(x \mid x')P(x',t) - w_t(x' \mid x)P(x,t) \qquad (1)$$

In the case of people flow observed through a single camera, $\dot{P}(x,t)$ represents the temporal variation of the stochastic process associated with given local characteristics x of image I_t. The right hand side of Eq. (1) indicates that the flow differential can be estimated by integrating the all possible transitions related to state x. If we limit the integration to being over a defined region of the image, we have a Markov process that can be easily estimated by accumulating local estimates of dynamics' variations. This can be implemented through reinforcement learning [11], using an iterative scheme.

Figure 1. Two examples of complex scenes

Currently our focus is on the problem of extracting local characteristics like scene motion. Background modelling [13] has been used to remove the static background. However, it gradually loses effect when the foreground/motion density increases, or when people gather in specific places (localized high density) most of the time. Also occlusion can become serious due to the complex structures in the scene. Thus it is crucial to have a descriptor that is robust and reliable to describe atomic features for retrieving the dynamics under such circumstances. To address these problems we propose a new method to derive local estimates of motion patterns by locally checking spatial temporal consistency of colour gradient in terms of interest points. This is further improved by adding local topology constraints. Extensive literature exists on spatial and spatial-temporal descriptors that have been used in image retrieval applications [10] as well as for analyzing image sequences in sport and monitoring applications [8][3]. In this paper we validate the Harris colour detector [5] and make use of local topology to make the

matching between frames more robust for cases of extreme clutter. To optimize computational load and to allow for a wider search a pyramidal search was also employed [9][2]. The paper is organised as follows: section 2 describes the matching method we propose, section 3 shows the experimental results, and section 4 explains how we could reconstruct the main path of the scene by the extracted displacement.

2 Proposed Method

In order to devise algorithms that automatically learn crowd/complex dynamics, local descriptors, or interest points, have been extracted using colour gradient information at the scale space. Besides the use of such descriptors, an advanced matching scheme is developed that provides improvement through the incorporation of topological constraints. Spatial pyramid is also used to enable fast matching for a large number of interest points across a sparse area, and to ensure time consistency of the matching results, a temporal smoothing procedure is adopted.

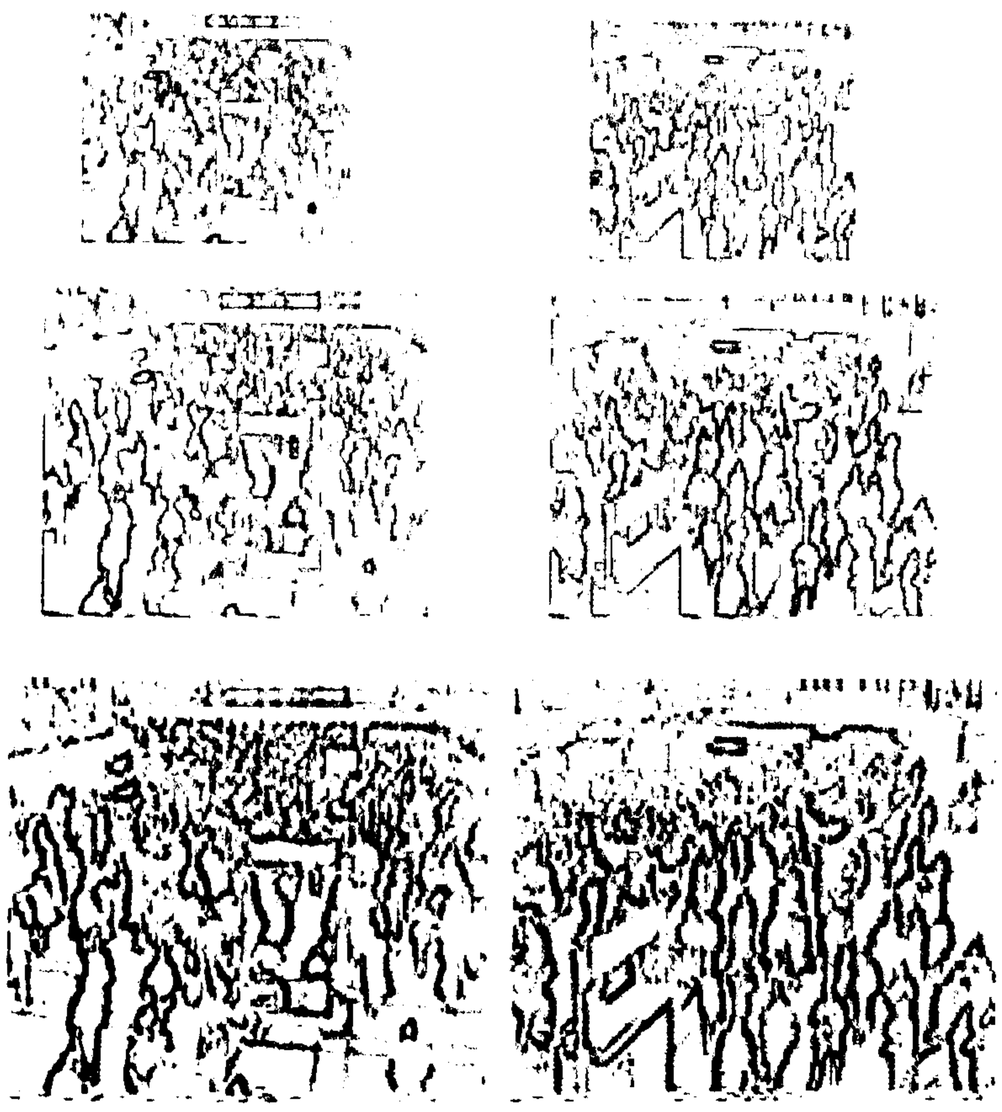


Figure 2. The related pyramids representations of the image frames in Figure 1, the bottom layers show the interest point with red dots.

2.1 Extraction of local descriptors

The proposed method makes use of the Harris interest point detector modified as in [4]. The Harris interest point detector is selected because it provides a repeatable and distinctive descriptor of the image features, which also has the advantages of being excellent repeatable under various conditions like view-point and illumination changes as a colour operator. The modified M matrix in such detector extracts feature points making use of the three chromatic channels $C=(R,G,B)$ (Equation (2)):

$$M = G(\sigma) \otimes \begin{pmatrix} C_x \cdot C_x & C_x \cdot C_y \\ C_y \cdot C_x & C_y \cdot C_y \end{pmatrix} \qquad (2)$$

In the operation the image is firstly smoothed using a standard Gaussian operator G of deviation σ . C_x and C_y are respectively the gradient in the x and y directions of the pixel chromatic triplet. They are estimated by applying the Gaussian derivative operator G , of deviation σ_0 , to the smoothed image, this is efficiently implemented by using the method from [15]. The interest points are then extracted using the term R , calculated as a combination of the Eigen values of the M matrix:

$$R = \det(M) + \kappa trace^2(M) \qquad (3)$$

Where κ is a constant in the range $0.04 <= \kappa <= 0.06$. The points with local maximum R are selected as interest points. This procedure takes place at the lowest (finest scale) layer and then the interest points are projected up to the top (coarsest scale) layer of the pyramid (Algorithm 1).

Algorithm 1: Creation of Interest Points

for N images in Δt (for temporal smoothing in 2.2) **do**
  generate pyramid image gradient
  detect interest points at bottom layer
  project interest point to top layer
end for

2.2 Advanced Scale Space Matching by topological constraints

Matching is carried out in two steps: for interest point in reference image searching of the candidate matching points in matching image by similarity and then applying topological constraints by comparing the neighbour interest points' arrangement. For each selected interest point $ip_i(t)$ in the reference image I_t corresponding candidate matching point $ip_j(t+\Delta t)$ is searched in the matching image $I_{t+\Delta t}$ inside a given search window $W \times W_{search}$ (indicated as a blue rectangular window in Figure 3) using the gradient similarity measure given by the formula:

$$sim(a,b) = \frac{\min(R_a, R_b)}{\max(R_a, R_b)} \qquad (4)$$

as introduced in [7], consisting of a $\Re^2 \rightarrow [0,1]$ mapping (The term R is the same as defined in equation (3)).

The matching thus far is not robust, due to the instability of interest points in a highly complex scene. Frequent occlusions

reduce the probability of identifying correct matches and, without local support, similar gradient localities might be found as feasible matches generating false positives.

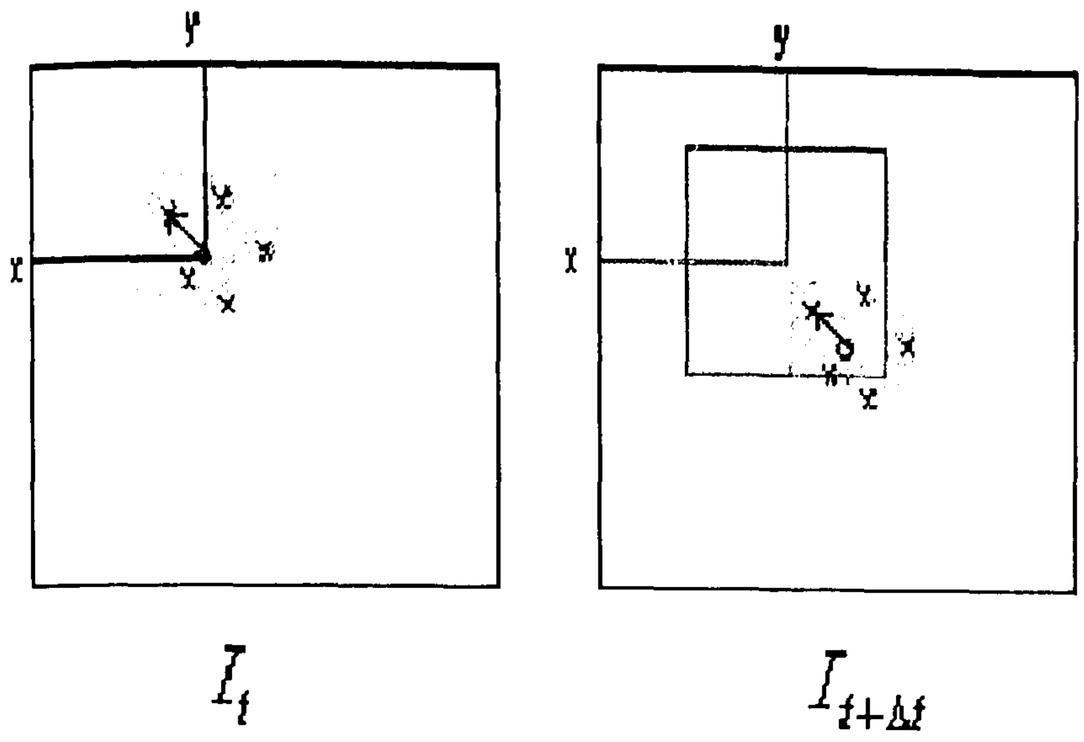


Figure 3. Topological Matching Window (the blue rectangle is the search window in the second frame to search for corresponding interest points. The circle point is the corresponding interest point $ip_i(t)$, $ip_j(t+\Delta t)$, the crosses are the local support interest points $ip_i^{k_t}$ and $ip_j^{k_{t+\Delta t}}$ (the local support window $Win \times Win_t$ and $Win \times Win_{t+\Delta}$ are illustrated as solid grey window).

This paper proposes a topological constraint to make the search for correspondences more robust. Gabriel *et al.* [3] proposed a similar method using topological information, however in their algorithm the area (object) of interest was predefined and the topological information was evaluated by an already known centre of the object. In our approach, we are not aiming to track in detail a particular object at this stage. Instead, we are interested in the local motions Therefore, the necessary local support is derived from local windows centred at the interest point and we make use of the relative location of the interest points in such windows. Support is estimated for the matched interest point pair $(ip_i(t), ip_j(t+\Delta t))$ inside the support window $Win \times Win_t$ and $Win \times Win_{t+\Delta}$ centred at $ip_i(t)$ and $ip_j(t+\Delta t)$, respectively in I_t and $I_{t+\Delta t}$ (grey windows in Figure 3). All interest points found in $Win \times Win_t$ (let us call them generic $ip_i^{k_t}$) are then matched. Support is then quantified in terms of the error by measuring the standard deviation of the ensemble of found correspondences

$$\varepsilon_{ij} = f(\sigma_{\theta_{ij}}, \sigma_{\rho_{ij}}) \qquad (5)$$

where

$$\theta_{k_t k_{t+\Delta t}}(\cdot) = \mathcal{N}(v_{ik_t} \cdot v_{jk_{t+\Delta t}}) \qquad (6)$$

$$\rho_{k_t k_{t+\Delta t}}(\cdot) = \| v_{ik_t} \| - \| v_{jk_{t+\Delta t}} \| \qquad (7)$$

are, respectively, the orientation difference (dot product) and length difference between the vectors (indicated as arrows in Figure 3):

Algorithm 2: Topology Matching

Matching:
for all $ip_i(t) \in I_t$ **do**
  search $ip_j(t+\Delta t) \in I_{t+\Delta t}$ in $W \times W_{search}$
  for all $(ip_i(t), ip_j(t+\Delta t))$ **do**
    define a topology window $Win \times Win_{sup\ p}$
    centre $Win \times Win_t$ at $ip_i(t)$ and $Win \times Win_{t+\Delta}$ at $ip_j(t+\Delta t)$
    for all $ip_i^{k_0} \in Win \times Win_t$ and $ip_i^{k_1} \in Win \times Win_{t+\Delta t}$ **do**
      find best matches $(ip_i^{k_0}, ip_j^{k_1})$
      let $v_{ik_t} = ip_i^{k_t} - ip_i(t)$
      let $v_{j_{k_{t+\Delta t}}} = ip_j^{k_{t+\Delta t}} - ip_j(t+\Delta t)$
      define $\theta_{k_t k_{t+\Delta t}}(\cdot) = \mathcal{N}(v_{ik_t} \cdot v_{jk_{t+\Delta t}})$
      $\rho_{k_t k_{t+\Delta t}}(\cdot) = \| v_{ik_t} \| - \| v_{jk_{t+\Delta t}} \|$
  end for
  estimate derivation
  $\sigma_{\theta_{ij}} \longleftarrow \{\theta_{k_0}, \theta_{k_1}\}$
  $\sigma_{\rho_{ij}} \longleftarrow \{\rho_{k_0}, \rho_{k_1}\}$
  end for
  define $\varepsilon_{ij} = (\alpha\sigma_\theta + \beta\sigma_\rho) / sim_{ij}$
  choose the smallest ε_{ij}
end for

$v_{ik_t} = ip_i^{k_t} - ip_i(t)$ and $v_{j_{k_{t+\Delta t}}} = ip_j^{k_{t+\Delta t}} - ip_j(t+\Delta t)$.

These vectors represent the relative position of the generic point and the centre point. Deviation for both θ and ρ are estimated for all interest points matches found in local support windows. In the proposed algorithm, deviations are also weighted by the similarity between interest points. The pair $(ip_i(t), ip_j(t+\Delta t))$ that attains the smallest deviation is then chosen as the candidate matching pair. However, we discard the pair where deviation is higher than a pre-defined maximum, to cater for completely disagreeing displacement vectors, e.g., generated by two people moving away from one another. The estimated motion vector is then propagated to the bottom layer of the pyramid.

2.3 Temporal Smoothing

We also carry out temporal smoothing and matching by comparing a number of N spatial pyramids (estimated with Algorithm 1), corresponding to a specific Δt time window. Thus a spatial-temporal pyramidal analysis of the sequence is generated for a number of frames. Temporal smoothing is

employed to enforce time consistency on matches, reducing false alarms generated by unstable interest points. To avoid the large change of the motion vector, Δt is chosen proportional to the employed frame rate.

So matching is carried out in both space and time, starting at the highest level (coarsest level) of each pyramid, searching interest point correspondences between the initial frame I_0 of the N frames and each other frame I_k within the given time period Δt (corresponding to N-1 matches). Spatial matching works from the top (finest scale) of a pyramid to the bottom (coarsest level). Then temporal integration of pyramidal matches of interest point j in 0^{th} frame can then be applied by combining the N matches as

$$\hat{ip} = \alpha \sum_{k=1}^{N} \cdot sim_{0k}(ip_{j0}, ip_{jk}) \cdot mip_j^k \qquad (8)$$

where α is a normalization constant, mip_j^k is the match vector with the kth frame, $sim_{0k}(ip_{j0}, ip_{jk})$ is the similarity of the matched interest points again using the definition in (4).

Algorithm 3 Matching over Δt

for all images in the sequence **do**
  for N image pyramid **do**
    for all interest points $ip_{j0} \in I_0$ and $ip_{jk} \in I_k$ do
      $mip_j^k = \arg\max(sim(ip_{j0}, ip_{jk}))$
    end for
    combine matches
    $\hat{ip} \propto sim_{0k}(ip_{j0}, ip_{jk}) \cdot mip_j^k$
  end for
end for

3 Results

A series of experiments were run on different video sequences. The assessment of results is not a trivial task, given that it is virtually impossible to generate ground truth data. However, results can be judged qualitatively, by overlapping the flow of dynamics to the image frames (Figure 4). A quantitative evaluation of results can also be provided. For a window of interest Win_k^t selected in an image I_t of a given sequence S_i, all interest points are retrieved and then their displacements were estimated against the image at the next frame $I_{t+\Delta t}$. All displacements are then combined to a resulting vector that indicates the position of the window of interest in the next frame. Comparing structure cannot work, because background structure would generate noise that could even be larger than the information we wish to compare. Therefore we decided to generate *Receiver Operating Characteristic* (ROC) curves [12]. A series of points (P_{tp}, P_{fp}) is estimated to produce the ROC curve using the following two formulas:

$$P_{fp} = \frac{FP}{TN + FP} \qquad P_{tp} = \frac{TP}{TP + FN} \qquad (9)$$

where the definitions of the parameters are shown in Table 1. Hence P_{tp} represents the fraction of positives correctly predicted and P_{fp} represents the fraction of negatives incorrectly predicted.

ROC		predicted	
		Positive	Negative
signal	True	TP	TN
	False	FP	FN

Table 1 The definitions of parameters for ROC curve

(a)

(b)

Figure 4 Motion vectors overlapped on the image frames ((a) shows the global motion flows a-1 and a-2 are two frames spans Δt and (b) shows a focus on a group of four pedestrians in (a).The group is illustrated with a circle in (a) to indicate their global position change in the two frames, while (b) gave the motion from a-1 to a-2. The red parts of the motion vectors are the start, while the green ones are the end).

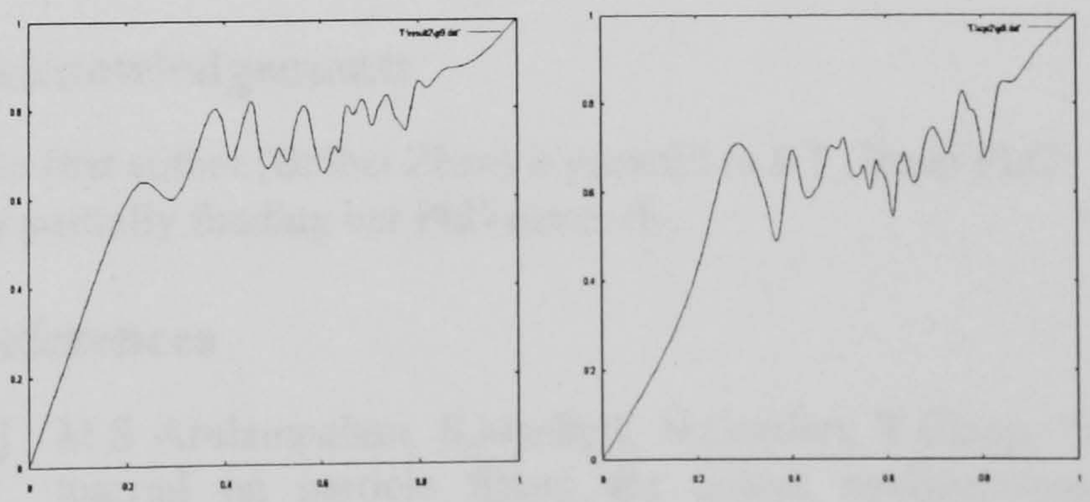


Figure 5 ROC curve of two image sequences (with the vertical axis as TP, the horizontal axis as FP)

In our case the work is done by comparing the predicted positions of all interest points in Win_k^t against the actual

interest points found in $Win_k^{t+\Delta t}$. If an interest point has been found in the location where we predicted, it counts as a true positive (TP), else it counts as a false positive (FP). An image sequence can be scanned every frame or more sparsely, m windows within the same frame placed at k random position (in the random process we could probably have a window which contains two objects that have opposite moving direction, so to avoid the error caused by this, we discard the window if we detect that the direction of the motion vectors are not uniform inside the window), and if this is carried out for n frames we would end up with $m \times k \times n$ ROC measurements (Figure 5).

4 From displacement to paths

Ongoing work concerns the reconstruction of the main paths of a scene (modal paths or trajectories) based on a grouping of the estimated local displacements accrued over time.

During the sequence analysis two probability density functions can be built modelling the crowd dynamics of the scene from both global and local dimensions: a global probability density function incrementally modelling the motion presence across the entire view of the scene, and local density functions built over time that model the displacement for a local unit (e.g. a pixel or cell), depending on how the view is tessellated.

The estimated ensemble of density functions can be used as a Reinforcement Learning (RL) solution to cast the solution to interpret the dynamics. Formally, the basic reinforcement-learning model consists of a set of environment states S and a set of actions A. At each time t, the agent perceives its state $s \in S$ and the set of possible actions $A(s)$. It chooses an action $a \in A(S)$ and receives from the environment the new state [14].

Rather than building the solution following the stochastic dynamic recurring formula provided by the RL method, we use the solution as an RL solution to estimate the modal scene paths.

An analogy can be drawn between density functions and RL model, and in theory the ensemble of density functions could also be incrementally built following the RL updating equation. At the building stage, each displacement will count toward the *quality* of a *(s, a)* (state and action pair), and in the end the set of density functions can be used to cast the solution as an RL solution to the given stochastic problem.

Problems solved following the RL method, allow the retracing of the optimal path by starting, either at a random state or at a predefined starting state and then following sequentially all the actions (a) at higher probability among the available and feasible actions (a) to the new state. In similar fashion, our solution can be used to trace the optimal trajectory among all possible trajectories, by starting from a peripheral state (border of the image, or entrance of the scene within the image), and then following the path at higher probability. One constraint must be imposed to the tracing, enforcing low probability to cyclic paths in the scene (in theory the tracing could bounce back and forth between a small number of states). This can be easily implemented by assuming that the current direction is the most likely one (another analogy to the RL term policy will not vary much in a suitably defined neighbourhood (a small window), making straight paths more likely than turning paths).

The above description explains how paths can be extracted using the ensemble of density functions. This is not the end of the story. In fact, the classification of modal trajectories based on likelihood of occurrence is of significant interest. This can be implemented by assuming a finite starting number of paths and a parallel tracing of such paths in the manifold defined by the ensemble of the estimate density functions (for instance following a dynamic programming scheme). Tracing can then be implemented with distributed modules keeping track of the path likelihood, and at the same time, classes can be defined by considering how far apart fall traced paths, for instance using a distance measure function of the Euclidean distance between paths and their lengths.

5 Conclusions and future work

In this paper we have introduced a novel method capable of automatically extracting the dynamics of crowd movements. The qualitative results obtained show that the estimated motion vectors can visually describe pedestrian motion. The resulting ROC curves indicate that the probability of true positive (P_{tp}) raises high when the P_{fp} is still low showing the robustness of the algorithm. However, improvements can still be made if more knowledge about local support can be extracted, and tracking methods like particle filter could be employed to get better performance.
We feel this is a suitable precursor to processes for learning modes of complex dynamics, describing behaviour and supporting for work in high-level vision and socio-dynamics modelling. As a result, it is expected that the method presented here to extract dynamics is likely to find use for building up crowd dynamics models.

Acknowledgements

The first author (Beibei Zhan) is grateful to BT Group PLC for partially funding her PhD research .

References

[1] M.S Arulampalam, S.Maskell, N.Gordon, T.Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking". IEEE Transactions on Signal Processing, pp 174-188, (2002).

[2] P.J.Burt and E.H.Adelson. "The laplacian pyramid as a compact image code". IEEE Transactions Communications, 9(4):532540 (1983).

[3] P.Gabriel, J.-B. Hayet, J.Piater, and J.Verly. "Object tracking using color interest points", In *IEEE International Conference on Advanced Video and Signal based Surveillance*, pp 159-162.
[4] V.Gouet and N.Boujemaa. "About optimal use of color points of interest for content-based image retrieval". Technical Report RP-4439, INRIA,(2002).
[5] C.Harris and M.Stephens. "A combined corner and edge detector", Proc. 4th Alvey Vision Conf. pp 189-192,(1988).
[6] D.Helbing. "Quantitative SocioDynamics: Stochastic Methods and Models of Social Interaction Processes". Kluwer Academic Publisher, (1995).
[7] I.-K.Jung and S.Lacroix. "A robust interest point matching algorithm". In IEEE International Conference on Computer Vision, volume 2, pages 538-543,(2001).
[8] I.Laptev. "On space-time interest points". *International Journal of Computer Vision*, 64(2-3):107-123,(2005).
[9] T.Lindeberg. "Scale-space theory: a framework for handling image structures at multiple scales". In Pror.CERN School of Computing, pp 1-12,(1996).
[10] K.Milkolajczyk and C.Schmid. "A performance evaluation of local descriptors". IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(10):1615-1630,(2005).
[11] S.Sutton and A.G.Barto. "Reinforcement Learning: an Introduction". MIT Press,(1998).
[12] J.A.Swets, "Measuring the accuracy of diagnostic systems". Science, 240(4857): 1285-1293,(1988).
[13] Stauffer, C. Grimson, W. E. L,"Adaptive Background Mixture Models for Real-Time Tracking", IEEE Computer Society Conference On Computer Vision And Pattern Recognition, VOL 2, pages 246-252, (1999).
[14] Wikipedia," Reinforcement learning", http://en.wikipedia.org/wiki/Reinforcement_learning
[15] I.Young and L. van Vliet. "Recursive implementation of the Gaussian filters". Signal Processing, 44(2): 139-151,(1995).

Motion Estimation with Edge Continuity Constraint for Crowd Scene Analysis

B.Zhan[1], P.Remagnino[1], S.Velastin[1], N. Monekosso[1], L.Xu[2]

1 DIRC, Kingston University, UK; 2 BT Group PLC, UK

Abstract. We present in this paper a new motion estimation method aiming at crowd scene analysis in complex video sequences. The proposed technique makes use of image descriptors extracted from points lying at the maximum curvature on the Canny edge map of an analyzed image. Matching between two consecutive frames is then carried out by searching for descriptors that satisfy both a well-defined similarity metric and a structural constraint imposed by the edge map. A preliminary assessment using real-world video sequences gives both qualitative and quantitative results.

1 Introduction

Understanding crowd behavior is a relatively new research topic in computer vision and can be applied to a variety of domain problems, including space optimization, ambient intelligence and visual surveillance. In this paper we describe a new technique that combines image descriptors and edge information to estimate the crowd motion of video sequences to better support behavior analysis.

Pedestrians' behavior differs when they walk as individuals and when they are part of a crowd. Crowds have been studied by sociologists and civil engineers and physics models were proposed to describe quantitatively complex behavior dynamics [1][2]. Le Bon compares a crowd to a chemical compound as it displays properties quite different from those of the bodies that have served to form it [3]. Methods normally employed to describe the behavior of an individual would fail in crowd situations, especially when the level of clutters is very high.

Computer vision methods have been used to focus on extracting information from video sequences of crowded scenes, for instance, to estimate crowd density, e.g., [4], [5]. In order to model behavior, the tracking of individuals and groups of people must be implemented. Two categories of tracking can be broadly defined: one for detecting and tracking single persons in a crowd, which requires a window of interest area to start with like the work of [6], employing Harris interest points detector and point distribution model; the second one is for detecting and tracking a few (3 to10) people using sophisticated tracking techniques like the particle filter [7], Markov Chains [8] and probabilistic frameworks [9]. Furthermore, recent work has focused on the interpretation of highly crowded scenes [10] employing statistical methods to extract the

main paths of a crowded scene or using hidden Markov models to describe the *normal* behavior of a crowded scene [11].

Our main goal is to employ computer vision methods to develop robust motion estimation methods and model statistically both the instantaneous and recurrent dynamics of a highly crowded scene.

In this paper we propose an algorithm to estimate crowd motion in scenes of different level of density and clutter. Inspired by deformable object tracking techniques [12], we make use of the edge information and its curvature to extract descriptor points (those with local maximum curvature) as salient features of an edge. Instead of using points, edge information is maintained by "edgelet constraint" to refine the estimation. Thus we combine the advantage of using point features which are flexible to track and the advantage of using edge features which maintain structural information.

The paper is organized as follows. Section 2 describes our proposed method in detail; Section 3 presents selected results of applying the method to different dense visual scenes, and in Section 4 we give concluding remarks and discuss future work.

2 Proposed Method

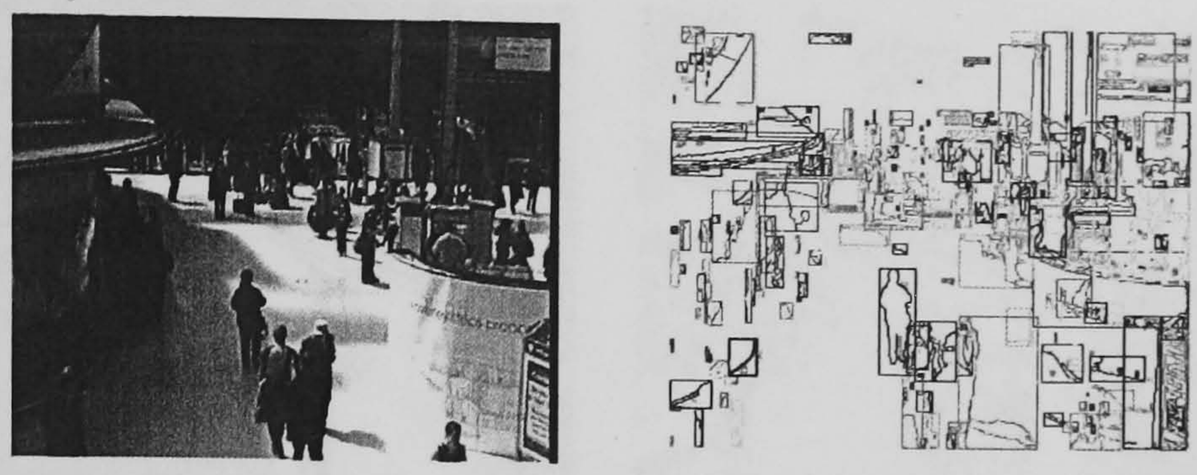
The motion estimation between frames is carried out in four steps:

- A conventional Canny edge detector is run over each image frame and the edge chains are retrieved;
- the curvature is calculated for every point on an edge chain, and along each chain the interest points at maximum curvature are chosen;
- then, for all the extracted points in the first frame of each frame pair, we search for matching candidate points in the second frame;
- and finally the *edgelet* constraint is applied to obtain the best point matches.

Sections 2.1 to 2.4 describe the above steps in detail; Section 2.5 explains the role of the background model so as to improve the performance of the proposed method.

2.1 Edge Retrieval

The Canny edge detector is employed to extract the edge information of a given frame. Each Canny edge is a chain of point S_p, and all the edges are stored in an edge list L_p. Fig. 1(a) and (b) show, respectively, an example image frame and the extracted edge chains with associated bounding boxes. It can be observed that even in a scene of medium density crowd, edge chains can occlude each other, increasing the descriptor matching complexity.



(a) (b)

Fig. 1. (a) Original frame, (b) Edge chains and their bounding boxes

2.2 Curvature Estimation and Corner Point Extraction

Interest points can be quickly extracted for a sequence frame, for instance, with the Harris corner operator [13]. Although interest points can represent the local characteristics of an image in isolation, they cannot represent a shape. We therefore propose the extraction of interest points from edges and then impose the constraint that they lie on a specific edge.

Each edge can be represented by a parameterized curve:

$$x = x(t) \text{ and } y = y(t) \tag{1}$$

We smooth the curve with a Gaussian filter, as follows

$$\begin{aligned} X(t) &= G(t) \otimes x(t) \\ X'(t) &= G'(t) \otimes x(t) \\ X''(t) &= G''(t) \otimes y(t) \end{aligned} \tag{2}$$

The curvature of each edgelet can then be given by [14]:

$$\kappa = \frac{(X'Y''-Y'X'')}{(X'^2+Y'^2)^{\frac{3}{2}}} \tag{3}$$

The Gaussian filter, the first- and second-order derivative filters can be easily implemented using the method described in [15].

Corner points are defined and extracted as the local maxima of the absolute value of curvature on each edge.

Thus we convert the edge representation from a point sequence S_p to a corner point sequence S_c, resulting in a list L_c of S_c for all the edges of the image.

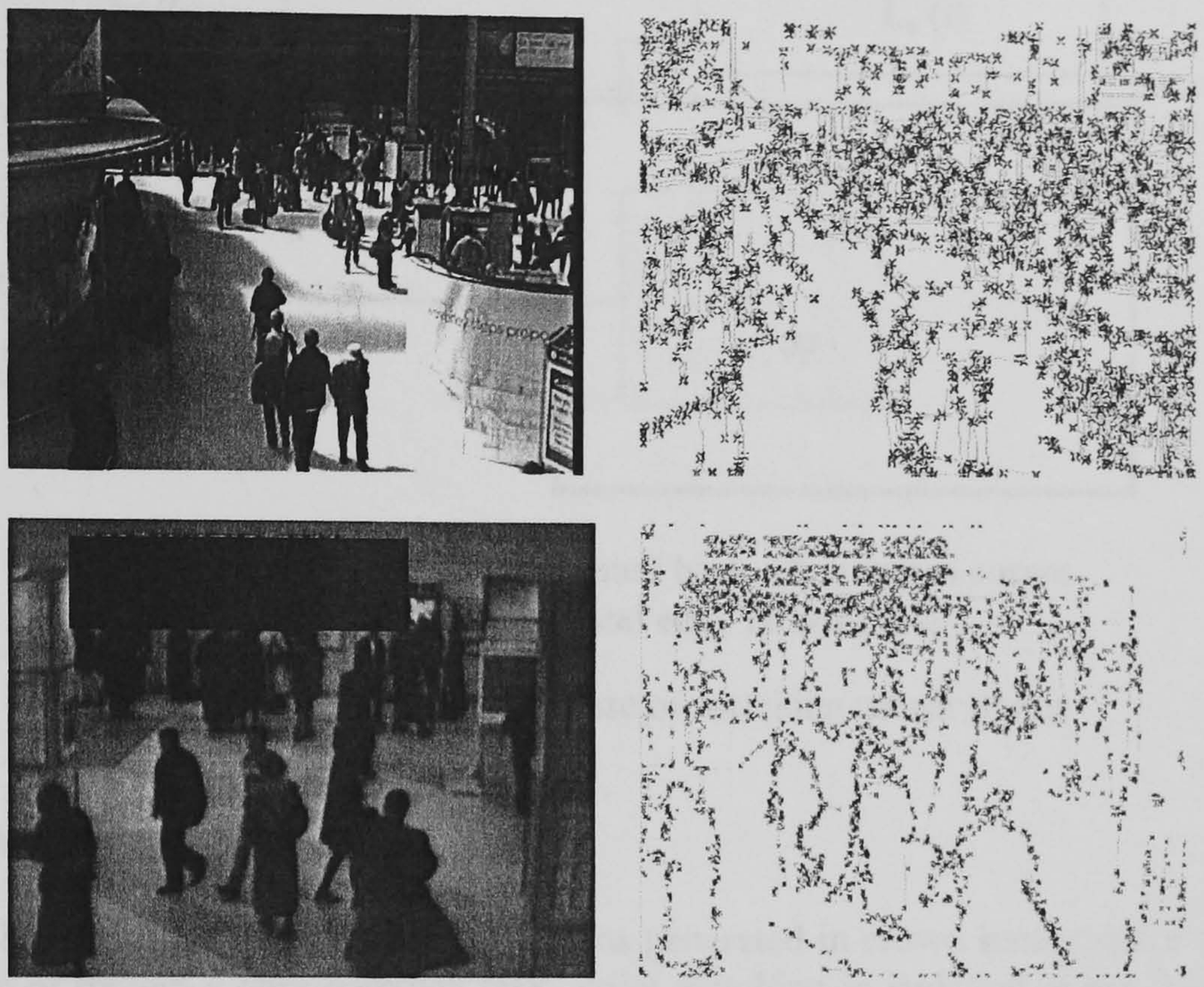


Fig. 2. Original frame (left) and extracted corner points (right), marked with red crosses on grey edges. Two scenes of different complexity levels are illustrated.

2.3 Point Matching by Curvature

Given two frames $I(t_0)$ and $I(t)$ and we estimate the motion of the points from $I(t_0)$ to $I(t)$. For each corner point $C_{t0}(j,i)$, with coordinate (x,y) from every $S_{t0}(t,j)$ of $L_c(t_0)$, we search in $I(t)$ the area inside a rectangular search window W centered at (x,y). A look-up table (LUT) contains corner point information of $L_c(t)$ is generated to enhance the matching. (Illustrated in Fig. 3).

The correspondence is matched by using curvature information of corner points in W in LUT against the reference point $C_{t0}(j,i)$. The error is calculated by the curvature :

$$error(j,i) = | \kappa(C_{t0}(j,i)) - \kappa(C_t(j',i')) | \tag{4}$$

The best n matches are then selected as candidate points and considered for the next step.

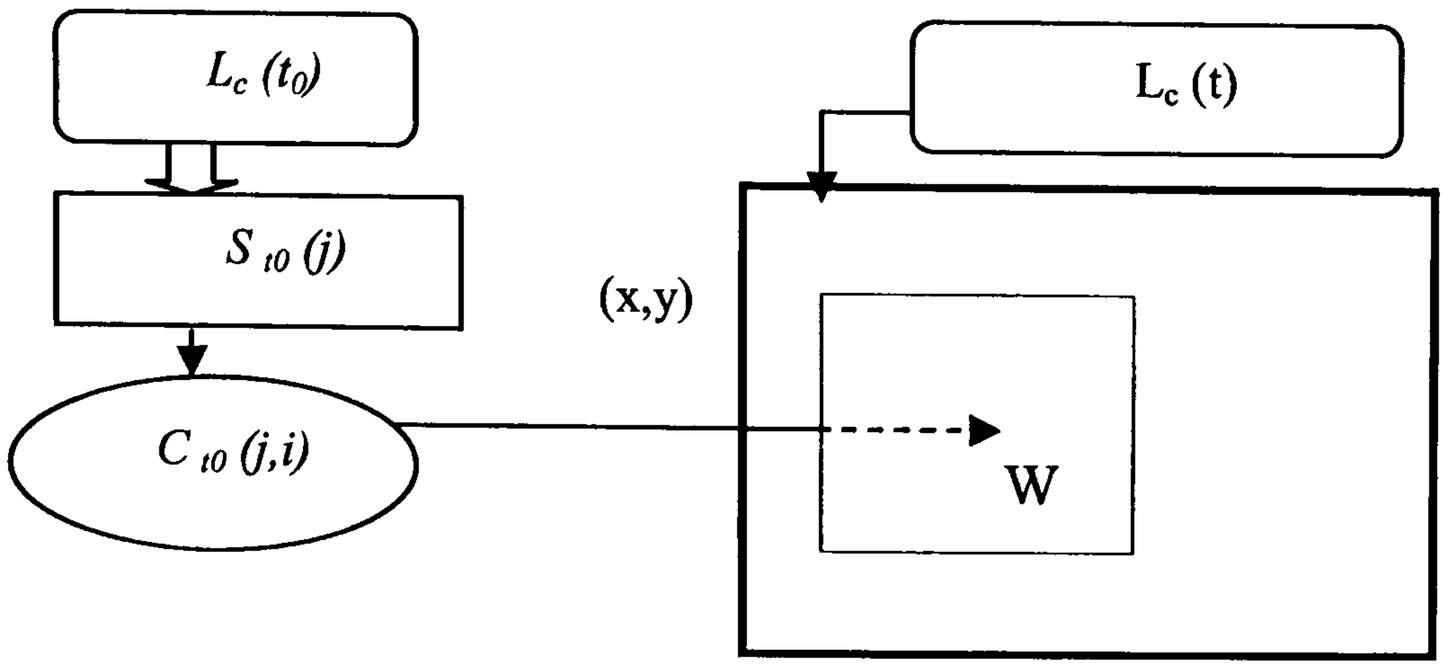


Fig. 3. The procedure of matching corner points

2.4 Applying the Edgelet Constraint

Complex dynamics and frequent occlusions generated in crowd scenes make the estimation of motion a very complex task. Point matching in isolation is too fragile and prone to errors to provide a good motion estimator. If interest points are extracted on edge chains, then the edge constraint can be imposed and used.

For image frame $I(t_0)$, we divide every $S_{t0}(j)$to uniform length *edgelets* represented by sub sequences $E_{t0}(j,k)$. There are two reasons for doing this: to avoid a very long edge that could be generated by several different objects, and to enhance the matching of the edge fragments that generated by occlusions. For each corner point $C_{t0}(j,k,i)$ in $E_{t0}(j,k)$ we have as result from the first step, n candidate matching points. Each candidate point belongs to a sequence $S_t(j')$ in $L_c(t)$, thus we have m $(m<=n)$ candidate matching sequences (or pieces of sequence) for each edgelet.

To find the best match of $E_{t0}(j,k)$, we use three parameters: *energy cost, variation of displacements* and *the match length for each candidate* and combine them into a single matching score. Here we assumes that the length of $E_{t0}(j,k)$ is small enough that it would not split again to two or more matches, it is, that their candidate points should belong to the same candidate sequence.

Energy cost due to deformable object matching is calculated by accumulating the errors (again calculate by difference of the curvatures as in (4)) along matching point pairs of $E_{t0}(j,k)$ and all the candidate match points that belong to the same candidate sequence.

$$Energy = \sum error(i,j,k) = \sum (\kappa(C_{t0}(j,i,k)) - \kappa(C_t(j',i',k'))) \qquad (5)$$

Variation of displacements For each matching point pair we have a displacement pair dx_i and dy_i, combination of the variation of the two displacement vectors:

$$V = \frac{1}{L_M} \times \sqrt{\sum (\frac{1}{S}(dx_i - dx))^2 + (\frac{1}{S}(dy_i - dy))^2} \tag{6}$$

where dx and dy are the average displacements between the matched point pairs, S is the size of the match window, L_M is the number of total matched points of from $E_{t0}(j,k)$ to candidate sequence. Hence V lies in the range between 0 and 1.

And ***match length parameter***:

$$M = \frac{L_M}{L_E} \tag{7}$$

Where L_E is the total number of points on $E_{t0}(j,k)$, M between 0 to 1.

So the overall matching score is given by:

$$Score = Energy + V + (1 - M) \tag{8}$$

The candidate sequence of minimum matching score will be selected. However, if the match length parameter M < 50%, we will discard the result.

2.5 Improving Results with Background Model

Background modeling is commonly employed to segment foreground, though here we use it to reduce noise. When a scene is very crowded and people frequently occlude one another, it is not practical to segment foreground solely with background modeling. However, on the other hand there could be still some parts of the scene never covered by foreground objects (e.g. ceiling) which we can call permanent background. Here what we want is to use the background model to eliminate the noise generated by the permanent background. We adopted the Gaussian mixture model proposed by Stauffer [16] which builds an adaptive and updatable background model on a pixel by pixel basis to generate a foreground (FG) mask. In some cases a FG mask may lose some edge points, which could cause some of the FG edges to be broken by the background and lose their consistency. To avoid this we do not apply the FG mask directly on the image, but apply the mask when we extract the corner points, that is, eliminating the corner points that fall into background while keeping the connect information.

3 Experimental Results

We have tested our algorithm against a few of video data, which contain different densities of crowd, with different frame rates, and different pedestrian sizes (different camera set-ups and therefore perspectives). We chose to assess some results quantitatively, those including video sequences with a few people (3 to 4), for which the ground truth is known, and to asses others qualitatively, illustrating the flow of crowded scenes.

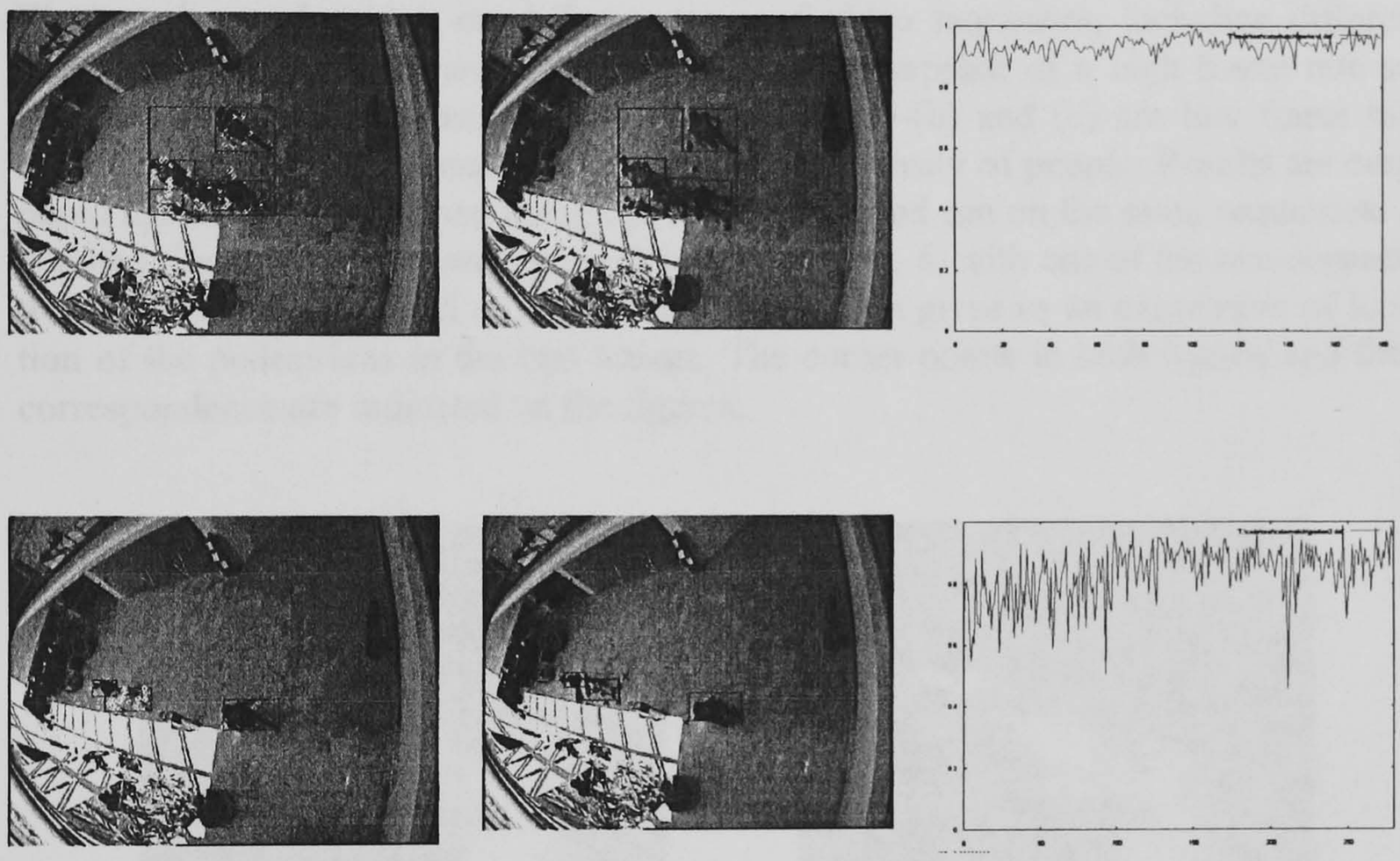


Fig. 4. Two test data set (a) and (b). The left two sample initial frames from both data set, with corner points indicated by white dots inside ground truth box; the middle two are the matched frame, with correct matched points CR_M marked by blue circle and incorrect matched points ICR_M marked by cross, the right two are the correct match rate R along the frames of the sequences.

3.1 Motion Estimation of Multiple People

The video data we use here are from the European project CAVIAR [17], ground truth information for these data is provided in XML format.

To test the result, for *I(t0)* we estimate the foreground object position (by means of a bounding box) and translate every corner point in the bounding box to the matching frame *I(t)* by its estimated motion, we then count all translated points still in the box as

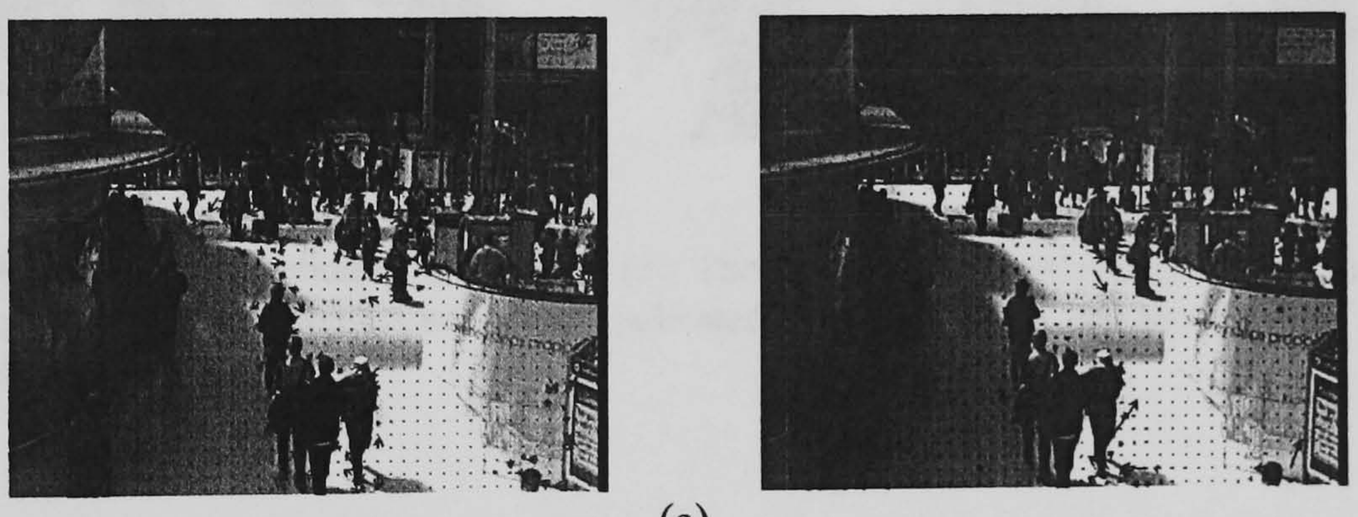
a correct match CR_M, and those falling outside of the box as an incorrect match ICR_M, and the correct match rate is calculated as:

$$R = \frac{CR_M}{CR_M + ICR_M} \tag{9}$$

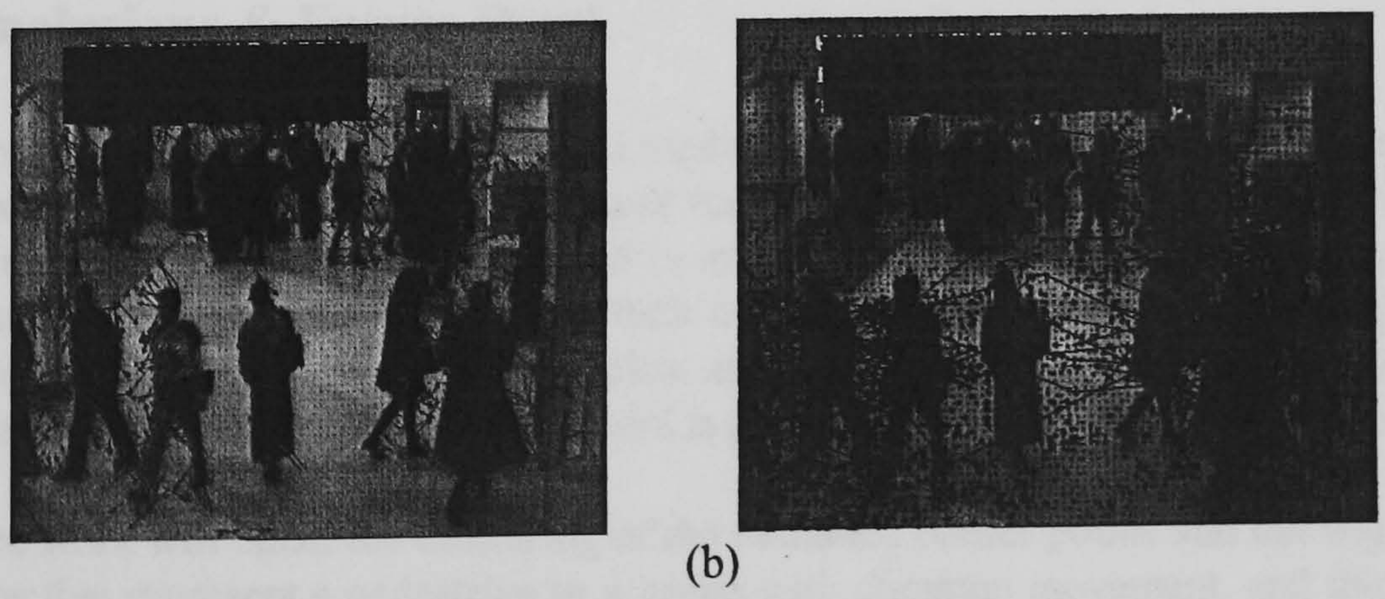
It is easy to see that in the optimal situation R should always be equal to 1.

3.2 Motion Estimation of More Complex Video Sequences

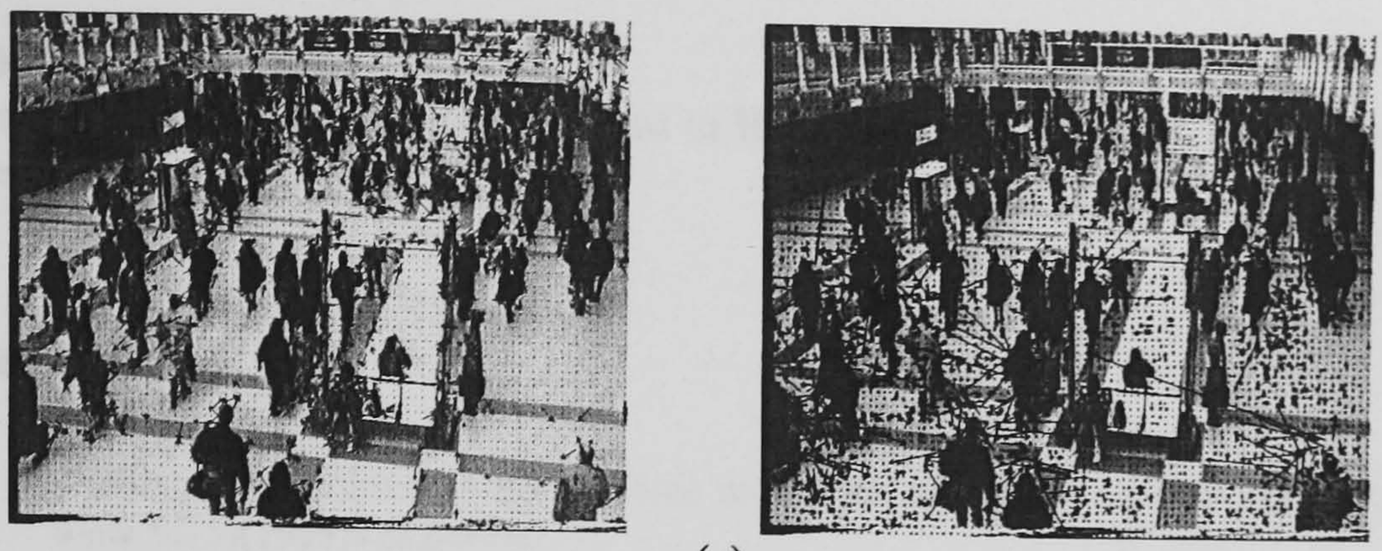
We tested our algorithm on different types of video sequences, including different number of people and frame rate. Fig. 5-(a) is a snapshot of a high frame rate sequence with medium density of people while Fig 5-(b) and (c) are low frame rate sequence (typically 1 frame/sec or less) with high density of people. Results are compared against those obtained using optical flow method run on the same sequences. A detailed illustration from sequence (c) is given in Fig. 6, with one of the two consecutive frames being overlaid on top of the other, which gives us an expression of location of the pedestrians in the two frames. The corner points in both frames and their correspondence are indicated on the figures.



(a)



(b)



(c)

Fig. 5. Visualized Results from different types of video sequences (left ones), against the results from optical flow method run on same sequences (right ones).

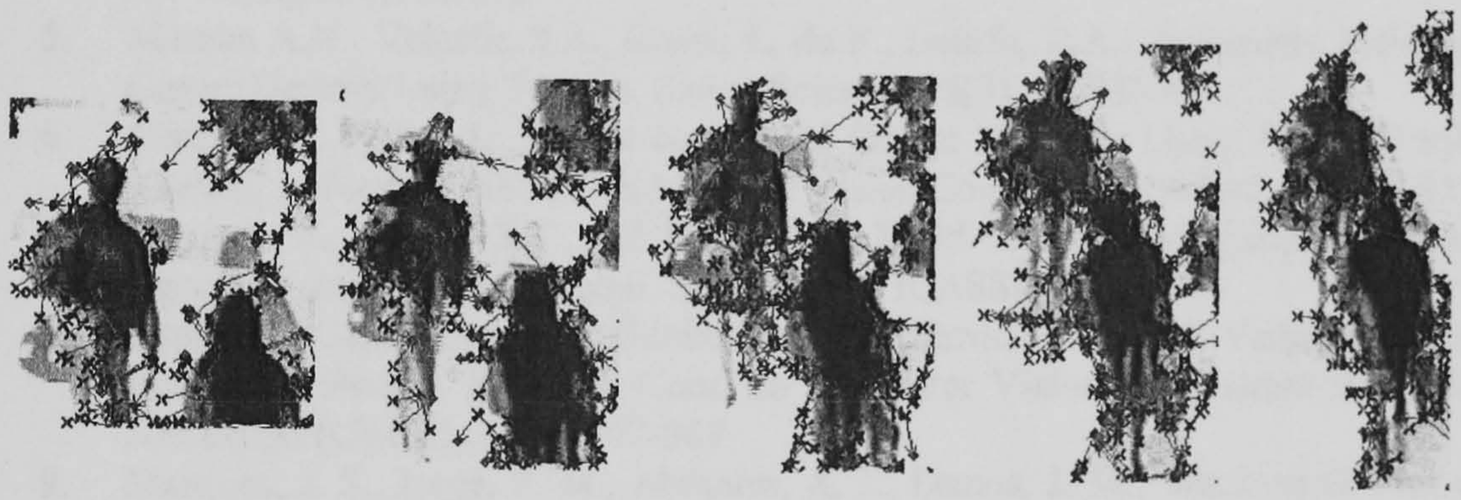


Fig. 6. A detailed illustration of a sequence (c): The figure is the overlapping of two consecutive images, with the initial corner points indicated by blue crosses and the matched points indicated by red circles.

4 Conclusions & Future Work

In this paper we have introduced a novel method to estimate the motion of a crowd in complex scenes and thus provided the basis for a high level description of a crowd for interpretation and modeling. The method in its essence relies on edge information for the extraction of image descriptors and their matching between frames. Corner points are selected as salient features and edgelets are employed to maintain the local edge information, an adaptive background model is associated with the system to reduce the noise.

Future work will entail the clustering of the extracted corner points and the edgelets to groups that represent a pedestrian or a group with common movement, and thus the method can be extended to the actual tracking of individuals or groups of people in very complex scenes.

Acknowledgements

The first author (Beibei Zhan) is grateful to BT Group PLC for partially funding her PhD research.

References

1. Helbing, D.; Molnar P.: Social force model for pedestrian dynamics. Physical Review,51(5)(1995)4282–4286
2. Pan, X., Han, C. S., Dauber, K. and Law, K. H: Human and Social Behavior in Computational Modeling and Analysis of Egress. Building Future Council Doctoral Program. Las Vegas, (2005)
3. Le Bon, G.: The Crowd. Cherokee Publishing Company (1895, reprinted 1982)
4. Ma, R., Li, L., Huang, W., Tian, Q: On pixel count based crowd density estimation for visual surveillance. Cybernetics and Intelligent Systems. 2004 IEEE Conference on, vol.1(2004)170-173
5. Marana A.N., Velastin S.A., Costa, L. da F., Lotufo, R.A.: Automatic Estimation of Crowd Density Using Texture, Safety Science, 28(3), (1988)165-175
6. Mathes, T., Piater, J.: Robust Non-Rigid Object Tracking Using Point Distribution Models. in Proc. of the British Machine Vision Conference, Oxford, UK, (2005)
7. Venegas, S., Knebel, S.F., and Thiran, J.P.: Multi-object tracking using particle filter algorithm on the top-view plan. Submitted to ICASSP04 (2003)
8. Smith, K., Gatica-Perez, D., Odobez J.: Using Particles to Track Varying Numbers of Interacting People. In: IEEE Conf. on Computer Vision and Pattern Recognition , San Diego (CVPR),(2005) 962-969
9. Marques, J. S., Jorge, P. M., Abrantes, A. J., Lemos, J. M.: Tracking Groups of Pedestrians in Video Sequences. cvprw, p. 101, 2003 Conference on Computer Vision and Pattern Recognition Workshop - Volume 9(2003)
10. Zhan, B., Remagnino, P., Velastin, S.A.: Mining paths of complex crowd scenes. Advances in Visual Computing: First International Symposium, ISVC 2005 (Eds. G Bebis, R Boyle, D Koracin, B Parvin), Lecture Notes in Computer Science (Vol. 3804/2005) Springer-Verlag GmbH, December, Nevada, USA, ISBN/ISSN 3-540-30750-8,(2005)126-133
11. Andrade, E., Blunsden, S., Fisher, B.: Hidden Markov Models for Optical Flow Analysis in Crowds. the 18th international conference on pattern recognition, (2006) accepted
12. Cohen, I., Ayache, N., Sulger, P.: Tracking points on deformable objects using curvature information. Proceedings of the Second European Conference on Computer Vision, Springer-Verlag London, UK,(1992)458-466
13. Harris, C., Stephens, M.: A combined corner and edge detector. Proc. 4th Alvey Vision Conf. (1988)189-192
14. Mokhtarian, F., Mackworth A.K.: A Theory of Multi-Scale, Curvature-Based Shape Representation for Planar Curves. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 14, no. 8, (1992)789-805
15. Young, I.T., Vliet, L.J. van: Recursive implementation of the Gaussian filter. Signal Processing, (44) (1995)139-151
16. Stauffer C., Grimson, W. E. L.: Adaptive background mixture models for real-time tracking. In Computer Vision Pattern Recognition, Ft. Collins, CO,(1999)246--252
17. EC Funded CAVIAR project/IST 2001 37540, found at URL:http://homepages.inf.ed.ac.uk /rbf /CAVIAR/.

Analysing Crowd Intelligence

B.Zhan, P.Remagnino, and S.A.Velastin
{B.Zhan,P.Remagnino,Sergio.Velastin}@kingston.ac.uk

DIRC, Kingston University, UK

Abstract. Crowded environments are extremely interesting to analyse and model. Scenes of crowds appear to be chaotic; our main research interest is to demonstrate that crowd dynamics is inherently intelligent and their dynamics follow the modes of a density function apparently very noisy but hiding well defined paths. This work falls within the category of automatic scene understanding. In extreme conditions only moving people are visible from monitoring cameras and it is virtually impossible or unrealistic to model the static background. This makes very hard the detection of motion in the scene, and the analysis of its dynamics. We present a method, based on optical flow that can capture the dynamics of a complex scene and we employ a technique to identify the main paths of people frequenting a very busy environment. Ambient Intelligence can benefit from such an automatic analysis by keeping track of a very busy shopping mall, concourse, street, station, to draw statistics for business dynamics, laying optimal paths for disabled people or identifying and flagging unanomalous behaviour.

1 Introduction

A large number of publications focus on basic and robust image processing techniques to identify events of interest and track them throughout a more or less complex network of cameras, installed to monitor a public or private area (shopping malls are a classic example). There are two main problems with current methods: (i) they make the crucial assumption that video data streams can be split in two stochastic processes, one for the background signal and the other one for the foreground signal, (ii) they use video data streams where an individual can be identified and tracked, more or less easily.

Although tracking methods have evolved during the last decade - particle filters superseded banks of Kalman filters [1], techniques such as MeanShift have proved robust and reliable in complex scenes (American football examples) [2] - one can not possibly ignore that in highly crowded scenes, such as those in metro or train stations, the background signal can rarely be learned, and, although, in theory this is possible by using a very long video footage, one can not trust its reliability under continuous changes in illumination and/or variable atmospheric conditions.

The understanding and simulation of crowd dynamics is not new, chaos theory and complex dynamic models have been proposed in the literature, examples

include the work of Still [3], Helbing [4][5] and Musse [6]. Our approach is complementary, as we are interested in discovering and learning dynamics' models from raw video data. Our research is in line with some previous work of the authors on behaviour analysis [7] and some existing work on modelling complex behaviours [8][9]. In this new context, the concept of behaviour is generalised to complex dynamics.

In this paper we propose a method that discovers the main paths (modes) of a crowded scene, and builds two density functions, that keep track of the occurrence (PDF^{π}) of the signal and the local direction (PDF^{ϕ}), simply using a conventional block matching technique. Once the two PDF are built, then a recovering algorithm combines the stochastic information to reconstruct the main paths. These are then evaluated using an information theoretic dissimilarity measure. Two examples are shown in Figure 1. In these scenes the background

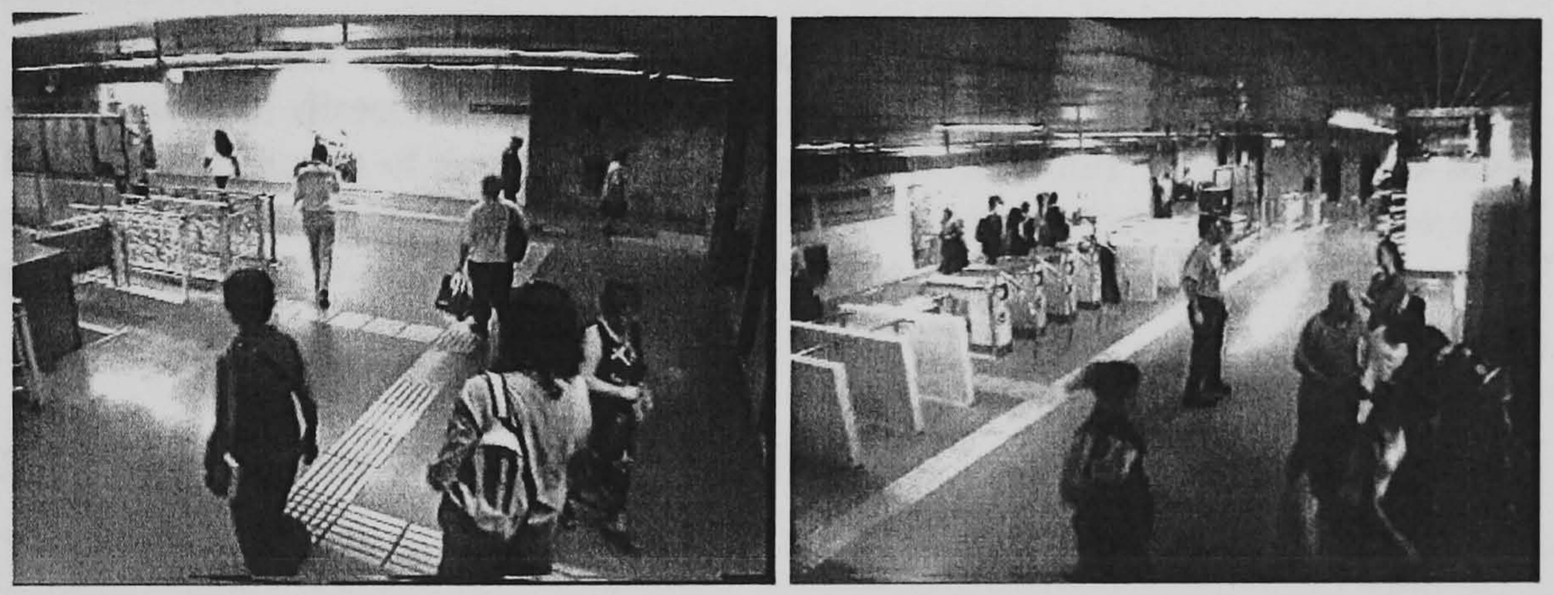


Fig. 1. Two frames of typical video data.

model might be learned, but the lighting conditions and video data quality make it extremely hard to rely on the built model.

The paper is organised as follows. Section 2 describes the algorithm; Section 3 discusses how paths can be extracted, while Section 4 describes an alternative method to estimate the density functions. Section 5 reports on preliminary results of the proposed algorithm tested on a few scenes. Concluding remarks are given in Section 6.

2 Conventional method

A probability density function for occurrence of foreground is constructed. This entails building the now well known pixel-based multivariate model of image dynamics, use of connected components to remove noise and the populating of an accumulator that, normalised to unit volume, represents a discrete probability density function of the occurrence of the foreground from the single view.

Attempts to segment such PDF have been tried before, but in this paper we leave the PDF as is. This is because segmenting continuous paths is not particularly interesting or useful. Also, we split the image into cells which might be interpreted as a fine and unorganised segmentation of the PDF. Finally, we prefer to keep an implicit representation of the PDF. In step 2 a PDF is built on the direction of structure. Structure for us are foreground connected components that move in the scene and whose local motion can be estimated by some similarity measure between consecutive frames. In this first implementation, a conventional block matching technique was implemented to identify the next position of the foreground data. Each cell/block in the image is then associated with a discretised orientation histogram, representing the occurrence of direction over the analysed sequence. In step 3 paths are discovered, by merging the information of both PDFs. In step 4 paths are approximated by spline curves and masks generated to rapidly calculate the FG blob-path distance and estimate fitting error.

The rationale of the outlined approach is justified by the need to identify main paths of direction in a complex scene, regardless of individual dynamics. Discovering modes of dynamics in a complex scene could be employed to build a coarse natural language narration of the scene and used to identify anomalies, such as people going in an unusual direction.

2.1 Occurrence PDF (PDF^{π})

It is unrealistic to precompile a background model of a complex real world scene, such as those video recorded by security cameras in public spaces. This is because of sudden or continuous changes in illumination, shadows and noise in the video signals. We have therefore adopted the Gaussian mixture model proposed by Stauffer [10] [11] that builds a dynamic and updatable background scene model on a pixel basis. The use of Stauffers algorithm allows a robust identification of foreground data. This foreground detector assumes background can be built, and therefore that the background stationary part of the scene can be seen over a large number of frames. This might not be the case in more complex scenes, for which a crowd might make invisible the background. In such cases other techniques will need to be employed. The foreground data is further processed to reduce noise. In particular, connected components have been implemented. Connectivity of foreground pixels gives more robustness to the foreground data and assures that only large foreground blobs are accepted for further analysis, while smaller blobs are rejected as likely noise.

For each frame we accumulate foreground features for every pixel, so that after a relatively long video sequence we have the accumulator of the foreground occurrence throughout the whole image. Figure 2 illustrates a typical occurrence PDF. The image can be segmented into cells, to speed the process of estimation of the PDF.

Fig. 2. Typical occurrence PDF.

2.2 Orientation PDF (PDF^{ϕ})

The image plane is segmented into a regular grid of cells (NxM). The dimension of each cell is a multiple of 2 and each cell is square-shaped (KxK). The idea is to speed up the matching process employed as a coarse estimator of motion between frames. Motion is estimated between consecutive frames, using the foreground blocks of the first frame as a reference/template and searching for an optimal match in the second frame. In the current implementation, block matching is carried out in a $3x3$ neighbourhood, around the selected foreground cell. A cell is labelled as foreground if the majority of its pixels are indeed foreground. Matching performance is improved by matching only between foreground cells, ignoring background cells.

A correlation measure [12] is used to calculate the distance between cells. The correlation method we used, for each pixel:

$$C(pixel) = \frac{1}{1 + \|p_1 - p_2\|} \tag{1}$$

Where p_1 and p_2 are respectively the pixel in the reference cell and the pixel in the neighbouring cell. Correlation for an entire cell is then calculated by summing over all the pixels of the cell: $C(cell) = \sum C(pixel)$. Each cell is therefore associated with a histogram representing the eight possible directions of motion. The intention here is to build a local representation of motion, similar to a discrete reinforcement learning technique [13] , where each cell of the table has associated a quality array, indicating the likelihood of transition from the current cell to a neighbouring cell. The final outcome is an orientation PDF, which could be interpreted as the global optical flow of the scene.

3 Path discovery

The work described in the previous sections provides two PDFs: one for the occurrence and one for the orientation of a scene. To discover the main paths,

we need to combine the information and extract those corresponding to higher likelihood/probability. Ideally we would like to identify the paths corresponding to the modes of a probability density function that combines both occurrence and orientation information.

In order to estimate the main paths we make a number of assumptions.

Path origin: we make the assumption that all paths originate from the boundaries of the scene. Consequently path discovery is started from a cell the boundary of the scene and having high occurrence probability. This assumption would not work if the scene had an entrance or exit in the middle of the image, but this can be overcome relatively easily by using user-defined boundaries.

Graceful continuation/Smooth trajectory: We observed that paths have a high probability to maintain their orientation (e.g. people are more likely to go on a straight line, and seldom go backwards.) So we model the expected direction of motion with a *Poisson* distribution [14], with its maximum in the neighbouring cell along the current direction of motion.

The idea is to spread the likelihood of change in direction unevenly, maintaining the previous orientation as the one at highest probability and forcing the other directions (change in direction) to have a lower likelihood. Table 1 illustrates the probabilities used given the distance from the current orientation.

From the start point, we calculate the probability for each neighbouring block using the occurrence pdf (PDF_{occ}), the block matching accumulator (P_b) and the orientation probability (PDF_{or}). Furthermore, to avoid repeating calculations from the same block, we mark the visited cells, and set their probability to 0 each time the path discovery process has to deal with them. The probability of each neighbouring cell $i : i \in [0..8]$ to be the next path cell is:

$$P_i = \frac{m_i \cdot PDF_i^{\pi} \cdot P_i^b \cdot PDF_i^{\phi}}{\sum_k m_k \cdot PDF_k^{\pi} \cdot P_k^b \cdot PDF_k^{\phi}} \tag{2}$$

where

$$k \in [0..8], m_i = \begin{cases} 1\ marked \\ 0\ unmarked \end{cases} \tag{3}$$

The process will follow the highest probability block and stop at a probability $\epsilon : \epsilon \to 0$. We also devised a way of deciding when to split a trajectory in two or more sub-trajectories. This technique works on a threshold that estimates whether two or more paths are viable given their associated likelihoods. However, we enforce only a single split along a trajectory, so as not to generate too many branches.

4 The Alternative method

This section describes an alternative method. This applies the conventional block matching technique described in Section 2.2 to estimate a coarse estimation of the motion field. The magnitude of such field is used to build the PDF^{π}.

In a very complex scene it may be very hard or impossible to discriminate between the foreground and the background stochastic processes. In order to analyse the dynamics of the scene, we developed an alternative method to estimate the occurrence and orientation density functions, using the optical flow to build the densities.

4.1 Optical Flow in brief

An image $f(x, y, t)$ refers to the grey level of (x, y) at time t. Representing a dynamic image as a function of position and time permits it to be expressed as a Taylor series.

$$f(x + dx, y + \Delta_y, t + \Delta_t) = f(x, y, t) + f_x\Delta_x + f_y\Delta_y + f_t\Delta_t + O(\delta^2) \quad (4)$$

f_x, f_y and f_z can be approximated, from $f(x, y, t)$.The brightness constraint equation is thus

$$f(x + \Delta_x, y + \Delta_y, t + \Delta_t) = I(x, y, t) \quad (5)$$

The motion velocity can then be estimated as

$$-f_t = f_x u + f_y v \quad (6)$$

As this is only one equation for two flow components, the optical flow is not uniquely determined by this constraint. Many optical flow estimation techniques exist. In this paper we employed the method proposed by Horn and Schunck [15], in which the optic flow constraint is embedded in the global energy functional.

$$E_S = \int_\Omega ((f_x u + f_y v + f_t)^2 + \alpha(|\Delta u|^2 + |\Delta v|^2))dxdy \quad (7)$$

4.2 Occurrence PDF

Occurrence PDF is built using an accumulator of the motion occurrence

$$\hat{v} = \sqrt{u^2 + v^2} \quad (8)$$

For each frame we accumulate the motion value for every pixel, so that after a relatively long video sequence we have the accumulator of the foreground occurrence throughout the whole image.

4.3 Orientation PDF

We have a horizon velocity map and a vertical velocity map for each pixel. To evaluate the orientation information we employed a 8-direction scheme.

We calculate the tangent of the direction angle by (u, v), and fit them to the 8-direction area according to the angle

$$\alpha = \arctan\frac{v}{u}, (u \neq 0) \quad (9)$$

We accumulate the motion vector corresponding to the current direction.

5 Experimental results

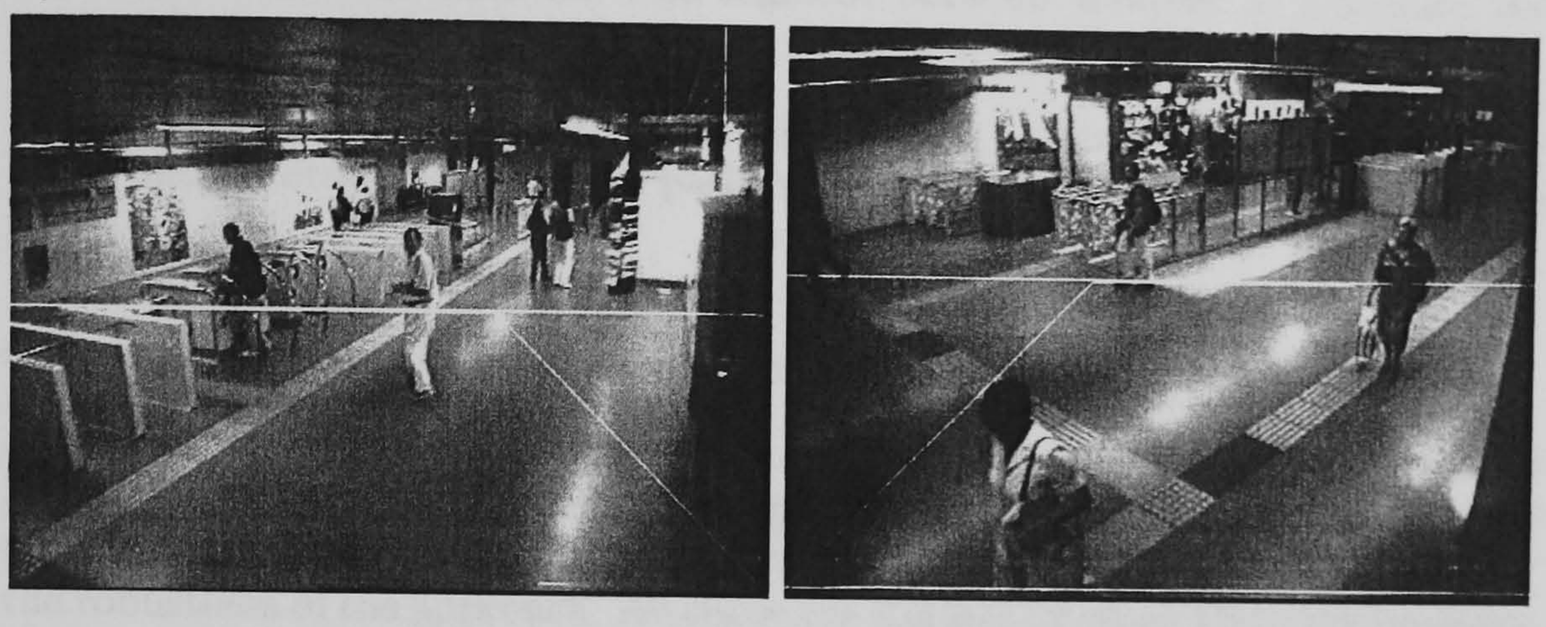


Fig. 3. Two examples of extracted paths.

Paths extracted using the method described in the previous sections correspond to the main modes of trajectories followed by people in the analysed scene. Rather than using the two PDFs (occurrence and orientation) to estimate an error and evaluate the performance of the technique, we provide a simplififed evaluation. We employed the idea of a stripe along the discovered paths using a decay factor (a Gaussian weighting) along the perpendicular to the trajectory.

Examples of automatically extracted paths are shown in Figure 3. These are preliminary and qualitative results, that indicate that some of the paths are identified and some others are misclassified as paths, mainly due to recurring noise, partially caused by regular illumination changes.

5.1 Support masks

The stripe is illustrated pictorially in the Figure 4. Suppose the black area represents the discovered path $f = f(x, y, t)$. A Gaussian distribution $G(\mu, \sigma)$ is then centred on the trajectory (μ corresponding to the generic path pixel), and σ being a predetermined standard deviation directly proportional to the size of the blobs estimated by the connected component process.

An approximated estimate of the error between a new sequence of the same scene and the built model can then be calculated by weighting the contribution of a foreground blob, making use of the described weighting scheme. Since error estimation can be performed off-line, when the model already exists, a mask for the entire image can be built before testing.

Masks are built once for all at the end of the path modelling process.

– We build an image look-up table (LUT), where each pixel is assigned a label, identifying the closest path in the scene.

– For each path we build a stripe mask. The mask contains the weights, inversely proportional to the distance between a pixel and the path/spline. To calculate the weights we sampled the curve of the path at equally spaced intervals Δt and used the line segment between samples to calculate the weight.

For each FG blob we detected, we examine it pixel by pixel with the image label LUT, and determine the closest path by taking the most frequent label of its pixels.

5.2 Measuring goodness of fit: an information theoretic approach

A number of scenes have been analysed and following the conventional machine learning approach each sequence was split in two halves, to build and test the model. Different percentages of frames to build the model were used, and to test the robustness of the approach. We chose the Kullback-Leibler (KL) dissimilarity measure to estimate the similarity between the PDF_{occ} and PDF_{or} of the model and the corresponding PDFs built using a fixed percentage of test data.

$$\hat{D}_{KL} = (D(PDF^{model}||PDF^{test}) \oplus D(PDF^{test}||PDF^{model})) \tag{10}$$

where

$$D_{LK}(p||q) = \sum_t p(t) \log_2 \frac{p(t)}{q(t)} \tag{11}$$

and, for PDF_{occ} the sum is over the entire image, and for PDF_{or} is a weighted sum over all the cells. The following table illustrates some preliminary results. The table illustrates results for two scenes, indicating the dissimilarity for PDF_{π} and PDF_{ϕ} independently. The composit, shown with the symbol $\oplus$, is a type of balanced non-negative dissimilarity measure that, in theory, is zero for $p \equiv q$, and should decrease as the model is refined and better represents the studied scene. These preliminary outcomes illustrate that a decreasing trend is present for PDF_{ϕ} but not quite for PDF_{π}. The number of frames we used is still fairly low. Our next goal will be to use longer sequences, for instance as long as hours.

6 Conclusions and Future Work

Ambient Intelligence requires the use of machine vision to interpret visual dynamics and produce a natural language description of unfolding events in a complex scene. This is possible only if an automatic interpretation is in place. In this paper we wanted to prove that a spatial-temporal model of the main modes of dynamics can be captured simply, without the use of a tracker. This is important, as a tracker might not work in very cluttered scenes. Approximating the main paths, means generating a model of normality which can in turn be used to identify anomalies.

This is the very first step towards a formalisation of crowd dynamics. We firmly believe that density estimation of dynamics can be built and left in implicit

Scene 1						
	PDF_π			PDF_ϕ		
N_{frames}	$D(p\|\|q)$	$D(q\|\|p)$	$D(p\|\|q) \oplus D(q\|\|p)$	$D(p\|\|q)$	$D(q\|\|p)$	$D(p\|\|q) \oplus D(q\|\|p)$
200	1.38744	6.41181	3.89963	0.992265	3.71003	2.35115
400	1.22145	4.72941	2.97543	0.74336	2.3779	1.56063
600	1.30149	4.22187	2.76168	0.550128	0.870776	0.710452
800	5.32319	1.50177	3.41248	0.960938	0.405281	0.68311
1000	5.43141	1.37666	3.40404	1.61853	0.448835	1.03368
1200	5.83901	1.39313	3.61607	2.20614	0.508669	1.3574
1400	5.87275	1.38677	3.62976	2.48443	0.547993	1.51621

Scene 2						
	PDF_π			PDF_ϕ		
N_{frames}	$D(p\|\|q)$	$D(q\|\|p)$	$D(p\|\|q) \oplus D(q\|\|p)$	$D(p\|\|q)$	$D(q\|\|p)$	$D(p\|\|q) \oplus D(q\|\|p)$
200	1.76456	4.49901	3.13179	1.07256	6.88155	3.97705
400	1.75548	3.60504	2.68026	0.665695	2.40205	1.53387
600	1.9534	2.89671	2.42506	0.434972	0.825922	0.630447
800	2.15736	2.32519	2.24128	0.395621	0.502726	0.449173
1000	3.9971	1.39806	2.69758	0.529596	0.429419	0.479507
1200	3.65854	1.29503	2.47678	0.653405	0.403238	0.528322
1400	3.62649	1.30134	2.46391	0.680506	0.403089	0.541798

form, the table shows some preliminary results and a possible way of evaluating the goodness of fit of the estimated functions.

Future work will include the refinement of the current model and a clustering of the paths.

References

1. B.Ristic, S.Arulampalam, N.Gordon: Beyond the Kalman Filter: particle filters for tracking applications. Artech House (2004)
2. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: IEEE Conference on Computer Vision and Pattern Recognition. Volume 2. (2000) 142–149
3. Still, K.: Crowd Dynamics. PhD thesis, University of Warwick (2000)
4. D.Helbing, L.Buzna, A.Johansson, T.Werner: Self-organised pedestrian crowd dynamics: experiments, simulations and design solutions. Transportation Science **39** (2005) 1–24
5. Helbing, D., Farkas, I., Vicsek, T.: Simulating dynamical features of escape panic. Nature **407** (2000) 487–490
6. Musse, S., Thalmann, D.: Hierarchical model for real time simulation of virtual human crowds. IEEE Transactions on Visualization and Computer Graphics **7** (2001) 152–164
7. P.Remagnino, T.Tan, K.Baker: Agent orientated annotation in model based visual surveillance. In: IEEE International Conference on Computer Vision. (1998) 857–862

8. Makris, D., Ellis, T.: Learning semantic scene models from observing activity in visual surveillance. IEEE Transactions on Systems Man and Cybernetics - Part B **35** (2005) 397–408
9. M.Brand: learning concise models of visual activity. Technical Report TR1997-025, MERL (1997)
10. C.Stauffer, W.E.L.Grimson: Adaptive background mixture models for real-time tracking. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Volume 2. (1999) 23–25
11. Stauffer, C., Grimson, W.E.L.: Learning patterns of activity using real-time tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (2000) 747–757
12. M.Sonka, V.Hlavac, R.Boyle: Image Processing: Analysis and Machine Vision. 2nd edn. Thomson Learning Vocational (1998)
13. S.Sutton, A.G.Barto: Reinforcement Learning: an Introduction. MIT Press (1998)
14. W.Mendenhall, J.B.Robert: Introduction to probability and statistics: study guide and solutions manual. Wadsworth Pub Co (1999)
15. Horn, B., Schunck, B.: Determining optical flow. Artificial Intelligence **17** (1981) 185–203

Mining Paths of Complex Crowd Scenes

B. Zhan, P. Remagnino, and S.A. Velastin
DIRC, Kingston University, UK
{B.Zhan, P.Remagnino, sergio.velastin}@kingston.ac.uk

Abstract. The Ambient Intelligence (AmI) paradigm requires a robust interpretation of people actions and behaviour and a way for automatically generating persistent spatial-temporal models of recurring events. This paper describes a relatively inexpensive technique that does not require the use of conventional trackers to identify the main paths of highly cluttered scenes, approximating them with spline curves. An AmI system could easily make use of the generated model to identify people who do not follow prefixed paths and warn them. Security, safety, rehabilitation are potential application areas. The model is evaluated against new data of the same scene.

1 Introduction

This paper describes the first steps towards automatic crowd analysis. Machine Vision research has been mainly concerned with accurate measurements of object dynamics and many algorithms have been proposed to track one of more individuals in more or less complex scenes. Not so long ago some researchers started to work on behaviour analysis, mainly concerned with the building of a reusable spatial-temporal model of a scene. Notable work is research carried out to identify patterns in time series of people working in an office, people and vehicles moving in a car park [1][2]. The basic problem with these approaches is that they tend to rely on accurate information extracted by trackers [3], or they make use of coarse information, extracted from video data of individuals or small numbers of people frequenting the analysed environment. What we are interested in are highly cluttered scenes, with many people moving about, with no apparent structure, such as those of large crowds recorded in highly frequented public spaces, such as railway or metro stations. This paper presents an initial study on how to tackle the described scenarios with simple machine vision algorithms that do not require sophisticated image understanding processing algorithms and that can be eventually implemented in hardware. Two examples are shown in Figure 1.

The paper is the first step to bridge two worlds: on the one hand machine vision research that attempts to deliver stochastic models of dynamics while on the other hand mathematical modelling of dynamics, such as fluid or aerodynamics, recently employed to describe the complex and apparent chaotic crowd dynamics [4][5][6][7]. Here we make use of simple image processing techniques to extract foreground data of a dynamic scene. We then build the probability

G. Bebis et al. (Eds.): ISVC 2005, LNCS 3804, pp. 126–133, 2005.

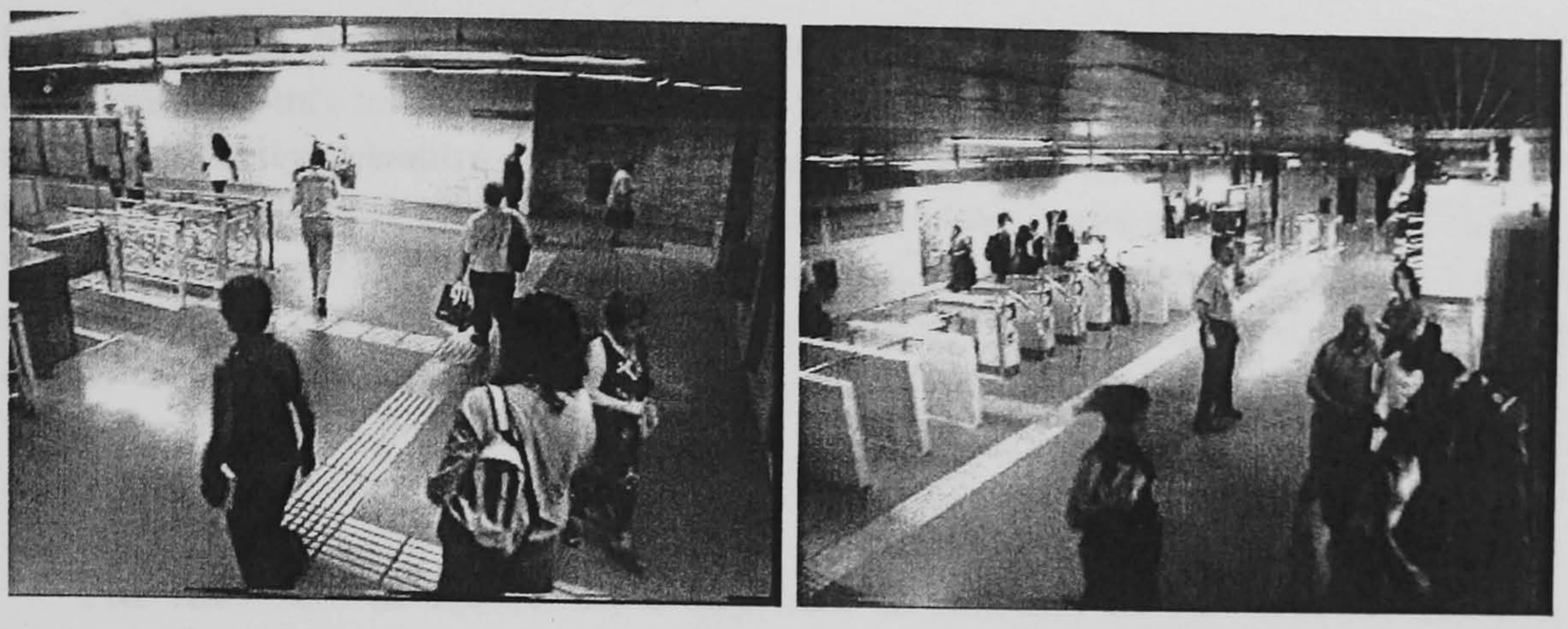


Fig. 1. Two frames of typical video data

distribution function (PDF) of the occurrence of the detected foreground and the motion orientation of the foreground so as to build a local model for it. We then make use of the two PDFs to trace the main paths of people who frequented the scene. These paths are then considered as the modes of paths in the scene and uncertainty around them is used to estimate an error measure to evaluate the performance of the proposed algorithm.

The paper is organised as follows. Section 2 describes the algorithm; Section 3 discusses how paths can be extracted, and spline curves can be employed to interpolate the extracted paths. Section 4 reports on preliminary results of the proposed algorithm tested on a few scenes. Concluding remarks are given in Section 5.

2 Proposed Method

The proposed method can be summarised in the following steps, described in later sections of the paper:

- Occurrence PDF: foreground detection, connected components, accumulator,
- Orientation PDF: correlation matrix, accumulator of block matching,
- Path discovery: previous orientation, probability calculation, path split.
- Path fitting: spline interpolators, path masks.

In step 1 a probability density function for occurrence of foreground is constructed. This entails building the now well known pixel-based multivariate model of image dynamics, use of connected components to remove noise and the populating of an accumulator that, normalised to unit volume, represents a discrete probability density function of the occurrence of the foreground from the single view. Attempts to segment such PDF have been tried before, but in this paper we leave the PDF as is. This is because segmenting continuous paths is not particularly interesting or useful. Also, we split the image into cells which might be interpreted as a fine and unorganised segmentation of the PDF. Finally, we prefer to keep an implicit representation of the PDF. In step 2 a PDF

is built on the direction of structure. Structure for us are foreground connected components that move in the scene and whose local motion can be estimated by some similarity measure between consecutive frames. In this first implementation, a conventional block matching technique was implemented to identify the next position of the foreground data. Each cell/block in the image is then associated with a discretised orientation histogram, representing the occurrence of direction over the analysed sequence. In step 3 paths are discovered, by merging the information of both PDFs. In step 4 paths are approximated by spline curves and masks generated to rapidly calculate the foreground blob-path distance and estimate fitting error.

The rationale of the outlined approach is justified by the need to identify main paths of direction in a complex scene, regardless of individual dynamics. Discovering modes of dynamics in a complex scene could be employed to build a coarse natural language narration of the scene and used to identify anomalies, such as people going in an unusual direction.

2.1 Occurrence PDF

It is unrealistic to precompile a background model of a complex real world scene, such as those video recorded by security cameras in public spaces. This is because of sudden or continuous changes in illumination, shadows and noise in the video signals. We have therefore adopted the Gaussian mixture model proposed by Stauffer [8] [9] that builds a dynamic and updatable background scene model on a pixel basis. The use of Stauffers algorithm allows a robust identification of foreground data. This foreground detector assumes background can be built, and therefore that the background stationary part of the scene can be seen over a large number of frames. This might not be the case in more complex scenes, for which a crowd might make invisible the background. In such cases other techniques will need to be employed. The foreground data is further processed to reduce noise. In particular, connected components have been implemented. Connectivity of foreground pixels gives more robustness to the foreground data and assures that only large foreground blobs are accepted for further analysis, while smaller blobs are rejected as likely noise.

For each frame we accumulate foreground features for every pixel, so that after a relatively long video sequence we have the accumulator of the foreground occurrence throughout the whole image. Figure 2 illustrates a typical occurrence PDF. The image can be segmented into cells, to speed the process of estimation of the PDF.

2.2 Orientation PDF

The image plane is segmented into a regular grid of cells ($N \times M$). The dimension of each cell is a multiple of 2 and each cell is square-shaped ($K \times K$). The idea is to speed up the matching process employed as a coarse estimator of motion between frames. Motion is estimated between consecutive frames, using the foreground blocks of the first frame as a reference/template and searching for an optimal

Fig. 2. Typical occurrence PDF

match in the second frame. In the current implementation, block matching is carried out in a 3×3 neighbourhood, around the selected foreground cell. A cell is labelled as foreground if the majority of its pixels are indeed foreground. Matching performance is improved by matching only between foreground cells, ignoring background cells.

A correlation measure [10] is used to calculate the distance between cells. Correlation for an entire cell is then calculated by summing over all the pixels of the cell:

$$D = \sum_{x,y \in C} \frac{1}{1 + \|P - P'\|} \tag{1}$$

where P and P' are respectively the pixel in the reference cell and the pixel in the neighbouring cell. The measurement of the correspondence between two cells uses the normalised cross correlation. This results in a distance falling in the $(0, 1]$ range. The distance is 1 when the two cells are exactly the same and becomes very small when the two cells bear large differences.

Each cell is therefore associated with a histogram representing the eight possible directions of motion. The intention here is to build a local representation of motion, similar to a discrete reinforcement learning technique [11] , where each cell of the table has associated a quality array, indicating the likelihood of transition from the current cell to a neighbouring cell. The final outcome is an orientation PDF, which could be interpreted as the global optical flow of the scene.

3 Path Discovery

The work described in the previous sections provides two PDFs: one for the occurrence and one for the orientation of a scene. To discover the main paths, we need to combine the information and extract those corresponding to higher likelihood/probability. Ideally we would like to identify the paths corresponding to the modes of a probability density function that combines both occurrence and orientation information.

In order to estimate the main paths we make a number of assumptions.

Path origin: we make the assumption that all paths originate from the boundaries of the scene. Consequently path discovery is started from a cell the bound-

ary of the scene and having high occurrence probability. This assumption would not work if the scene had an entrance or exit in the middle of the image, but this can be overcome relatively easily by using user-defined boundaries.

Graceful continuation/Smooth trajectory: We observed that paths have a high probability to maintain their orientation (e.g. people are more likely to go on a straight line, and seldom go backwards.) So we model the expected direction of motion with a *Poisson* distribution, with its maximum in the neighbouring cell along the current direction of motion.

The idea is to spread the likelihood of change in direction unevenly, maintaining the previous orientation as the one at highest probability and forcing the other directions (change in direction) to have a lower likelihood. Table 1 illustrates the probabilities used given the distance from the current orientation.

Table 1. Likelihood as function of orientation distance

d	0	1	2	3	4
P_d	0.6830	0.1335	0.02	0.0045	0.0001

From the start point, we calculate the probability for each neighbouring block using the occurrence pdf (PDF_{occ}), the block matching accumulator represents the orientation probability, that is PDF_{or}, and also the direction likelihood P^d. Furthermore, to avoid repeating calculations from the same block, we mark the visited cells, and set their probability to 0 each time the path discovery process has to deal with them. The probability of each neighbouring cell $i : i \in [0..8]$ to be the next path cell is:

$$P_i = \frac{m_i \cdot PDF_i^{occ} \cdot P_i^d \cdot PDF_i^{or}}{\sum_k m_k \cdot PDF_k^{occ} \cdot P_k^d \cdot PDF_k^{or}} \quad where \quad k \in [0..8], m_i = \begin{cases} 1\ marked \\ 0\ unmarked \end{cases}$$

The process will follow the highest probability block and stop at a probability $\epsilon : \epsilon \to 0$. We also devised a way of deciding when to split a trajectory in two or more sub-trajectories. This technique works on a threshold that estimates whether two or more paths are viable given their associated likelihoods. However, we enforce only a single split along a trajectory, so as not to generate too many branches.

Once all paths are identified, a fitting process takes place. This serves two purposes: (i) to have a compact representation of the path, (ii) to have a faster way of estimating the distance between a blob/bounding rectangle, identified by new foreground data, and the spline, and consequently estimating an error. The following figure (Figure 3) illustrates splines approximating the identified paths. The scene of Figure 3 left is highly complex, due to clutter, poor illumination and reflections. Although some of the paths are incorrect, most paths reflect the main dynamics of the scene: people moving from the gates to the exit and viceversa. Future implementations will include a refining process of the paths. Apriori knowledge about the scene might also help, if semi-automatic analysis was enabled.

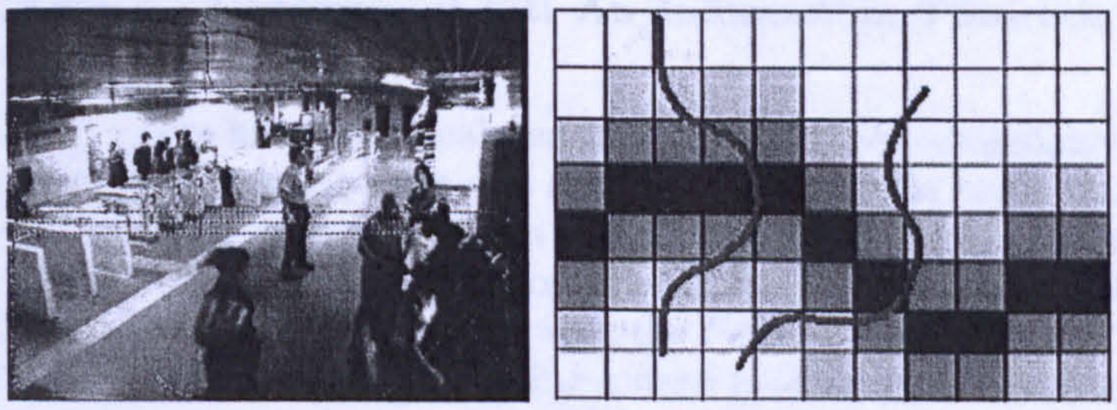


Fig. 3. Left: spline interpolators superimposed on a sequence frame. Right: The stripe.

4 Experimental Results

Paths extracted using the method described in the previous sections correspond to the main modes of trajectories followed by people in the analysed scene. Rather than using the two PDFs (occurrence and orientation) to estimate an error and evaluate the performance of the technique, we provide a simplififed evaluation. We employed the idea of a stripe along the discovered paths using a decay factor (a Gaussian weighting) along the perpedicular to the trajectory.

4.1 Support Masks

The stripe is illustrated pictorially in the Figure 3 right. Suppose the black area represents the discovered path $f = f(x, y, t)$. A Gaussian distribution $G(\mu, \sigma)$ is then centred on the trajectory (μ corresponding to the generic path pixel), and σ being a predetermined standard deviation directly proportional to the size of the blobs estimated by the connected component process.

An approximated estimate of the error between a new sequence of the same scene and the built model can then be calculated by weighting the contribution of a foreground blob, making use of the described weighting scheme. Since error estimation can be performed off-line, when the model already exists, a mask for the entire image can be built before testing.

Masks are built once for all at the end of the path modelling process.

- We build an image look-up table (LUT), where each pixel is assigned a label, identifying the closest path in the scene.
- For each path we build a stripe mask. The mask contains the weights, inversely proportional to the distance between a pixel and the path/spline. To calculate the weights we sampled the curve of the path at equally spaced intervals Δt and used the line segment between samples to calculate the weight.

For each FG blob we detected, we examine it pixel by pixel with the image label LUT, and determine the closest path by taking the most frequent label of its pixels.

4.2 Measuring Goodness of Fit: An Information Theoretic Approach

A number of scenes have been analysed and following the conventional machine learning approach each sequence was split in two halves, to build and test the model. Different percentages of frames to build the model were used, and to test the robustness of the approach. We chose the Kullback-Leibler (KL) dissimilarity measure to estimate the similarity between the PDF_{occ} and PDF_{or} of the model and the corresponding PDFs built using a fixed percentage of test data.

$$\hat{D}_{KL} = (D(PDF^{model}||PDF^{test}) \oplus D(PDF^{test}||PDF^{model})) \qquad (2)$$

where $D_{LK}(p||q) = \sum_t p(t) \log_2 \frac{p(t)}{q(t)}$ and, for PDF_{occ} the sum is over the entire image, and for PDF_{or} is a weighted sum over all the cells. The following table illustrates some preliminary results. Table 2 illustrates results for a scene,

Table 2. Result of goodness of fit

	PDF_{occ}			PDF_{or}		
N_{frames}	$D(p\|\|q)$	$D(q\|\|p)$	$D(p\|\|q) \oplus D(q\|\|p)$	$D(p\|\|q)$	$D(q\|\|p)$	$D(p\|\|q) \oplus D(q\|\|p)$
200	1.38744	6.41181	3.89963	0.992265	3.71003	2.35115
400	1.22145	4.72941	2.97543	0.74336	2.3779	1.56063
600	1.30149	4.22187	2.76168	0.550128	0.870776	0.710452
800	5.32319	1.50177	3.41248	0.960938	0.405281	0.68311
1000	5.43141	1.37666	3.40404	1.61853	0.448835	1.03368
1200	5.83901	1.39313	3.61607	2.20614	0.508669	1.3574
1400	5.87275	1.38677	3.62976	2.48443	0.547993	1.51621

indicating the dissimilarity for PDF_{occ} and PDF_{or} independently. The composit, shown with the symbol $\oplus$, is a type of balanced non-negative dissimilarity measure that, in theory, is zero for $p \equiv q$, and should decrease as the model is refined and better represents the studied scene. These preliminary outcomes illustrate that a decresing trend is present for PDF_{or} but not quite for PDF_{occ}. The number of frames we used is still fairly low. Our next goal will be to use longer sequences, for instance as long as hours.

5 Conclusions and Future Work

In this paper we wanted to prove that a spatial-temporal model of the main modes of dynamics can be captured simply, without the use of a tracker. This is important, as a tracker might not work in very cluttered scenes. Approximating the main paths, means generating a model of normality which can in turn be used to identify anomalies. This is the very first step towards a formalisation of crowd dynamics. We firmly believe that density estimation of dynamics can be built and left in implicit form, the table shows some preliminary results and a possible way of evaluating the goodness of fit of the estimated functions.

References

1. Makris, D., Ellis, T.: Learning semantic scene models from observing activity in visual surveillance. IEEE Transactions on Systems Man and Cybernetics - Part B **35** (2005) 397–408
2. M.Brand: learning concise models of visual activity. Technical Report TR1997-025, MERL (1997)
3. Buzan, D., Sclaroff, S., Kollios, G.: Extraction and clustering of motion trajectories in video. In: Proceedings of the 17th International Conference on Pattern Recognition. Volume 2. (2004) 521–524
4. Still, K.: Crowd Dynamics. PhD thesis, University of Warwick (2000)
5. D.Helbing, L.Buzna, A.Johansson, T.Werner: Self-organised pedestrian crowd dynamics: experiments, simulations and design solutions. Transportation Science **39** (2005) 1–24
6. Helbing, D., Farkas, I., Vicsek, T.: Simulating dynamical features of escape panic. Nature **407** (2000) 487–490
7. Musse, S., Thalmann, D.: Hierarchical model for real time simulation of virtual human crowds. IEEE Transactions on Visualization and Computer Graphics **7** (2001) 152–164
8. C.Stauffer, W.E.L.Grimson: Adaptive background mixture models for real-time tracking. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Volume 2. (1999) 23–25
9. Stauffer, C., Grimson, W.E.L.: Learning patterns of activity using real-time tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (2000) 747–757
10. M.Sonka, V.Hlavac, R.Boyle: Image Processing: Analysis and Machine Vision. 2nd edn. Thomson Learning Vocational (1998)
11. S.Sutton, A.G.Barto: Reinforcement Learning: an Introduction. MIT Press (1998)