The University of Newcastle
School of Electrical Engineering and Computer Science
Faculty of Engineering and Built Environment

Time Series Classification for Analysing the Impact of Architectural Design on Pedestrian Spatial Behaviour

by

Arash Jalalian

BSc and MSc in Computer Engineering

Thesis Advisor: Stephan K. Chalup

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy in Computer Science

June, 2012

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. I give consent to this copy of my thesis, when deposited in the University Library[1], being made available for loan and photocopying subject to the provisions of the Copyright Act 1968.

Signature

[1] Unless an Embargo has been approved for a determined period.

Acknowledgments

I consider myself very fortunate to have had the opportunity to live and study in Newcastle. With its natural beauty, Newcastle provides an ideal setting for academic excellence. Not only I have received world-class training in modern machine learning, I have had the pleasure of forming many close friendships and exciting collaborations amongst the numerous highly esteemed academics I have worked with.

As is ever so common in research today, progress is rarely made without collaboration amongst colleagues from various areas within the local and distant academic community. I would like to take this opportunity to thank all those who have contributed to the fields of scientific endeavour perused throughout this thesis. Not only those mentioned below but the countless workers that dedicate their talents to this respectively important field.

First and foremost, I would like to express my deepest gratitude to Associate Professor Stephan Chalup for granting me this opportunity. He has continuously instilled confidence in me and has truly believed in my abilities. Under his guidance, I have been able to realise a dream and life long academic passion. His extensive experience has been invaluable and his level of patience, not only admirable but critical to my success.

Professor Michael Ostwald has provided uniquely insightful suggestions and I appreciate the experience in the architectural application domain that he has facilitated. His inspirational personality and extraordinary intelligence is priceless in guiding students. The level of attention and genuine care he expresses is remarkable.

I would like to thank to Doctor Eamonn Keogh, and Doctor A. Frank for providing very useful machine learning data sets in UCR and UCI repositories. In addition, I would like to thank Doctor Sepp Hochreiter and Doctor Klaus Obermayer for introducing P-SVMs. To all paper reviewers who have provided outstanding feedback.

Newcastle City Council and Screen Hunter Central Coast have been very helpful in supporting the Human Ethics approval process. I gratefully acknowledge Roseanne Linich for providing ESL support during my doctorate studies.

This work would not have been possible without the financial support provided by the Australian Research Council - "Modelling and Predicting Patterns of Pedestrian Movement". Grant No. DP1092679.

On a personal note, I express my appreciation to my colleagues in the Interdisciplinary Machine Learning Research Group, for their stimulating discussion, ideas, encouragement, expertise and genuinely keen approach to team work. To my friends for being understanding of my commitment during this period, especially as I complete this thesis by writing this acknowledgement. Of special note Doctor Houman Ebrahimi, my dear friend and imminent business co-founder.

Being away from family has not been easy but this endeavour has not only been possible through focus and dedication but it has taken years of preparation and ongoing support from my dear family. My father Hassan Jalalian, my mother Mansure Azam Feizmanesh and my brother Ashkan Jalalian. They have been amazing in bridging the immense physical distance that separates us. Their supports and prayers have truly been invaluable.

Time Series Classification for Analysing the Impact of Architectural Design on Pedestrian Spatial Behaviour

Abstract

Pedestrian spatial behaviour is defined as the pedestrians' reaction to their immediate surroundings. Analysis of changes in this behaviour due to alternation in the environmental settings is an important facet of architectural and urban design. To measure the changes, human body dynamics, such as head position, gaze direction, movement direction, speed of movement, and trajectory can be employed. In this research the main purpose is to support architects and urban designers to better assess the impact of the spatial environment ion the pedestrian's behaviour in planned urban spaces. To this end, an analysis system is proposed to learn the patterns of behaviour observed in a simulated and real-world architectural space.

The simulated environment is generated using the proposed pedestrian and urban models. The models provide important behavioural characteristics in a multi-agent-based simulation system. They support complex spatial interactions between agents and their environment, including agent-to-agent interactions, different spatial desires, and interpersonal distance. The simulated environment can be automatically generated using scanned line drawings of two-dimensional street maps or public spaces. In the simulation model, a variety of scenarios can be defined and modified by altering different parameters. Using the example of Wheeler Place in Newcastle (Australia), the experiments demonstrate how pedestrian behavioural characteristics can depend on selected abstract features in urban spaces. The characteristics are used in the analysis system to distinguish between different patterns of spatial behaviour.

The analysis system consists of a proposed technique for sequential data classification where each data object may have different lengths. The new technique, called GDTW-P-SVMs, is a maximum margin method for the construction of classifiers with variable-length input series. It employs potential support vector machines (P-SVMs) and dynamic time warping (DTW) to waive the fixed-length restriction of feature vectors in standard support vector machines (SVMs). The new technique elaborates on the P-SVM kernel function, by utilising DTW to provide an elastic distance measure for the kernel function. Benchmarks for classification are performed with several real-world data sets from the UCR Time Series Classification/Clustering page, GeoLife trajectory data set, and UCI Machine Learning Repository. The data sets include data with both variable and fixed-length input series. The results show that the new method performs significantly better than the benchmarked standard classification methods.

To learn patterns of spatial behaviour the proposed classification technique is employed with simulated and real-world characteristics. The characteristics are collected from Wheeler Place using the proposed simulation software and pedestrian tracking system. GDTW-P-SVMs classify patterns of behaviour using the whole sequence of data series as a single input to increase the classification performance. As a result, they can provide the highest classification accuracy using the simulated and real-world data sets, when compared with the other existing methods.

Keywords: Spatial Behaviour Analysis, Trajectory Data Analysis, Support Vector Machines, Dynamic Time Warping

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

In recent years, the automatic analysis of pedestrian behaviour has been attracting an increasing amount of attention from researchers because of its important applicative aspects and intrinsic scientific interest. At the same time, progress on sensors, sensor networking, computer vision, audio analysis and speech recognition are making available the building blocks for automatic behavioural analysis. This research has included the analysis of spatial behaviour, using human body dynamics while moving in urban space, to monitor the impact of visually attractive objects. The proposed analysis system is able to measure changes in pedestrians' spatial features and distinguish between several behavioural classes.

To analyse the behaviour, a selected set of body dynamics was simulated. The use of simulation software was proposed in order to avoid the difficulties associated with collecting noisy data from real-world environments, to collect as much data as we require to classify different categories of behaviour, to be able to change the environmental configuration as we require, and to ensure that our hypothesis of using human body dynamics to analyse pedestrian behaviour works with the simulated data.

The proposed simulation approach consists of two separated models; an urban model and a pedestrian model. Previously, the simulation of pedestrian body dynamics was based on one of two approaches: i) analytical approaches and ii) cellular automata. Analytical approaches use a mathematical formula to model the average behaviour of group of people. As human reactions are characterised by a significant degree of randomness, analytical models fail to simulate this randomness with mathematical functions. On the other hand, in cellular automata (CA) approaches the accuracy of the model depends on cell size used in CA. This implicitly restricts movement of the simulated agents to the centre of each cell.

The proposed simulation is a multi-agent system with the characteristic dynamics of a crowd of moving pedestrians in a section of an abstract urban 2-dimensional environment. In the simulation, each agent is able to have different behaviours with a large degree of freedom. A point-based movement approach is employed so the movements are not limited to the cell size. The first version of the simulation models only basic behavioural characteristics such as gaze vector, speed vector, and trajectories. In the second version more advanced characteristics that simulate spatial behaviour more accurately are included in both the urban and pedestrian models. This includes crowd attraction, agent need vector, agent to agent interaction, interpersonal distance, attractive object category and virtual attractive objects with dynamic level of attraction. The simulated characteristics are employed in the analysis system to differentiate between several classes of spatial behaviour.

The analysis system should be able to classify not only simulated noise free data but also real-world noisy data. To evaluate the proposed analysis system a pedestrian detection and tracking system was developed for extracting pedestrians' trajectories from Wheeler Place, Newcastle, Australia. The system includes three main parts: *i*) background detection, which is capable of upgrading the background image dynamically, *ii*) pedestrian detection, which uses a set of shape-based features in a classification technique to distinguish between pedestrian and non-pedestrian foreground objects, and *iii*) pedestrian tracking, which recognises the detected pedestrians in a stack of video frames using motion-based and depth-based features. The detection is carried out using a learning system that employs Support Vector Machines.

Support vector machines (SVMs) have become a popular approach to pattern classification, as they can deliver state-of-the-art performance on a wide variety of real-world classification problems. SVM is a maximum margin kernel-based classification technique. SVMs are a learning system that uses a hypothesis space of linear functions in high dimensional feature space. The kernel used in SVMs maps the data to the higher dimensional space. The kernel includes a distance measure that finds the similarities/dissimilarities between data objects. The most common kernel function that has been used in SVMs is the Radial Basis Function (Gaussian Function) with the Euclidean Distance (ED).

The kernel function used in the standard SVMs classification technique is required to satisfy Mercer's conditions, otherwise the existence of Reproducing Kernel Hilbert Space (RKHS) is not guaranteed and it is no longer clear what it is that is being optimised. ED can be only applied on data vectors with fixed length. The use of an elastic distance measure called Dynamic Time Warping (DTW) is a tempting solution for the analysis of input series with different lengths. However, it will be shown that the constructed kernel that used DTW is not always positive semi-definite, which violates the Mercer's theorem. To overcome this problem, a classification method is proposed. It is a new classification technique for sequential data analysis, where each data object is characterised by a series of numerical values that may have different lengths for different data objects.

Employing the new classification technique makes it possible to avoid segmenting data entries into fixed-length parts. The fixed-length data segments may fail to contain the information needed to properly describe samples in the input space. Therefore, in classifying trajectory data the accuracy of the classifier using fixed-length feature vectors is very dependent on the segmentation or feature selection algorithm. Instead of using some fixed-length segments (features) to describe the trajectory data, a new system that considers the entire trajectory for each user as a single input to the system is proposed. To achieve this, the proposed classification technique is employed to classify trajectories obtained from both real-world and simulated pedestrians.

The analysis method can classify behavioural characteristics extracted from the real-world or generated in the proposed simulation software. Although the classification accuracy using real-world data was lower than using simulated data, GDTW-P-SVMs significantly improved the error rate compared to other existing methods. The classification accuracy of GDTW-P-SVMs was compared with the other classification techniques, and GDTW-P-SVMs showed the highest accuracy using real-world and simulated data sets.

1.1 Motivation

Analysing how pedestrians' dynamic behaviour in space is influenced by environmental settings is an important component in the design of transportation facilities, pedestrian walkways, traffic intersections, markets, and other public spaces. The analysis is also an essential component required by a variety of applications. The areas of applications include automated surveillance, indoor and outdoor architectural design, advertisement and marketing, and emergency management. Due to a large number of potential applications, pedestrian behaviour analysis has become an active area in sociology research. However, analysing the spatial behaviour of pedestrians using an automated intelligent system has rarely been investigated. This is either because extracting such an amount of information from observers was a time consuming process, or an accurate classifier able to work with variable-length data objects was not available. Existing analysis approaches mainly suffer from applying a constraint to obtain fixed-length data vectors [94].

This thesis presents an intelligent method to model the impacts of environmental visual attractors on pedestrian spatial behaviour. The method is applied to generated as well as real-world data sets. The generated data sets are obtained by utilising the proposed simulation software. The simulation software is capable of modelling dynamic and static properties of spatial behaviours affected by personal and social interactions. The intelligent method is a maximum margin classifier with the ability to handle data objects with different lengths. The classifier makes it possible to use trajectories with different lengths in the input space. Using the whole sequence instead of some fixed-length segments of the trajectory data improves the accuracy of behaviour classification.

There are still many open questions about how the aesthetics of the environment interact with human pedestrians or users of space, and how it compares to other, more functional factors, such as path widths, the availability of open space, and the presence of obstacles or attractive objects. While behavioural data extraction and information retrieval from the data is the main focus of this thesis, data collection for specific applications could be a complex and time consuming task and it is out of the scope of this thesis.

1.2 Application Areas

The proposed approaches to analyse spatial behaviour are presented in two separate parts in this thesis. The first part is associated with behavioural data collection and the second part is related to the analysis of data that are extracted using the methods described in the first part. The application areas of this research are also split into two parts using the same criteria.

1.2.1 Behavioural Characteristics

Theile [178] has discussed the importance of virtual architecture in design and using the computer world as a suitable medium by which designers can convey the human side of design. Virtual pedestrians play an important role in analysing the effectiveness and improving the design of an architectural space. By using a simulated environment we can avoid the difficulties of extracting complex pedestrian behavioural characteristics. In addition, the analysis of simulated spatial behaviour provides an assessment of proposed architectural designs and helps to design the environment more effectively [178]. Simulating the behaviour of pedestrians in "normal" situations is also important in urban planning [97], land use [150], and traffic operations [27].

While simulated characteristics have long been used in the above mentioned applications, the analysis of extracted behavioural features using pedestrian detection and tracking systems has become popular in the last decade [59]. Pedestrian detection and tracking can be seen as a key enabling technology within the framework of a variety of intelligent systems. This technology is key to knowing who is where in a scene and what their actions have been. Accurate pedestrian detection and tracking is a prerequisite for the viability of a variety of computer vision applications, such as multimedia storage [95] automated security systems [82], public service applications [154], and multidisciplinary paradigms, for example ambient intelligence [191].

1.2.2 Spatial Behaviour Analysis

In the study of spatial behavior we are interested in finding the rules for spatial choice which, when applied to any unique distribution of spatial opportunities, are capable of generating spatial behavior patterns similar to those observed. Several studies have concentrated on the analysis of the relationship between the configurational characteristics of urban spaces and pedestrian spatial behaviour [189]. A constant pattern of movement, characterising urban spaces with the presence of pedestrians, would improve our architectural understanding of the space. This aids in analysing the design of an urban space from architectural and sociological point of view [93]. In addition, the literature on human cognition suggests that configurational aspects of built environments have significant consequences on pedestrians' spatial behaviour. Golledge and Stimson [64] have emphasised that the path or network structure used in everyday spatial behaviour is a critical feature of the image of a spatial environment. Others suggest that the spatial layout of the built environment influences the accuracy of cognitive representations of real-world spatial information [143].

Behaviour normality analysis is an active research topic in increasing the security of public places such as museums, parks, and cinemas [176]. The importance of fear of crime on affecting the use of urban spaces has been given special attention by many authors [117, 32]. In particular, authors in [117] showed that there is a correlation between the spatial features of layout and crime distribution patterns.

Crowds occur frequently, usually without serious problems. Occasionally venue inadequacies and deficient crowd management result in injuries and fatalities. Management of emergency response for both man-made and natural incidents has become a key research field. Effective crowd management requires accurate prediction of the impact of such incidents on the crowd as well as the environment. By analysing the behaviour of a group of pedestrians in simulated emergency situations, the responsible agencies would be able to evaluate different evacuation and damage control policies beforehand. This will allow the execution of the most effective crowd evacuation scheme during the actual emergency scenario [173].

The purpose of advertisement in public spaces is to attract the attention of people as they pass by. The behaviour of pedestrians in a public space can provide useful information regarding the attention paid to an attractive object, i.e. its visiting frequency. By analysing visiting frequencies for all attractive objects in different configurational scenarios, the best configuration with the highest number of attracted pedestrians can be obtained. This approach is applicable in advertisement and marketing where highest number of visitors is always desirable.

1.3 Challenges

1.3.1 Behavioural Characteristics Extraction

Selection of the behavioural characteristics that can reveal the desired differences between normal and attracted trajectories is the first challenge that has been encountered. Two main criteria should be considered when selecting the characteristics: *i*) the differences between normal and abnormal behaviour described by the characteristics should be detectable using an automatic learning system and *ii*) the characteristics should be extractable and measurable. For instance it is assumed that the physical characteristics of an environmental setting influence our attitudes and actions more than a biological or cultural trait [92]. Therefore the biological or cultural characteristics do not show a detectable difference between normal and abnormal spatial behaviours. In addition, socio-cultural variables are harder to measure and less obvious across behavioural settings.

To generate a set of behavioural characteristics a simulation software that models pedestrians' behaviour in urban spaces is proposed. Pedestrian flows are characterised by a significant degree of randomness, so that one could consider each individual's trip is unique. Simulating and analysing such a big data set of behaviours involves many ingredients, such as navigation and orientation, evaluation and decision making, variation of personal behaviour, and crowd attraction. Therefore, analytical approaches that use an average mathematical function to model such randomness would be too restricted as they apply the same function to all simulated individuals. On the other hand, pedestrians' movements are not restricted to cell sizes and therefore cellular automata cannot describe pedestrian movements for our purpose.

Extraction of the selected behavioural characteristics that describe pedestrians' spatial behaviour in the real-world is a challenging problem. This includes the detection of non-rigid body, different colours and shapes for different pedestrians, background detection and pedestrian occlusion.

1.3.2 Spatial Behaviour Analysis

The system for the automatic analysis of behavioral information classifies and collects statistics of human activities. The behavioural data may contain strong noise due to the collection method or environmental settings. The system therefore needs to distinguish small changes in noisy data. It also needs to be trained on normal behaviour using a small amount of data since data collection for this purpose is a time consuming and expensive process.

As individuals move at different speeds, the data inputs of pedestrian behaviour analysis in Wheeler Place (or within the model of Wheeler Place) will have different lengths. The analysis method should therefore be able to handle data objects with different lengths. Since we need to compare each journey with others, each data entry should contain the information for the entire period of travel. So the analysis method is required to accept each journey as one single input to reveal the differences/similarities between casual and attracted behaviours.

1.4 Aims and Objectives of This Thesis

This research investigated approaches and methods for analysing pedestrian spatial behaviour in urban spaces. Throughout the research, consideration was given to the purpose of the analysis, for example improving the architectural design to support effective use of public spaces and for modelling the impacts of different configurations on pedestrians' behaviour. The work is based on the following aims:

1. *Simulating pedestrian spatial behaviour*: To demonstrate how pedestrians' behavioural characteristics can depend on selected abstract features in urban space, a simulation software was developed. The simulation includes urban models, pedestrian models, and a multi-agent-based simulation method. The simulation is used to support architects and urban designers to better assess the impact of planned urban spaces and streetscapes on pedestrian spatial behaviour. It is also used to show that the changes in spatial behaviour due to the influence of attractive objects in urban spaces are detectable.

2. *Collecting pedestrian trajectories*: To show the impacts of attractive objects on pedestrian spatial behaviour in urban spaces, a pedestrian detection and tracking system was developed. The system includes methods to recognise pedestrians from non-pedestrian objects, and a tracking scheme to match pedestrians in a stack of frames. It is also employed to evaluate the simulated trajectories.

3. *Developing a trajectory-based classification method*: To analyse the collected and generated behavioural characteristics, an intelligent classifica-

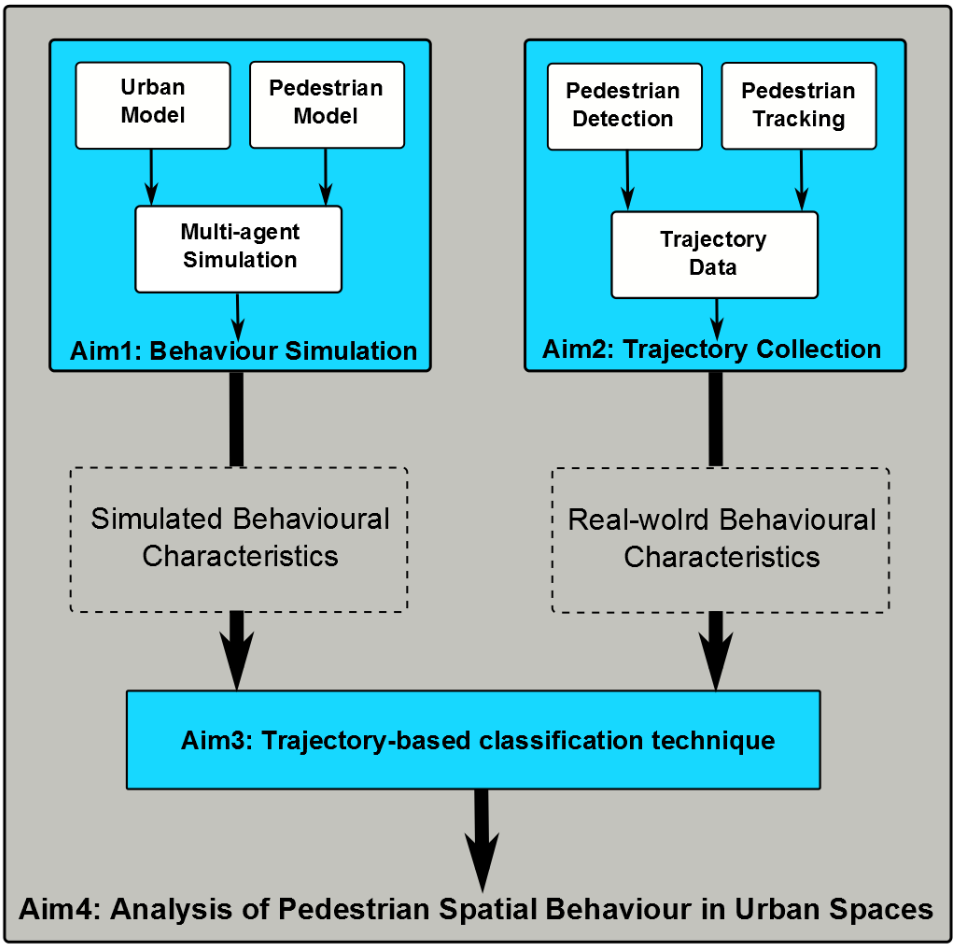

Figure 1.1: Relationship between the research aims.

tion technique is proposed. The technique needs to be capable of handling the extracted and generated trajectory data sets. The technique also needs to be robust and to accept input series with different lengths.

4. *Analysing the behavioural characteristics*: The main aim of this research is to develop a system to analyse pedestrian spatial behaviours in urban spaces. This aim is addressed iteratively by employing existing and proposed classification techniques on the simulated and collected data to distinguish between defined classes of behaviours.

Thus, this research aimed to propose, develop and discuss approaches for analysing the impacts of attractors in urban spaces on pedestrian spatial behaviour. In the course of this work data were collected on the actual behaviours that people demonstrated in an urban space. These data were analysed only to support further understanding of the approaches and methods. The relationship between the aims of this research is shown graphically in Figure 1.1. The main research contributions regarding to each aim will be discussed in Chapter 7.

1.5 Publications

The following publications have been written within the timescale of this PhD research:

Journals

- A. Jalalian, S. K. Chalup, and M. J. Ostwald, Architectural evaluation of simulated pedestrian spatial behaviour, Journal of Architectural Science Review, Vol 54, no. 2, pp 132-140, 2011
- A. Jalalian, and S. K. Chalup, GDTW-P-SVMs: Variable-length time series analysis using support vector machines, 2011. (under review, revised version submitted)
- A. Jalalian, S. K. Chalup, and M. J. Ostwald, Learning attention-driven activities using pedestrian trajectory analysis in video data, 2012. (under review)

- Aaron S. W. Wong, Stephan K. Chalup, Shashank Bhatia, Arash Jalalian, Jason Kulk, Steven Nicklin, and Michael J. Ostwald. Visual gaze analysis of robotic pedestrians moving in urban space. Architectural Science Review, 2012.

Conference Proceedings

- A. Jalalian, S. K. Chalup, and M. J. Ostwald, Analysis of pedestrian spatial behaviour using GDTW-P-SVMs, International Joint Conference on Neural Networks (IJCNN 2012), IEEE Computer Society, 2012.

- A. Jalalian, S. K. Chalup, and M. J. Ostwald, Simulating pedestrian flow dynamics for evaluating the design of urban and architectural space, in 44th Annual Conference of the Architectural Science Association, C. Murphy, S. J. Wake, D. Turner, G. McConchie, and D. Rhodes, (Eds.) Auckland, New Zealand, 2010.

- A. Jalalian, S. K. Chalup, and M. J. Ostwald, Agent-agent interaction as a component of agent-environment interaction in the modelling and analysis of pedestrian visual behaviour, in The 16th International Conference of the Association for Computer- Aided Architectural Design Research in Asia, C. M. Herr, N. Gu, S. Roudavski, and M. A. Schnabel, (Eds.), Newcastle, Australia, 2011, pp. 555-564.

- A. Jalalian, S. K. Chalup, and M. J. Ostwald, Intelligent evaluation of urban streetscape designs by analysing pedestrian body dynamics, in The Third International Workshop on Advanced Computational Intelligence, Washington, DC, USA: IEEE Computer Society, 2010, pp. 442-447.

- A. S. W. Wong, S. K. Chalup, S. Bhatia, A. Jalalian, J. Kulk, and M. J. Ostwald, Humanoid robots for modelling and analysing visual gaze dynamics of pedestrians moving in urban space, in 45th Annual Conference of the Architectural Science Association, Sydney, Australia, 2011.

1.6 Thesis Overview

- **Chapter 2** depicts the proposed models to simulate pedestrians' spatial behaviour in urban spaces. The models provide a set of behavioural characteristics that can be used to analyse pedestrians' behaviour in urban designs. The initial models are described with and without considering pedestrians' visual attention. Virtual attractive objects with variable levels of attraction along with the "agent's need vector" and "object category vector" are introduced in this chapter. A comparison between goal-driven and stimulus-driven attentions is given and the proposed approach to model these spatial behaviours is represented. The results of using the simulation are discussed and compared with and without considering the impacts of visual attention.

- **Chapter 3** outlines the proposed method to recognise and track pedestrians in video data captured at Wheeler Place, Australia. The trajectory data are then used to reveal the impact of visual attractors on pedestrians' spatial behaviour. A literature review of previous works in pedestrian detection and tracking is given and shortcomings and advantages of using each method are discussed. Different stages of the proposed system such as background detection, shape-based, motion-based and depth-based feature extraction, feature classification, and feature extraction for tracking are presented. The results of applying the system to Wheeler Place video data are discussed in this chapter.

- **Chapter 4** reviews a maximum margin kernel-based classification technique known as Support Vector Machines (SVMs). It provides an introduction on the main concepts of kernel-based learning and discusses the SVMs as an example classifier. The usage of common kernels in SVMs and their features are discussed. The Potential Support Vector Machines (P-SVMs) and their advantages over SVMs with conventional Gaussian Kernel are highlighted.

- **Chapter 5** presents the proposed technique for sequential data analysis where each data object is characterised by a series of numerical values that may have different lengths for different data objects. Pairwise classification is discussed as a method for multi-class classification problems. The usage of Dynamic Time Warping in Gaussian kernel in SVMs is argued. The proposed classification technique is compared with several other classification techniques that are capable of handling data objects with different lengths. The advantages of the use of the proposed method over using DTW-SVMs are highlighted.

- **Chapter 6** presents the proposed spatial behaviour analysis system. The system is employed to classify simulated as well as real-world data sets. Different classes of behaviours that are detectable using each of the data sets are introduced. The classification techniques that are capable of handling data objects with different lengths are employed to analyse pedestrians' behaviour. Classification results are compared for different classifiers as well as simulated and real-world data.

- **Chapter 7** concludes the research and recommendations are made for future work into approaches for analysing pedestrians' behaviour in urban spaces.

1.7 Summary

This chapter has laid the foundations for the thesis by introducing the background and origins of the research into pedestrian behaviour analysis approaches for investigating the impacts of visual attractors on spatial behaviours in urban spaces. The aims and objectives of the research have been described. The main research contribution will be discussed in Chapter 7. The structure of the various studies and investigations within the thesis has been discussed. This chapter has prepared the reader to follow the evolution of the research as it investigated approaches for analysing pedestrians' behaviour in urban spaces.

The next chapter will describe the proposed methods to simulate pedestrians' behavioural characteristics. It will demonstrate the proposed software that will be employed to simulate pedestrians' reactions to modelled visual attractors in an urban space. The generated behavioural data will be utilised in the analysis system to distinguish between different classes of behaviours.

Part I

Human Behavioural Characteristics

Chapter 2

Human spatial behaviour simulation

The content of this chapter has been published in [91, 92, 93].

Contents

The analysis of pedestrian flow dynamics in indoor and outdoor areas is an important facet of architectural and urban design. While a range of software programs have been developed in recent years to model idealised pedestrian flow between entry and exit points in a plan, such software typically neglects to consider the impact of "attractors" and that of the human gaze. In public places, including both external environments (streets and plazas) and internal spaces (malls and museums), attractors, like billboards or display stands, distract pedestrians from following a direct path towards their destinations. While there are a range of possible names for these visual distractions, in the present study they will be referred as attractors. It can further be assumed that these objects have different levels of attraction; some will attract a pedestrian's gaze, while others will completely interrupt their passage. An object with a low level of attraction, such as a piece of writing on the wall, would typically not attract pedestrians who walk fast. However, by placing objects with higher levels of attraction into the scene it is possible to reduce the speed of pedestrians, making them susceptible to being attracted to the objects with lower levels of attraction. The importance of this type of analysis is that it can be a useful predictive and analytical tool not only in architecture and urban design, but also in crowd management, transport facilities management, crime prevention, disaster planning, marketing, and epidemiology. Thus the ability to predict the response of a pedestrian in an urban area to his or her surroundings is important in estimating the effects of changes in the built environment.

2.1 Previous Works

2.1.1 Analytical Models

The reality is that, contrary to most software simulations, pedestrians do not always follow a simple path along the line connecting origin to destination. Furthermore, in contrast to vehicular flows, which circulate along fixed corridors of the road environment and are subject to specific traffic rules, pedestrian flows are characterised by a significant degree of randomness, so that one could consider that each individual's trip is unique [199]. Simulating and analysing such a big data-set of behaviours calls on an advanced intellectual method involving many ingredients, such as navigation and orientation, evaluation and decision making, variation of personal behaviour, and crowding. Therefore, computational tools are crucial to map this randomness into a real-time model that simulates pedestrian behaviour. Regression models [142], gravity model formulations [138], doubly-constrained models [70], generic coupled differential equations [63], and discrete choice models [4] are all examples of applied analytical computational models in pedestrian behaviour simulations. In analytical models, changes in pedestrian behaviour are expressed as a mathematical function that controls the average pedestrian movement. Since the analytical models apply the same rules to all simulated individuals and perform a simulation based on average characteristics, they are restricted in simulating pedestrian behaviour that in reality would vary much more depending on different plans, personal behavioural characteristics, and social preferences.

2.1.2 Cellular-Automata

Cellular automata (CA) is a discrete model employed for modeling complex phenomena in many research areas, including statistical physics, computer networks, sociology, architectural design, and fluid dynamics [33]. It is named after the principle of automata (entities) occupying cells according to localised neighborhood rules of occupancy [18].

One of the important usages of CA is in modelling traffic flow dynamics. It also has been used in pedestrian modelling because pedestrians are flexible and intelligent in changing directions and their movements have a large degree of freedom. Blue and Adler have applied Cellular Automata (CA) micro-simulation to model pedestrian flows and demonstrated that these models produce acceptable fundamental flow patterns [18]. Authors in [203] describe a method that uses simple CA local rules describing the behavior of each automaton to create an approximation of actual individual behavior.

The proposed methods that employed CA for pedestrian dynamics are based on the following approaches:

- The models in this group can be considered as generalizations of the Biham- Middleton-Levine model for city traffic [16], and named as biased random walker model [187, 128, 134].
- Models in this group have employed a floor field that modifies the transition rates to neighbor cells, inspired from the process of chemotaxis as used by some insects [23, 111].

Although cellular automata models are computationally fast and therefore suitable for large scale computer simulation [12], the accuracy of the models depends on the cell size used in CA. For instance all the proposed methods that used CA to model pedestrian flow dynamics have to occupy individuals in different cells. This implicitly restricts movement of the simulated agents to the centre of cells.

2.1.3 Multi-Agent Based Simulations

One of the useful ways to model pedestrian behaviour and overcome the shortcomings of the analytical models is to develop an agent-based simulation. An agent-based simulation is a computational model for modelling the behaviour of autonomous agents. In agent-based modelling (ABM), a system is modelled as a collection of autonomous decision-making entities called agents [19]. For an architectural or urban simulation to be useful for designers, the agents should mimic real pedestrians and they should be of a number of different types with different movement abilities. It also should be possible to change their characteristics and numbers to fit the circumstances being examined.

Considerable research has been done on the topic of multi-agent-based simulation systems [19, 199, 198]. For instance researchers in Catania [24] performed an agent-based simulation of people visiting and evacuating Planimetry of Castello Ursino in emergency situations. They modelled the environment in 2D using the NetLogo platform, which is a multi-agent simulation software [190]. Francesca Camillen [24] has confirmed the usefulness and the effectiveness of agent-based simulations in the design and analysis of complex social systems. Such simulations are used to support more traditional strategies already available to the engineers. For example Gipps and Marksjo [62] have presented a macro-simulation cellular approach to model interactions between pedestrians, which is intended for use in graphical computer simulation.

In agent-based pedestrian simulation, cellular automata and intelligent agents have had a huge growth in adoption in recent years [63, 24, 177]. In another example, the team of Kazuhiro Yamamoto [197] have proposed the use of real coded cellular automata as a new numerical model for pedestrian dynamics. They have obtained the critical number of people beyond which clogging appears at the exits of rooms. In all cellular approaches, researchers have assumed that only one agent can be placed in each cell. Cellular approaches ease the way for simulating pedestrians; however they also reduce the accuracy of pedestrian models in urban areas. While agent movements in cellular models are limited by cell sizes, in the real-word pedestrians are not limited to follow cells and can choose their next step in any direction.

2.1.4 Pedestrian Simulation in Urban Spaces

Placing virtual pedestrians in architectural space aids in analysing the effectiveness of the space and improving the design of the space. Thiele [178] has discussed the importance of virtual architecture in design and using the computer world as a suitable medium by which designers can convey the human side of design. To demonstrate the idea of using simple intelligence (SI) instead of artificial intelligence (AI), an outline of Cura, a virtual presenter, was presented by the author. Simulating the behaviour of pedestrians in "normal" situations is also important in urban planning [97], land use [150], and traffic operations [27]. By analysing this behaviour in different spaces the usage of those spaces can be assessed.

Behaviour normality analysis is an active research topic in increasing the security of public places such as museums, parks, and cinemas. One of the most popular methods used to detect abnormal behaviours is comparing the behavioural characteristics with the behaviours most frequently observed in the past. Utilising advanced technology increases our computational abilities to employ complex algorithms for comparing pedestrians' behavioural characteristics. For instance [176] proposed a computational method to learn motion patterns and detect anomalies by human trajectory analysis. They employed HMMs (Hidden Markov Models) to model time-series features of human positions. Using a similarity matrix of HMM mutual distances and k-means clustering they acquired features of human motion patterns.

2.2 Pedestrian Spatial Behaviour Simulation

A multi-agent system with the characteristic dynamics of a crowd of moving pedestrians in a section of an abstract urban 2-dimensional environment is proposed. A similar approach to 2D mapping of environments has previously been proposed [118] and it has also been used in robotics for the exploration of an office-like indoor environment using a multi robot team [116]. The proposed approach is based on artificially generated abstract point-like agents. In the proposed agent-based pedestrian behaviour simulation, each agent individually assesses its current situation and makes decisions on the basis of its current state and a set of rules.

In the simulation, agents may carry out various behaviours resembling real world pedestrian behaviour in an urban streetscape. The simulation generates a set of behavioural characteristics - for example walking, running, standing, getting attracted to an obstacle, and associated changes of the gaze direction. The developed analyser software (described in section 5) evaluates the simulated behaviours in order to identify the impacts of different walking environments on pedestrian behaviours. The analyser uses a combination of machine learning classifiers and statistical algorithms to allow it to learn past normal behaviours and distinguish between normal and abnormal behaviours in future data.

Figure 2.1 shows the relationship between different blocks of the pedestrian analysis system. The system generates pedestrian behavioural data using the simulation software. A classifier then is employed to distinguish between different spatial behaviour stored in the data. A simple visualisation scheme is also developed to demonstrate simulated pedestrians (also known as agents) and the classification results.

To demonstrate the software developed by the author, a simulation using

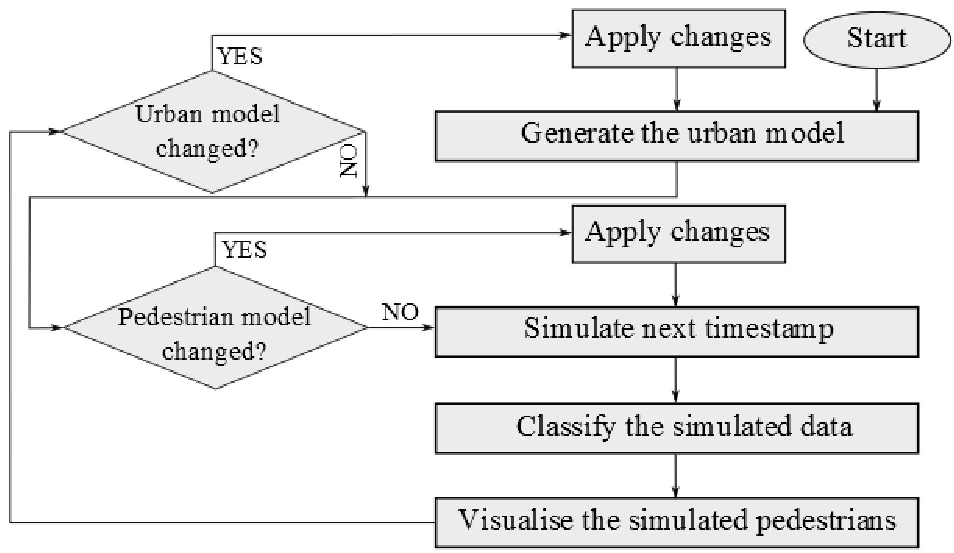

Figure 2.1: Architecture of the pedestrian spatial behaviour analysis system.

the plan of an urban space called Wheeler Place is presented. This space, located between the two busiest streets in Newcastle (Australia) features a constellation of attractive objects, offers an excellent test environment for analysing different pedestrian behaviours (Figure 2.2). The space is next to Newcastle's Civic Theatre and City Council Chamber and these two busy buildings have made this area one of the Newcastle's most crowded public spaces. There are also several places of different levels of attraction in this area itself including the City of Newcastle Information Centre and Climate Meter, Juicy Beans Restaurant and Internet Cafe, a big public art work, the Civic Theatre and Civic Theatre Restaurant. In combination these contribute to offer several destinations for pedestrians who can be differentiated in behavioural characteristics, physical characteristics, and personal preferences. As shown in Figure 2.5 the vicinity's five entrances (black squares) and exits (red exit signs) are restricted to several distinguishable points and pedestrians mainly choose one of these points to enter and exit Wheeler Place.

2.2.1 Pedestrian Behavioural Model

Figure 2.3 demonstrates the agent model in the proposed multi-agent-based pedestrian simulation. In the real world individuals have several characteristics representing their spatial behaviours in an urban streetscape, such as speed of movement, head direction, and location. Table 2.1 describes the parameters that were used for modelling a pedestrian's behaviour.

In the simulation each agent is initialised using the described parameters. Some parameters are generated randomly (such as sight, starting point, destination, speed category and sight category) and others are calculated based on those randomly generated (such as location, speed, and angle). In the initialisation step $end^j_{(x,y)}$ and $start^i_{(x,y)}$ are generated based on random values, and one start-point and one end-point are selected among the defined start-points and end-points in the proposed urban model.

In the urban model we have different categories of behaviours, which are defined by the S_{cat} (speed category) and V_{cat} (sight category). In each category we have varieties of behaviours defined by a random function. To separate

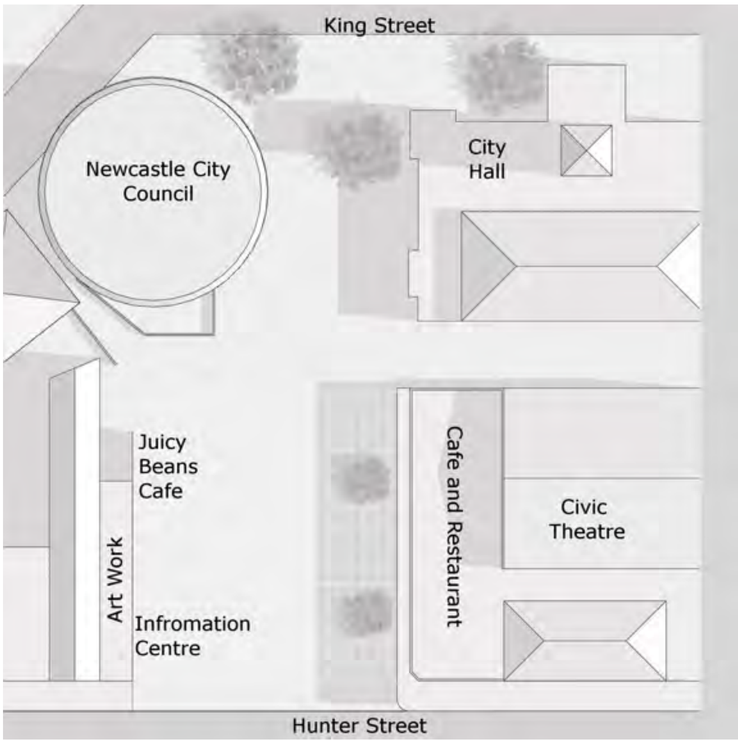

Figure 2.2: Aerial one-point perspective plan of Wheeler Place. The locations of several places of different levels of attraction including the City of Newcastle Information Centre and Climate Meter, Juicy Beans Restaurant and Internet Cafe, a big public art work, the Civic Theatre and Civic Theatre Restaurant are shown.

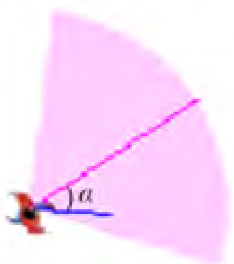

Figure 2.3: Simulated pedestrian; UFOV (pink area), speed vector (short line), gaze direction (long line), and the angle between them (α).

Parameter	**Description**	**Unit,Symbol**
Location	Location of an agent in the urban plan at time t	pixel, (x_t, y_t)
Speed	Speed of an agent at time t, determined by speed category and location of the agent relative to its surroundings.	pixel, $(\frac{dx}{dt}, \frac{dy}{dt})$ or β, $\|s\|$
Angle	The angle between the head direction and speed vector at time t	degree, α_t
Field of view	Standard pedestrian's *functional* or *useful* field of view (UFOV)which is assumed to be 95°[10].	degree, $(UFOV)$
Sight	Random number that represents an agent's visual ability. It is associated with the agent's "sight category".	pixel, V
Starting point	The first location of the agent in the plan.	pixel, $start^i_{(x,y)}$
Destination	Final destination where the agent is going to leave the urban area.	pixel, $end^j_{(x,y)}$
Speed category	5 different speed categories to model different pedestrian behaviours subject to speed.	S_{cat}
Sight category	5 different sight categories to model different pedestrian behaviours subject to vision capability.	V_{cat}

Table 2.1: Pedestrian model parameters

the behaviour of each category from others, in each category the range of randomness is restricted by defined maximum and minimum values. After initialising S_{cat} and V_{cat} for each agent, $\|s\|$ (speed) and V (sight) are calculated based on S_{cat} and V_{cat} respectively.

In the simulation, when there are no stimuli to attract pedestrians' attention the angle α, controlling the gaze direction, remains a small random value. This is because under normal conditions, pedestrians would typically look straight ahead such that the gaze vector can be assumed to be almost parallel to the speed vector. In all other conditions, α will be changed accordingly. The aim of each agent is to reach $end^j_{(x,y)}$, and therefore the direction

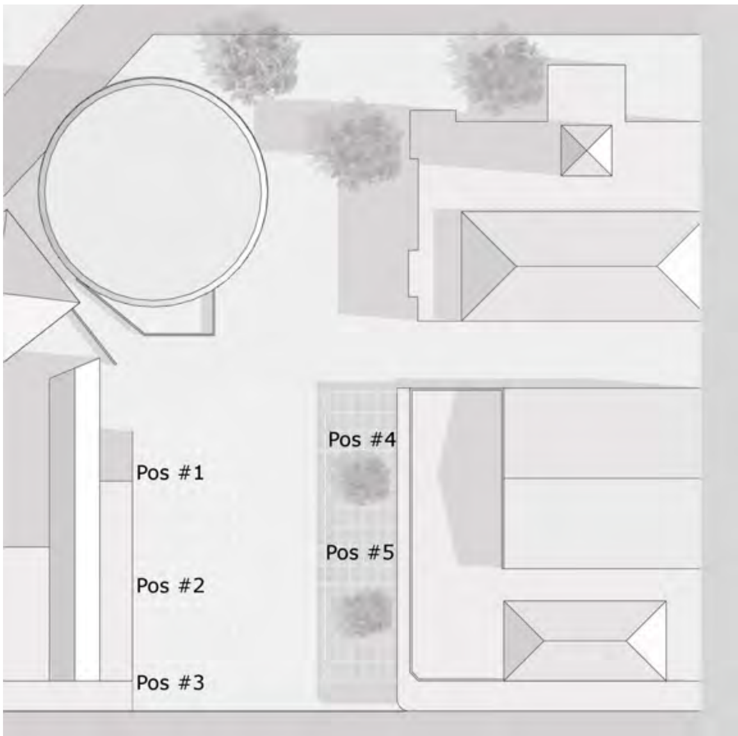

Figure 2.4: Five possible positions in Wheeler Place where the attractors could be placed.

of movement, β, is determined by a random value and the direction of an imaginary line passing through $end^j_{(x,y)}$ and $start^i_{(x,y)}$.

2.2.2 Urban Model

In the proposed simulation the urban model was obtained by using a scan of a plan of Wheeler Place. As shown in Figure 2.5 the considered area has five entrances and exits ($start^i_{(x,y)}, end^j_{(x,y)} : i, j = 1...5$). Each attractor is surrounded by a circle that indicates its assumed level of attraction. The radius corresponds to the attraction level. In the reported simulation experiments five positions (Pos#1, Pos#2, ..., Pos#5) were selected, where attractors can be placed (see Figure 2.4).

During simulation, once an agent's $UFOV$ [1] overlaps with the attraction area of an object the agent will move towards the object with the maximum allowed speed in its speed category S_{cat}. Therefore, objects with higher levels of attraction can attract agents with the same V_{cat} from further distances. On the other hand agents with bigger V can be attracted by further objects with the same level of attraction.

In the urban model the following terms are used:

- *image plan*: An aerial one-point perspective plan of the area
- *start points*: The locations of the entrances
- *end points*: The locations of the exit points
- *object location*: The location of the attractive object on the image plan ($obj_{(x,y)}$)
- *object visit counter*: The number of agents that have been attracted to the object
- *object attraction level*: The level of attraction for an attractive object

It is also assumed that boundaries of pathways and other obstacles such as trees are impermeable.

2.2.3 Spatial Behaviour Simulation

In the simulation each agent assesses its behaviour according to several decision making rules defined by the pedestrian model and the urban model. In this section, two major types of agents are defined; "attracted agents" and "normal agents". While the former is attracted to an attractive object, the latter has not seen any attractive object and just moves along the "normal direction of movement" towards its destination. Normal directions of movement are defined by the lines between start points and destinations. By selecting a start point and a destination, each agent has selected one normal direction of movement in the initialisation stage.

[1] Useful Field Of View

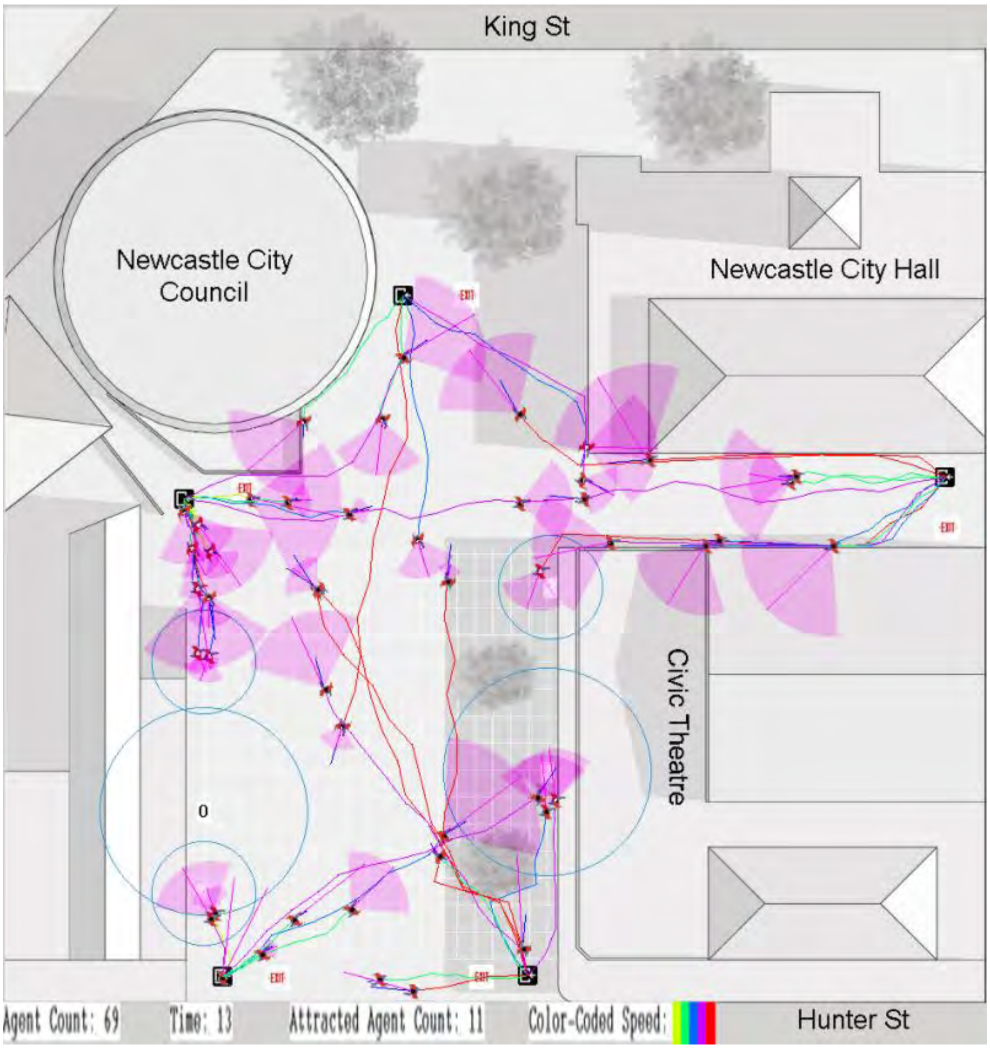

Figure 2.5: Wheeler Place simulation in Scenario 1. Representation of: Trajectories (color-coded tracks), Start Points (black squares), Exit Points (red exit signs), Color-coded Speed (each color represents a speed category), and Attractors (blue circles).

Agents may carry out various behaviours roughly resembling real world pedestrian behaviour in an urban streetscape, such as walking, running, standing, or becoming attracted to an obstacle. Regarding to pedestrian's reaction to the attractors in an urban area, pedestrians may have four different spatial behaviours:

1. *becoming attracted*: when they can see the object for the first time, and they believe the object is attractive for them,
2. *not becoming attracted*: when they can see the object for the first time, and they believe the object is not interesting enough for them to move closer to it,
3. *visiting*: stopping while they are studying the object, and
4. *normal*: while they cannot see any attractive object.

In the simulation various behaviours are defined by changing an agent's characteristic parameters. To obtain the next location for each agent, the direction of movement, $\beta_{(t+1)}$, is calculated as follows:

$$\beta_{t+1} = \begin{cases} tan^{-1}(\frac{end_y^i - y_t}{end_x^j - x_t}) + rnd \times \beta_{osc} & : normal/not\ becoming\ attracted \\ \beta_t & : visiting \\ tan^{-1}(\frac{obj_y^i - y_t}{obj_x^j - x_t}) + rnd \times \beta_{osc} & : becoming\ attracted \end{cases} \tag{2.1}$$

Here rnd is a random value in $[0, 1]$ at time t. Since pedestrians do not walk exactly in a straight line, this behaviour was simulated by using $\beta_{osc} = 20°$, which indicates the oscillation range for β.

When the $UFOV$ for an agent overlaps with the circle indicating an object's level of attraction, the simulated agent will be notified of the attractive object. In this case, the agent uses the *becoming attracted* equation to calculate $\beta_{(t+1)}$. By using this equation the agent moves towards the object instead of the end point. When the agent comes close enough to the object, it will spend some time standing at the object and studying it. During this period (*visiting*), the direction of movement remains constant. At all other times the agent shows normal spatial behaviour and uses the *normal* equation to calculate $\beta_{(t+1)}$.

In a real world scenario, while pedestrians are moving they do not always look straight ahead. To simulate this behaviour, α_{osc} is defined as the oscillation range for the relative gaze angle α_t in normal behaviour, which is calculated as follows:

$$\alpha = (rnd \times 2 \times \alpha_{osc}) - \alpha_{osc} \tag{2.2}$$

In *not becoming* and *becoming attracted* behaviours the agent's gaze vector points to the attractive object.

To obtain the next location for each agent the speed of movement, $\|s\|$, needs to be calculated. In the simulation five different speed categories were used for each agent [2]. Each category has a maximum speed and a minimum speed. Agents choose their speed category when they are initialised and this category remains constant during the agent's lifespan. $\|s\|$ is obtained as follows:

$$\|s\| = \begin{cases} \frac{1}{5}s_{max}(rnd + s_{cat} - 1) & : normal/not\ becoming\ attracted \\ 0 & : visiting \\ \frac{1}{5}rnd(s_{max}s_{cat}) & : becoming\ attracted \end{cases} \tag{2.3}$$

Where s_{max} is the maximum possible speed, and rnd is a random value in $[0, 1]$. The $\frac{1}{5}$ coefficient depends on the number of speed categories. Since we assumed that agents travel with five different speed categories, then $\frac{1}{5}$ is used.

By applying $\beta_{(t+1)}$, $\|s\|$ and (x_t, y_t) in the following equations and solving them for $(x_{(t+1)}, y_{(t+1)})$, we obtain the next location for each agent:

$$\begin{cases} \|s\|^2 = (x_{t+1} - x_t)^2 + (y_{t+1} - y_t)^2 \\ \beta_{t+1} = tan^{-1}(\frac{x_{t+1}-x_t}{y_{t+1}-y_t}) \end{cases} \tag{2.4}$$

Since we assumed the boundaries of pathways and other obstacles such as trees to be impermeable, an algorithm to extract the boundaries and the obstacles from the image plan was developed using image processing techniques.

[2] Although human walking speed can vary greatly depending on factors such as height, weight, age, terrain, surface, load, culture, effort, gender and fitness, it can be categorized into five distinguishable ranges in urban spaces [22].

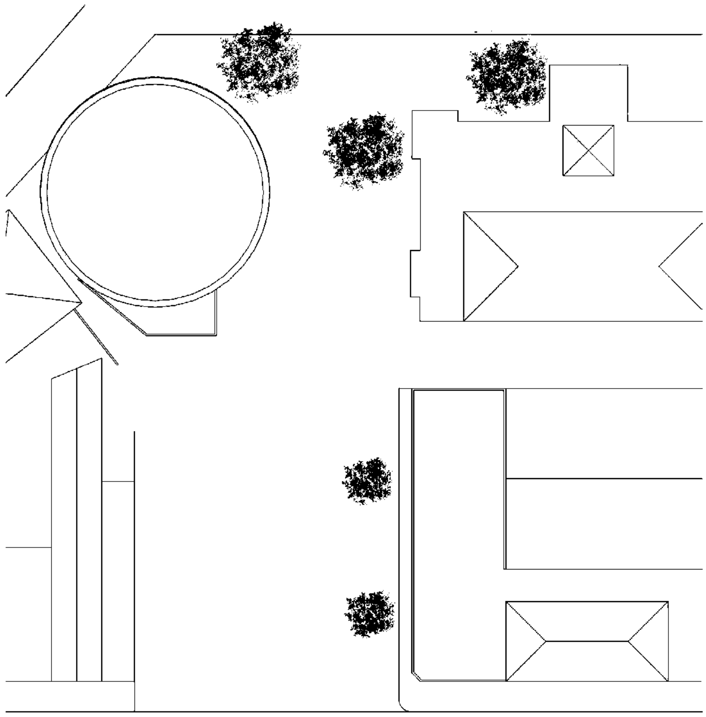

Figure 2.6: Extracted boundaries from an aerial one-point perspective plan of Wheeler Place.

Otsu's method, which chooses the threshold to minimize the intraclass variance of the black and white pixels, was applied to extract objects from the image plan [146]. The extracted boundaries (as shown in Figure 2.6) and obstacle information are then used to validate the obtained next location for each agent $(x_{(t+1)}, y_{(t+1)})$.

2.3 Visual Attention Modelling

Environmental perspective is an extremely important viewpoint of human spatial behaviour analysis. This view assumes that the physical characteristics of an environmental setting influence our attitudes and actions more than a biological or cultural traits. Spatial behaviour modelling focuses on the visible, *static* and *dynamic* properties of the physical environment as important factors, since socio-cultural variables are harder to measure and less obvious across behavioural settings. Analysing this behaviour in indoor and outdoor areas is an important facet of architectural and urban design.

The simulation of pedestrian behaviour allows us to investigate spatial visual behaviour without the difficulties of real-world data extraction. Planning pedestrian environments requires assumptions about how real pedestrians will respond to characteristics of the planned environment. Placing virtual pedestrians in an architectural space simulation significantly aids in making the assumptions more real [178].

Representation of the physical space plays a central role in modelling pedestrian spatial behaviour. In *public* places, including external environments (streets and plazas) and internal spaces (malls and museums), *attractors*, like billboards or display stands or even a group of people, can distract pedestrians from following a direct path towards their destinations. In urban spaces the attractors have different conspicuity areas [200]. For instance a high-rise tower in the centre of a town may attract more attention from a greater distance than a poster on a nearby wall. The variety of conspicuities were taken into account by defining different *levels of attraction* for attractors [90, 91, 93].

In this section an innovative approach is proposed to equip simulated objects with *dynamic* levels of attraction [92]. Pedestrian-environment interactions in parallel with pedestrian-pedestrian interactions have significant impacts on pedestrian spatial behaviour and walking patterns [9]. Since vocal interactions are harder to measure and depend on many geometrical and cultural variables, we focus on the visual part of pedestrian-pedestrian interactions. We investigate the dynamics of human head pose and eye gaze behaviours, which can provide significant insight into the context of a spatial behaviour. It has been shown that gaze direction and eye contact are essential features of group communication, as they help to elicit behavioural characteristics from others [44, 5, 106, 11].

Sudden changes in gaze direction may or may not be linked with attractors. A substantial amount of research in psychology has examined whether such behaviour is caused by internal stimuli (such as changing the route) or by external stimuli [87, 152, 137, 158]. While attractors take up a significant portion of visual external stimuli, other people's gaze direction could be considered as another important category of external stimuli to attract a pedestrian's gaze [200]. As the number of people looking at an attractive object (attractor) [3] increases, the object can attract more attention. The presented system incorporates the proposed pedestrian simulation system [93].

The system described in this section has been designed to support pedestrian-pedestrian visual interaction, and models the impacts of eye interactions in changing pedestrian spatial behaviour. The purpose of the simulation is to model and analyse this spatial behaviour in order to have a better assessment of a planned urban design and its impacts on pedestrian behaviour.

2.3.1 Attention and Eye Movement

The fact that eye movements are linked by attention does not mean that these two systems are completely mutually dependent. Authors in [77] confirm that observers can direct their visual attention to different areas of visual space even while the eyes remain fixed. Thus the relationship between attention and eye movements is one of partial interdependence. Attention is free to move independently of the eyes, but eye movements require visual attention to precede them to their goal [80].

While pedestrians are walking in an urban space their eyes scan their surroundings using saccadic eye movements. Saccadic eye movements occur about 3-4 times per second [14]. The eyes are essentially blind during these movements and the information from the scene is acquired only at fixation points. Authors in [201] pointed out that the location and sequence of saccades is not completely random. Eye movements from one fixation point to another are fast enough that in the simulation we can assume that all the objects in a certain range are scanned during a timestamp (4 seconds). The range of this scan varies from one person to another [112] and it mainly depends on the location that the pedestrian is looking at while walking.

This is modelled by defining different effective ranges for *useful field of view* (UFOV) [10]. In the initial stage of the simulation a sight category is assigned

[3]Sometimes the term "attractive object" is used instead of "attractor" to emphasise that the attractor is an object in an urban space.

to each agent. The *sight category* is a random number that limits the minimum and maximum range of UFOV. The number of sight categories can be defined and changed by the user in the user interface of the analyser software.

2.3.2 Agent's Need and Object Category Vectors

To model pedestrians' visual attention and the visual impacts of attractors on pedestrians' behaviour, "agent's need vector" and "object category vector" are introduced here. By using these two vectors we are able to model different reactions from different agents for the same attractor.

The agent's need vector (ANV) corresponds to the agent's aim of walking. Each element of the ANV reflects a category of need for the agent. The value assigned to each element expresses how desperate the agent is to satisfy that particular type of need. For example if an agent is simulating a pedestrian who is window shopping while he is hungry, the two elements of ANV that correspond to shopping and food will have higher values than other elements in ANV for that agent.

On the other hand, the object category vector describes the types of attraction attributed to an attractive object or what types of requests could be fulfilled by that attractive object. Each element of an object category vector expresses how much the object is able to satisfy an agent's need. For example if an attractor is modelling a gift shop that has a drinks vending machine, the two elements of the object category vector that correspond to shopping and food will have higher values than other elements in the vector for that attractor.

2.3.3 Goal-Driven vs. Stimulus-Driven Attention

According to [200], visual attention is either goal-driven or stimulus-driven. Attention is said to be goal-driven when it is controlled by an observer's intention and needs. For instance if the observer is hungry and looking for food, automatically all the food shops will attract his attention. In the simulation, pedestrians' goal-driven attentions are modelled by the agent need vector (ANV) and the *object category vector*. These vectors are assigned to the agents and attractors at the initial stage.

In contrast, stimulus-driven attention is controlled by visual attributes of the object that are not necessarily relevant to the observer's perceptual goal. In the previous example, if all the food shops tended to be fast food restaurants, then a single Persian restaurant among them would capture the observer's attention automatically. Stimulus-driven attention is modelled by the *object level of attraction*. A bigger value for the object level of attraction means the object has a stronger impact on an agent's attention.

2.3.4 Virtual Attractive Objects

The social group is a fundamental and universal feature of human social life. Group formation is the expansion of bonds of interpersonal attraction due to the existence of common desires among individuals [81]. Having the same goal-driven attentions could be considered as one of the major determinants of group formation in urban environments [26]. In the simulation we assume that agents who are visiting the same attractive object can form a group.

Group size is an important variable for capturing attention: Large groups are able to draw more attention than small groups. In order to model how *group formation* and *group size* impact on pedestrian spatial behaviour, we consider a group of people as an attractive object with a *dynamic* attraction level and call it a *virtual attractive object* Figure 2.7. A virtual attractive object (VAO) has the same feature set as an *actual attractive object*. However, in the case of VAOs, the location and level of attraction vary according to the location and number of agents in the group.

As agents are walking in the scene they may get attracted to a selection of attractive objects (virtual or actual). The selection depends on agent needs, agent location, sight category, UFOV, object location, object category, and crowd locations. Agents consider attractive objects as temporary destinations. Once they finish visiting the attractive objects they will continue on their way towards exit points.

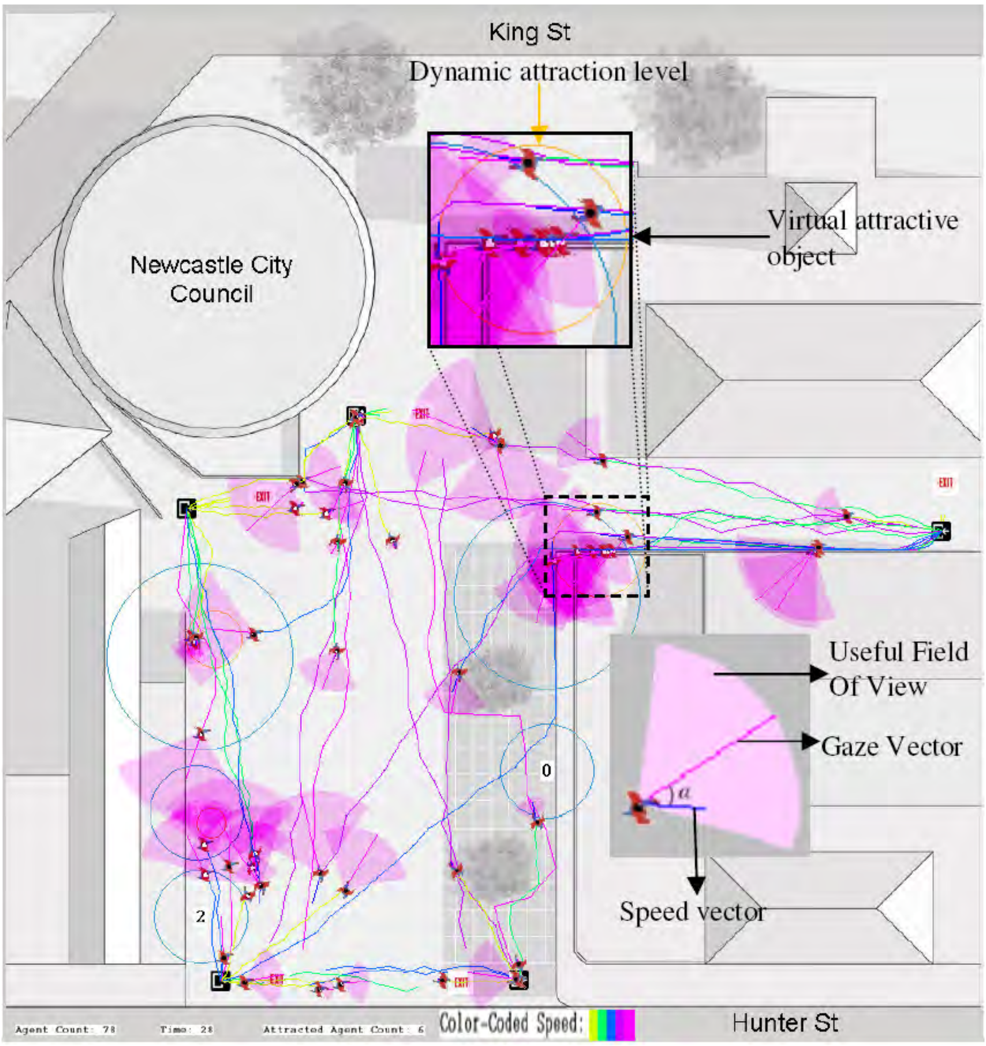

Figure 2.7: Pedestrian spatial behaviour simulation on the Wheeler Place Plan. Levels of attraction for attractors are shown with blue circles, and dynamic attraction levels for virtual attractive objects are shown with orange circles.

2.3.5 Object Selection Process

Attractive object selection starts by calculating the distance of the agent's UFOV and the level of attraction of all the attractive objects that the agent can see. Among the objects, the one that minimises this distance is the *best object*. We assume that actual attractive objects have higher priority than virtual attractive objects. Therefore the search for finding the best object is done over the real objects first, then over virtual objects.

Once the best object is found, the *Euclidean distance* between the agent need vector and object category vector is calculated. If this distance is less than a threshold it means that the agent likes/needs the object and its state will be changed to *becoming attracted*, otherwise it will be changed to *not becoming attracted*. Agents show the same behaviour if the selected object is a VAO, but instead of studying the object it will join the group and this will increase the level of attraction of the VAO.

2.4 Experimental Results

The results are reported with and without considering the impact of the "crowd attraction" in two separate subsections.

2.4.1 Results without Crowd Attraction Impacts

Although human walking speed can vary greatly depending on factors such as height, weight, age, terrain, surface, load, culture, effort, gender and fitness, it can be categorized into five distinguishable ranges in urban areas [22]. Figure 2.10 shows five different speed categories and their typical random characteristics for five agents in the proposed simulation. As shown in this figure, there is a considerable difference in $\|s\|$ among different speed categories. Each agent moves with a random speed within the boundaries of its speed category. The use of different categories of speed and sight simulates different spatial behavioural characteristics.

The experiment shown in Figure 2.8 displays the simulated agents' trajectories. In this example, the five speed categories were used and distinguished by *colour-coding* in the *Hue Saturation Value* (HSV) colour system. From left to right the colour bar at the bottom of Figure 2.8 represents speed categories with higher speed. In this figure each circle represents an attractive object. The number of agents attracted to each object is shown in the centre of the circles. The number of agents that have been attracted to an object depends on many factors including level of attraction for the object, and location of the object relative to the start points, end points and other objects.

The simulation software, the analyser and their Graphical User Interface (GUI) were developed in the MATLAB R2010a environment (Figure 2.9). Our GUI provides several tools to change different parameters of simulation. This includes changes in maximum number of agents in the scene, methods of training the classifiers, image plan, start points, end points, number of speed categories, number of sight categories, levels of attraction for attractive objects, animation settings and visualization settings. The interface also provides the ability to save a running simulation and load it in the future. This helps designers to change a variety of settings and examine the impacts of those settings on the same design. Employing MATLAB and developing the simulation software from scratch enables us to store the simulation results in any desired data format. This provides enough flexibility for other software developers to use the proposed simulation results in other software and programming platforms.

Figure 2.11 and 2.12 illustrate typical simulated behavioural characteristics for an agent. As shown in these figures, although the agent could see two objects, it was attracted to the first one and did not like the second one. The period that the agent was attracted by an attractive object is called the visiting period (dark grey areas in the figures). During this period the agent was not moving and most of the behavioural characteristics remain constant. Just before the visiting period there is the *becoming attracted* period. In this period the agent moves towards the object with the highest allowed speed defined in its speed category (Figure 2.11).

The β curve also shows a significant change during this period, which represents change in direction of movement. Figure 2.13 highlights trajectories for normal and attracted agents in scenario 10. The type of each track is identified by the analyser using the simulated values for $((x_t, y_t)$, β_t, $\|s\|$, α_t and $d\alpha_t/dt)$.

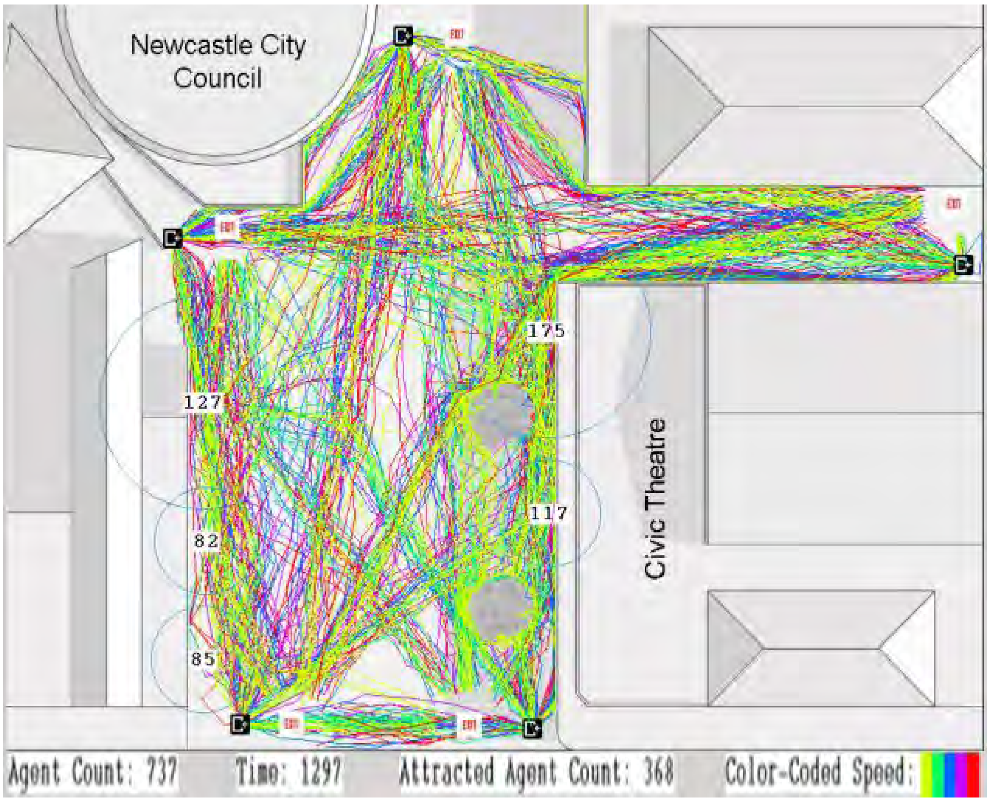

Figure 2.8: Colour-coded speed representation in scenario 2

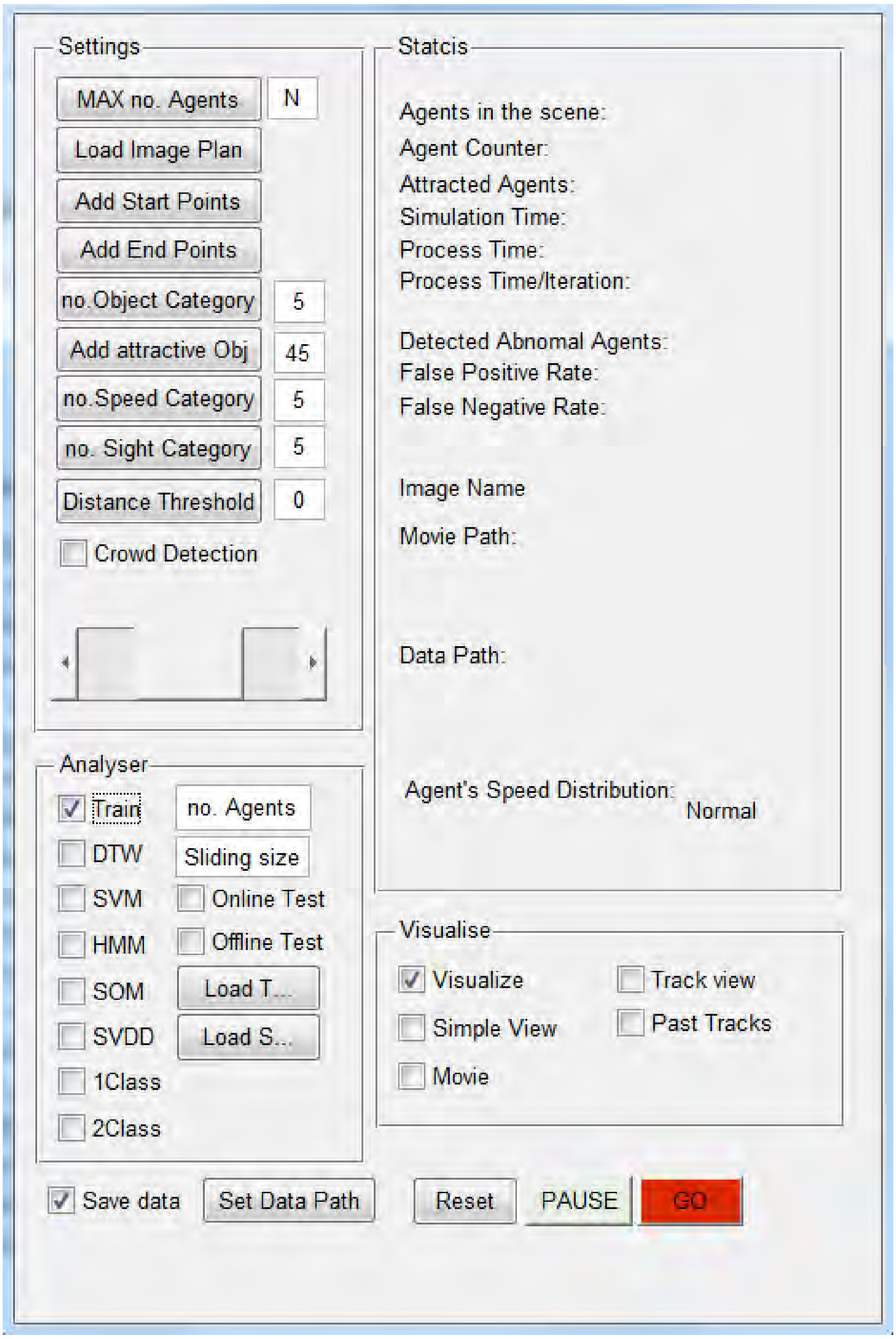

Figure 2.9: Graphical User Interface for the simulation software

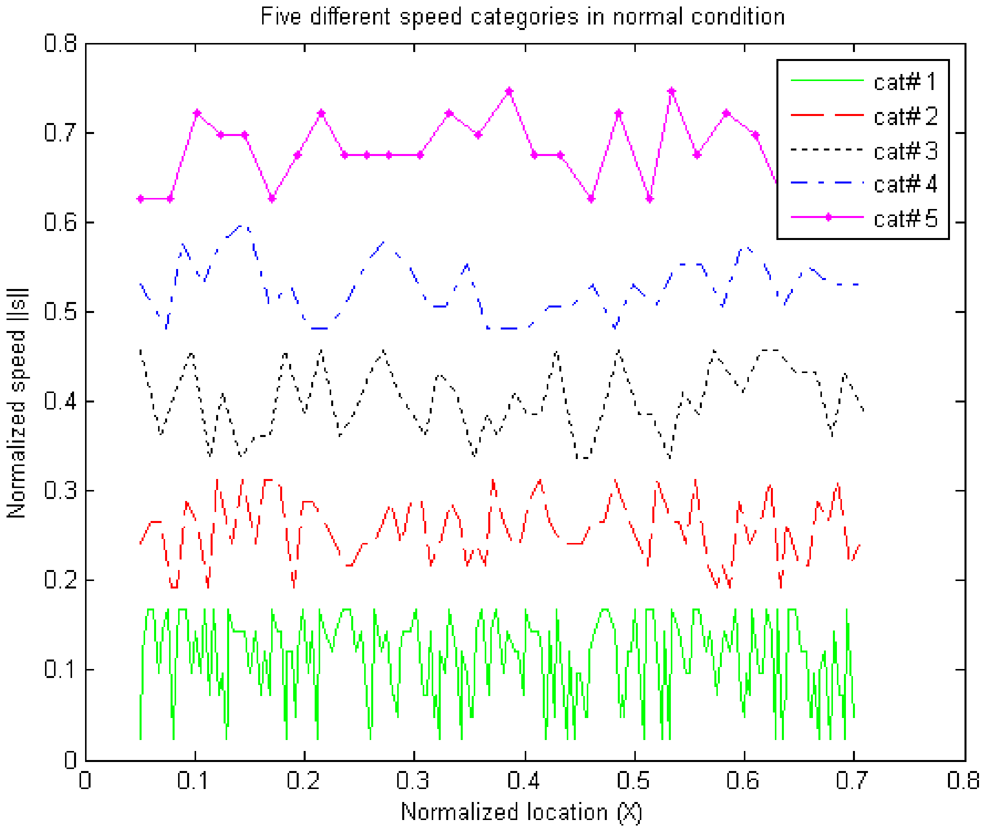

Figure 2.10: Five agents with five different speed categories

While the actual level of attractiveness of an object, to any given person, cannot be predicted with any accuracy, we can make several informed assumptions about object attractiveness to support the early stages of the simulation testing. For example at Wheeler Place there are five obvious locations that can represent attractive objects, including the art work and the Civic Theatre, which can be regarded as highly attractive and the rest as less attractive. In Figure 2.4, five positions in which to put attractive objects were indicated on the plan. We assume there are two levels of attraction: *high* and *low*. Table 2.2 lists 10 possible scenarios [4] that were considered in the simulation experiments of Wheeler Place (Figures 2.15, 2.16, 2.17 and 2.18).

Figure 2.14 shows the probability of attraction for different positions in all possible scenarios with respect to the number of simulated agents. The results shown in this figure were obtained from simulating 5000 pedestrians for each scenario. As shown, the 4th position has the highest probability of attracting agents in all scenarios. Placing a highly attractive object in this position and a less attractive object in 5th position (as in scenario 2) will balance the attracted crowds on both sides of Wheeler Place. The last column of Figure 2.14 shows the probability of attracting a pedestrian to an attractive object in Wheeler Place. These simulations demonstrate that arranging the objects based on scenario 2 will result in the highest number of attractions compared to the other scenarios.

Pedestrian spatial behaviours depend on different plans, personal behavioural characteristics, and social preferences. Using analytical models [63] to simulate these behaviours based on average characteristics, leaves a considerable proportion of behaviour categories unexplained. Using multi-agent-based cellular models, on the other hand, to overcome this problem has its own difficulties. In the real world, pedestrian movements include a great variety of speeds and directions. Using cellular approaches [23, 24, 177] to model pedestrian spatial behaviours restricts this variation to several available cells for each agent. The cellular approaches not only restrict freedom of choice in direction

[4]Because we have 5 positions to put 5 objects, two of which have a low level of attraction and three with a high level of attraction, the number of permutations for placing objects in positions without repetition is $5!/(3! \times 2!) = 10$.

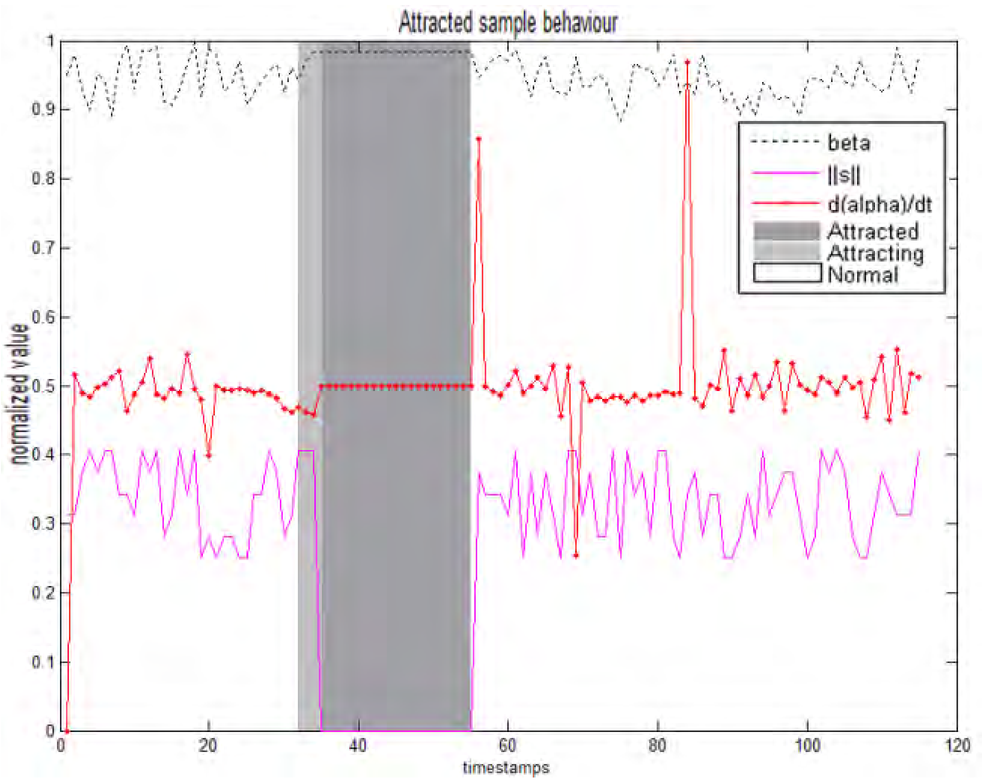

Figure 2.11: Simulated behavioural parameters for an agent; visiting period (dark grey area), becoming attracted period (light grey area), not becoming attracted period (patterned area) normal behaviour (white area)

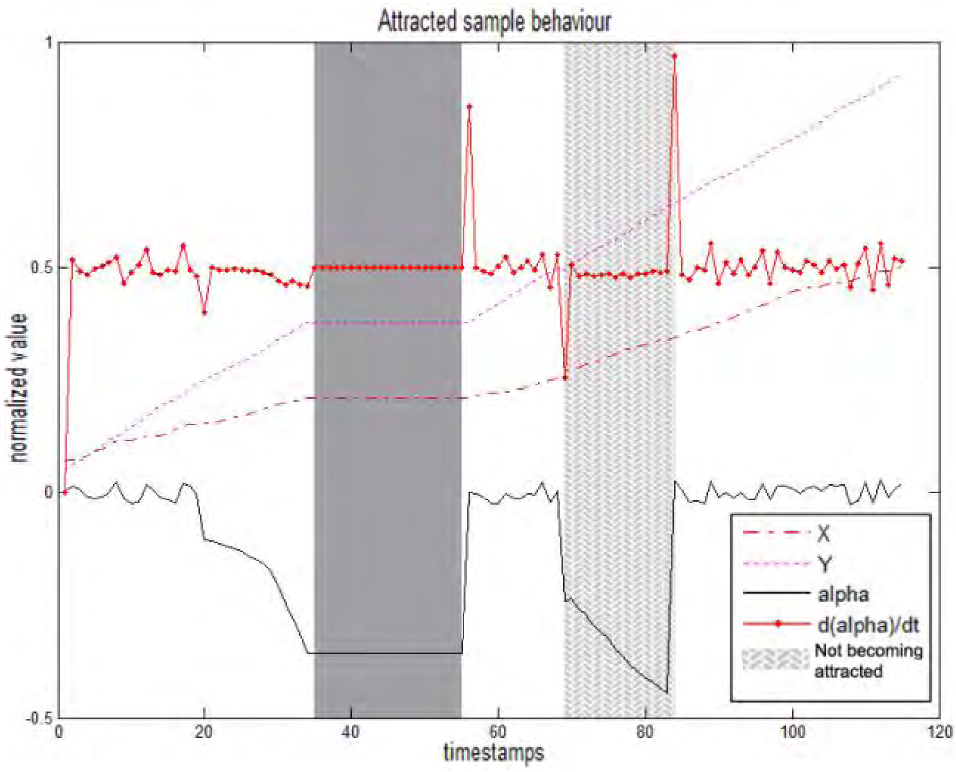

Figure 2.12: Simulated behavioural parameters for an agent; visiting period (dark grey area), becoming attracted period (light grey area), not becoming attracted period (patterned area) normal behaviour (white area)

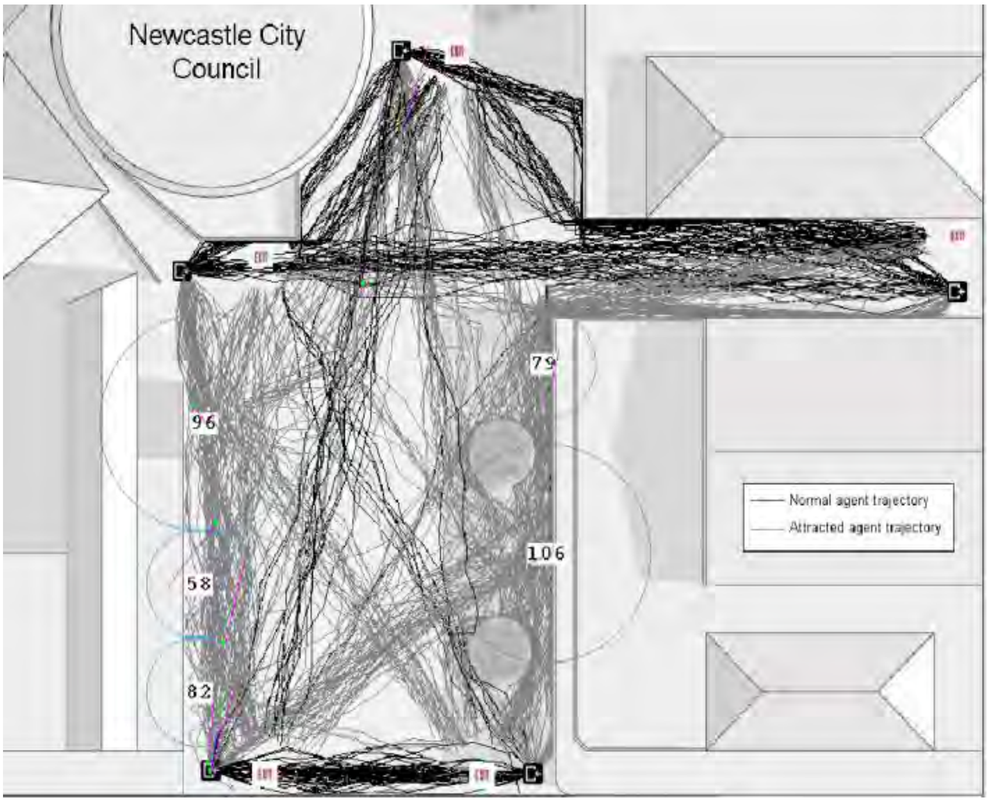

Figure 2.13: Normal (black) and attracted (grey) trajectories of agents in scenario 10

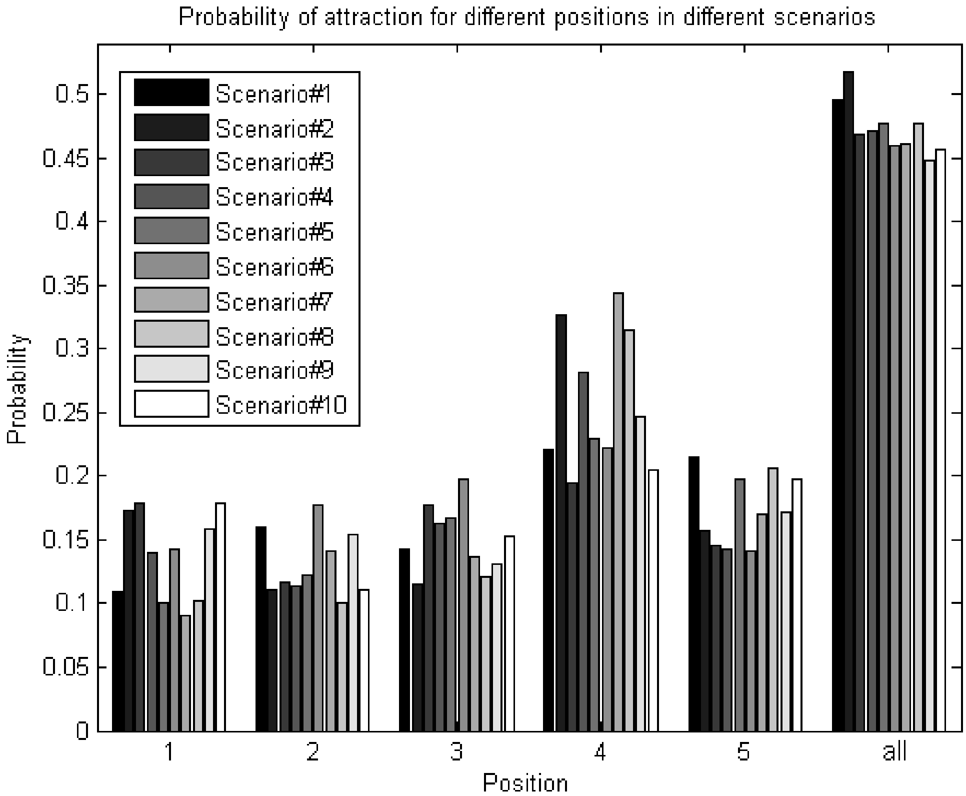

Figure 2.14: Probability of attraction for different positions in different scenarios

Scenario#	Pos #1	Pos #2	Pos #3	Pos #4	Pos #5
1	low	high	low	low	high
2	high	low	low	high	low
3	high	low	high	low	low
4	low	low	high	high	low
5	low	low	high	low	high
6	low	high	high	low	low
7	low	high	low	high	low
8	low	low	low	high	high
9	high	high	low	low	low
10	high	low	low	low	high

Table 2.2: Possible scenarios for two categories (low, high) of attractive objects in Wheeler Place

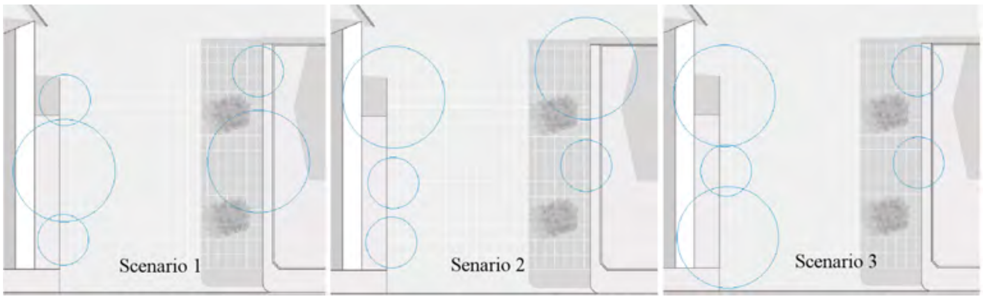

Figure 2.15: Scenarios 1, 2 and 3.

at each step but also agents' speed can only be generated based on the cell sizes. In contrast, in the proposed approach agents can be designed with a high degree of freedom. They can have a wide variety of speeds and directions at each time step (as shown in Figure 2.11 and Figure 2.8).

2.4.2 Results with Crowd Attraction Impacts

The new software system runs a pedestrian simulation on a plan of Wheeler Place. This place has been selected because: The space is next to Newcastle's Civic Theatre and City Council Chamber. These two buildings have made this area one of Newcastle's most crowded public spaces. Here *crowd attraction* is a

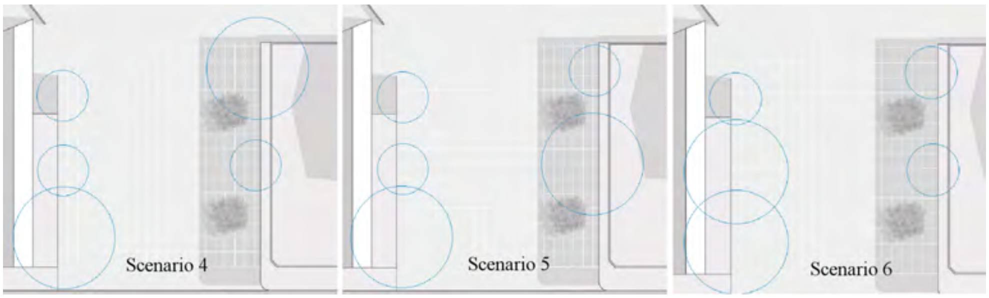

Figure 2.16: Scenarios 4, 5 and 6.

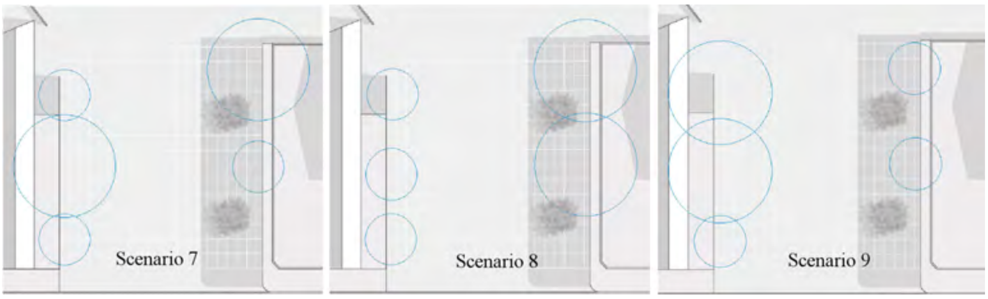

Figure 2.17: Scenarios 7, 8 and 9.

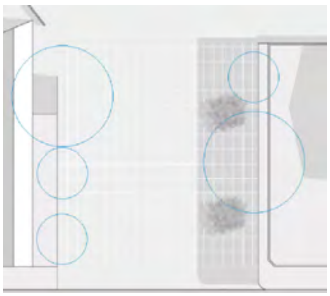

Figure 2.18: Scenario 10.

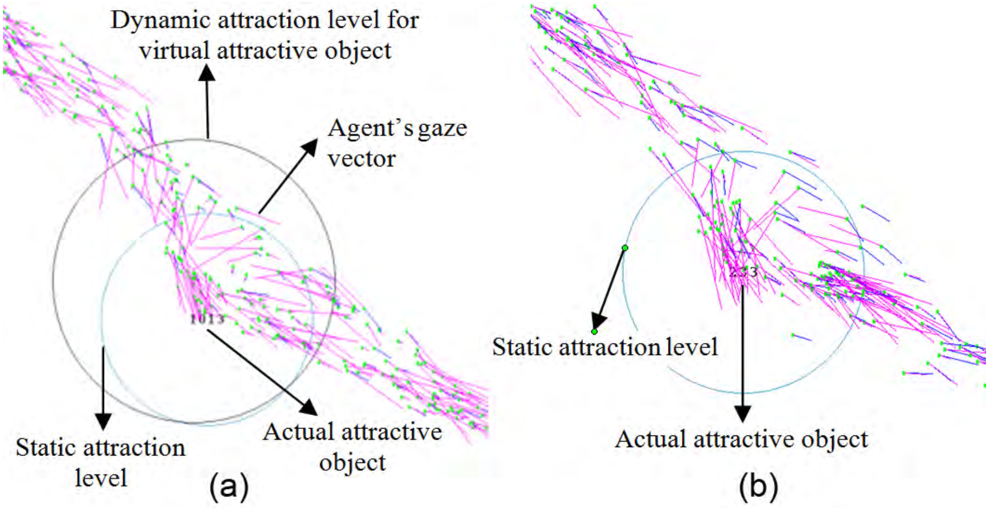

Figure 2.19: Agent's behavioural characteristics with (a) and without (b) the impact of a VAO

common behaviour, and there are several places of different levels of attraction such as the City of Newcastle Information Centre and Climate Meter, Juicy Beans Restaurant and Internet Cafe, a big public art work, the Civic Theatre and Civic Theatre Restaurant.

Figure 2.19(a) shows the impact of *dynamic attraction level* on agent trajectory. Agents who are looking at an attractive object may form a group and this is shown as a virtual attractive object. The level of attraction of a virtual attractive object changes according to the number of agents in the group. Agents can be distracted by VAOs as well as by actual attractive objects.

As shown in Figure 2.19(b) some agents who have a small *sight value* might miss the attractive object. In Figure 2.19(a) the dynamic level of attraction is large enough to direct all agents' attention and therefore all agents can see the attractive object. Agents' behavioural response to virtual attractive objects models pedestrian behaviour in the real world where crowd attraction is a significant component of visual behaviour.

Figure 2.20 illustrates typical simulated behavioural characteristics of an agent with the impact of VAOs. It shows the speed of movement $\|s\|$ and the speed of the angle between the movement direction and gaze vector ($d\alpha/dt$).

The period that the agent was attracted by an attractive object is called the visiting period. Just before this period there is a *becoming attracted* period. In this period the agent moves towards the object with the highest allowed speed defined in its speed category (see the black curve in Figure 2.20). Agents might turn their attention to the VAOs prior to the actual ones.

The patterned area in Figure 2.20 represents this behaviour. As shown in this figure ($d\alpha/dt$) shows significant change once an agent's attention is directed to a virtual/actual attractive object. This describes a pedestrian's behaviour when he/she suddenly changes gaze to a visually attractive object. The proposed analyser can detect this behaviour.

In the previous section ten possible scenarios (SC1-SC10) were defined and considered in the simulation experiments of Wheeler Place. The scenarios describe the permutations of the positions of five selected attractive objects (blue circles in Figure 2.7) [93]. In this section we employed them again with the new system to show the effects of VAOs and ANVs on the number of attracted agents.

Table 2.3 shows the average probability of attraction for different positions in different scenarios with respect to the number of simulated agents. The results in this table were obtained from simulating 5000 pedestrians for each scenario. These results suggest three conclusions:

1. The introduction of ANVs always decreases the average probability of attraction.

2. The introduction of VAOs (i.e. step from row 1 to row 3 or from row 2 to row 4 in Table 2.3) increases the average probability of attraction in the absence of ANVs for all scenarios. However, in the presence of ANVs this increase was not consistently observed.

3. Scenario 2 (shown in Figure 2.19 and SC2 in Table 2.3) has the highest probability of attraction independent of the presence or absence of VAOs or ANVs.

The experiments also confirm that if we increase the number of agents in

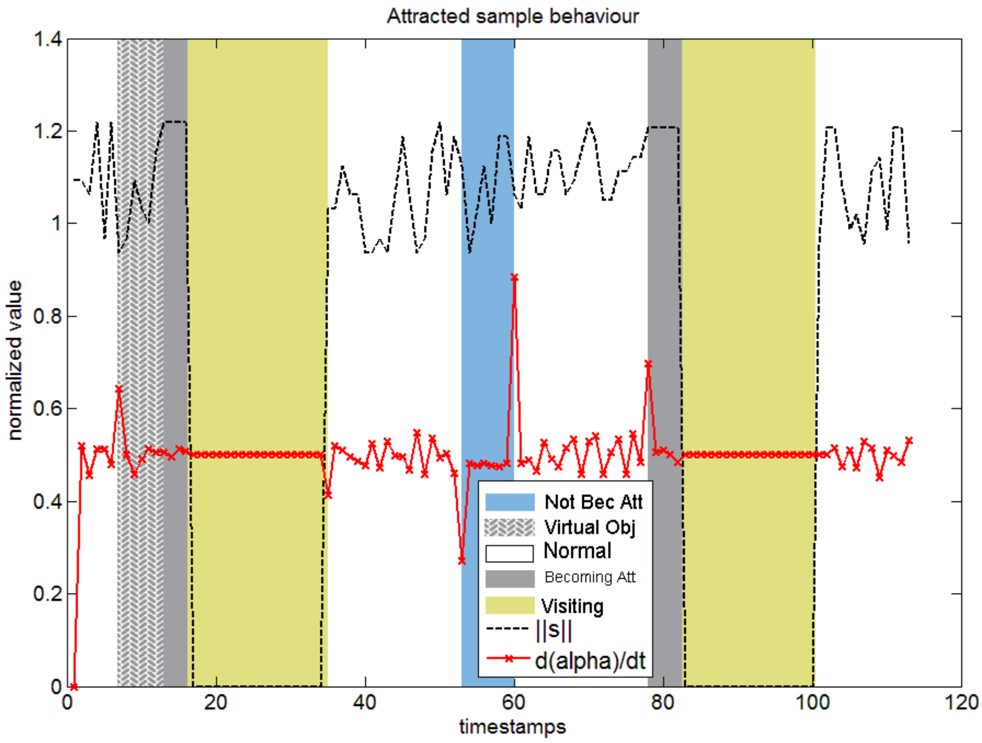

Figure 2.20: Simulated behavioural parameters; becoming attracted to a virtual attractive object (patterned area), becoming attracted to an actual attractive object (grey area), visiting period (green area), attention without becoming attracted (blue area)

VAO	ANV	SC1	SC2	SC3	SC4	SC5	SC6	SC7	SC8	SC9	SC10
NO	NO	0.49	0.52	0.46	0.47	0.47	0.45	0.46	0.47	0.44	0.45
NO	YES	0.25	0.26	0.24	0.19	0.25	0.22	0.24	0.25	0.18	0.21
YES	NO	0.54	0.57	0.52	0.53	0.55	0.51	0.54	0.54	0.50	0.52
NO	YES	0.24	0.32	0.30	0.25	0.24	0.27	0.24	0.23	0.29	0.23

Table 2.3: Average probability of attraction

the scene, the average probability of attraction will be increased. This clearly shows the impact of crowd attraction on pedestrian spatial behaviour.

2.5 Summary and Discussion

This section records the development and testing of a new multi-agent-based software simulation for pedestrian spatial behavioural analysis in urban space. The simulation adds two unique features to the conventional model: gaze vector and attractor objects. The benefits of this new approach become especially clear in cases of complex spatial arrangements where changing the configuration of walking environments (and thus adapting designs) is possible. A software system and its GUI were developed to read plans, extract boundaries and obstacles from them, run the simulation and then analyse behavioural characteristics of simulated agents. The GUI offers a user friendly environment for architects who may not be familiar with complex computer programming problems.

The dependency of attention and visual gaze direction was discussed. An innovative model was proposed and tested to simulate goal-driven attention and stimuli-driven attention with the *agent need vector* (ANV) and the *object level of attraction*. The proposed model includes *group formation* and the effects of *group size* on directing pedestrians' attention. The simulation was run for a real-world space, Wheeler Place in Newcastle. Different scenarios with different configurations and the impacts of considering crowd attraction on pedestrian behaviour were presented and discussed. The experimental results demonstrate that the new system can provide significant support for understanding how changes in the configuration of the physical/visual built environment are reflected by measurable changes in agent behaviour.

To compare the simulated and real-world behaviours and provide real-world behavioural characteristics to the analysis system, the next chapter will describe the proposed system for pedestrian detection and tracking. The system employs a single optical camera on a fixed platform to track individuals at Wheeler Place.

Chapter 3

Pedestrian detection and tracking

Contents

This chapter explains the methods that are utilised to extract objects from the background image, distinguish pedestrians from other objects, and track detected pedestrians. The methods are developed to track pedestrians at Wheeler Place. Extracted trajectory data will be used in the proposed outlier detection system to analyse pedestrians' walking patterns.

3.1 Background and Previous Works

3.1.1 Region of Interest Reduction

A region of interest (ROI) is a selected subset of samples within a data set identified for a particular purpose. In image processing it is defined as a portion of an image that you want to filter or perform some other operations. The concept of a ROI is commonly used in image processing and pattern recognition. An ROI contains no image data. It is not an image itself, but a placeholder that remembers a defined location within an image [133].

The impacts of using ROIs become more significant when different algorithms are applied on different areas of an image. For instance in pedestrian detection algorithms, ROIs contain candidate regions where pedestrian existence is possible. As the algorithm progresses towards recognising pedestrians, the number of candidates will be reduced and eventually all regions contain pedestrians only. This routine is called *multistage ROI reduction*. A set of algorithms is applied at each stage to narrow down the number of candidates. The complexity of the algorithms increases as the pedestrian detection progresses to its final stage. Since the number of candidates in later stages is lower than early stages, it is more efficient to apply complex algorithms on the later stages.

The first stage in ROI reduction consists of algorithms to distinguish between background and foreground objects. There are two types of background; "moving background" and "fixed background" [180, 131]. Moving background is a result of installing the camera on a moving platform (such as vehicle). In this case, the background detection is a complicated and time consuming process and it needs additional hardware. Authors in [155] modeled the intensity value of each pixel as a Gaussian distribution that can be learned and adopted along the image sequence. In their approach each pixel in each frame is modeled as an independent statistical process, which is very time consuming. A special hardware is required to obtain background images in real-time using multiple vector calculation in parallel. In the applied method for pedestrian detection and tracking at Wheeler Place an optical camera installed on a fixed platform is used.

The most popular approach to detect background in a fixed platform camera is to use an initial background image. The initial background image is obtained when there is no object in scene. Using an initial background image is very sensitive to light changes. To solve this problem, instead of using one static background image, a dynamic background image with an updating method is used. In the field of pedestrian detection and tracking, the average framing has been employed for upgrading the initial background [71, 83, 147, 131]. The average framing is based on averaging the intensity values of background regions in a stack of images [31]. It could be employed along with the "block matching algorithms" to obtain more accurate results [95, 30]. In the proposed method for pedestrian detection, the average framing along with a block matching algorithm is employed for background detection.

Depth information has been utilised to distinguish between background and foreground objects and also to reduce the number of hypotheses in ROIs [49]. By using depth information, we are able to estimate the real-world size of an object in the image and compare it with the size of the object (pedestrian) that we are looking for [59].

Stereo cameras have been extensively used to extract depth information from images [206, 78, 103]. Stereo cameras have been used in order to segment the scene into blobs using disparity discontinuity. A split-and-merge procedure has been applied to form objects with size/shape constraints for pedestrians [206, 103]. The most common algorithm used for calculating the distance of objects using stereo cameras is the "dense disparity map estimation" algorithms [104, 49, 59, 105]. A dense disparity map shows the relative distance of objects to each other in the scene. It uses the difference between the location of an object in the left and right images and the camera specifications to find the distance of objects with respect to the camera. Currently stereo cameras have a small field of view and therefore using them to cover large outdoor areas would not be feasible. Also object matching algorithms to obtain the dense disparity map are very sensible to noise and they have been employed in indoor areas [103].

3.1.2 Pedestrian Detection

Feature extraction is a fundamental stage in detecting pedestrians. Features describe the possible patterns of pedestrians' visible characteristics. Two types of features have been extracted to recognise pedestrians in digital images:

1. *Motion-based features:* These features describe pedestrians' movement dynamics in a sequence of video frames. They represent differences between human dynamic characteristics and other objects in the scene. Speed of movements, acceleration, and direction of movements are examples of these types of features. The block matching algorithm has been extensively used to estimate the "motion vector"[1] for each block in the image [95]. The motion vectors improve background modelling by reducing the number of candidate regions in the ROIs [30, 61, 59, 71]. For instance excluding regions that move faster than the highest possible speed for pedestrian movement can reduce the number of non-pedestrian regions considerably [30, 71].

2. *Shape-based features:* Shape-based features describe the appearance of the objects, such as height, width, height to width ratio, shape, and area. The simplest usage of these features is to set a threshold to remove objects when their shape-based features do not fall into the desired range defined by human body shape. For instance in [59, 180] a threshold has been enforced on both height and width of the foreground objects to remove too wide and too tall objects from ROIs. Shape-based features and motion-based features could be combined to remove non-pedestrian objects from ROIs more accurately [95].

Extracting features to detect pedestrians has also been performed using "active sensors". An active sensor is a remote-sensing system that transmits its own radiation to detect an object or area for observation and receives the reflected or transmitted radiation. Table 3.1 summarises different types of

[1] The motion vector is the key element in the motion estimation process. It is used to represent a block in a picture based on the position of the block (or a similar one) in another picture, called the reference picture.

Sensor Type	**Range**	**Feature**	**Cost**	**Algorithm Complexity**
Optical Camera	Medium	Rich color and shape information	Cheap	High
Laser Scanner	Large	Work in darkness, cold and hot weather	Very expensive	Low
Thermal Infrared	Medium	Work in darkness	Medium	Medium
Radar	Small	High range resolution	Expensive	Low

Table 3.1: Comparison of different type of sensors

active sensors used in the pedestrian detection research area. In terms of image processing, active sensors refer to special types of sensors that provide physical information of objects in a scene, such as temperature and distance. Using active sensors was not very common in the past because they were too expensive and not easily accessible. Although they are currently affordable for applications that need few of them, they are still too expensive for applications that need many sensors to cover a large outdoor area.

Because thermal infrared radiation is emitted by the human body, thermal infrared sensors have been used for pedestrian detection extensively [122]. Infrared sensors can only be used under special conditions. For example pedestrians usually wear very thick clothes in winter or cold weather so their body temperature cannot be sensed by an infrared sensor. We have the same problem in hot weather conditions. In some hot weather conditions where the environmental temperature is higher than the body temperature or where it is close to body temperature, thermal infrared radiation is emitted by the environment more than the body. So in both cases, very hot and very cold weather conditions, infrared sensors cannot provide the information that we need to detect pedestrians [122, 41]. For solving this problem sensor fusion[2] has been recommended. Infra-red sensors and laser scanners have been used

[2]Sensor fusion is the combining of sensory data or data derived from sensory data from disparate sources such that the resulting information is in some sense better than would be possible when these sources were used individually.

in combination with visual cameras in order to improve the performance and reliability of visual systems [21, 41]. Currently, the use of laser scanners and infrared sensors is too expensive to cover large areas.

The next stage, after extracting the desired features from objects in ROIs, is to use the features in a learning-based technique to model pedestrians [48, 103, 21]. [148] pioneered the use of Haar-wavelet features in combination with a Support Vector Machines (SVMs) [181]; this approach was subsequently adapted by [47] and others. To reduce the complexity of pedestrian detection algorithms, component-based analysis has been utilised. Dividing ROIs into sub-regions is one of the most popular approaches that uses component-based analysis [170]. Individual results of analysing each component are combined by a second layer of classifiers to recognise pedestrians. While dividing ROIs into some fixed sub-regions simplifies the detection, authors in [133] have employed a more dynamic method and constructed sub-regions according to the location of certain body parts. The proposed approach in [133] has extended the work of Papageorgiou and Poggio [148] to four component classifiers for detecting head, legs, and left/right arms separately. Additional attempts have been made towards reducing classification complexity by manually separating the pedestrian training set into non-overlapping sub-sets (i.e. based on pedestrian heading direction or gaze vector) [170, 174].

3.1.3 Pedestrian Tracking

Tracking has been proposed to localise the objects of interest (pedestrians) in time space. Though as a natural extension of detection, tracking has its own problems in recognising and identifying pedestrians in consecutive frames. Tracking could be regarded as the most popular topic in visual surveillance.

Template matching algorithm is the most popular method for visual tracking. The templates with different sizes have been sequentially applied to detect pedestrians with different shapes and sizes [41, 180]. The algorithm has been coupled with a graph-matching-based tracking algorithm combined with "Hausdorff distance"[3] to compare two point sets and find the best match be-

[3] Hausdorff distance [43] is the "maximum distance of a set to the nearest point in the other set."

Table 3.2: Summary of previous approaches for pedestrian detection and tracking.

References	The Main Purpose	Feature Extraction Method	Hardware
[103]	Intelligent transportation systems	Motion estimator	Stereo cameras
[122]	Intelligent transportation systems	Sensor fusion	Infrared sensors
[100]	Guiding idle customers at a shopping centre	Background modelling using a sensor network	Laser scanners
[21]	Pedestrian localisation in urban areas	Sensor fusion	Laser scanners
[49]	Collision avoidance	Depth information	Stereo cameras
[101]	Estimating visiting patterns	RFID tag localisation	RFIDs
[41]	Detection and tracking at night time	Head detection	Infrared sensors
[71]	Detection and tracking	Dynamic background estimation	Optical cameras
[95]	Collision avoidance	Motion estimators	Optical cameras
[179]	Evaluating visitor's behaviour	dense grid visibility graph	Robots

tween templates [83]. Table 3.2 summarises different methods for pedestrian detection and tracking.

Depth-based features are commonly used to detect and track pedestrians. As shown in Table 3.2 depth-based features have been extracted using laser scanners, stereo cameras, singular optical cameras, RFIDs, and infrared sensors [55, 56]. However, compared to active sensors of this nature, optical vision provides a much higher spatial resolution at a lower cost. Furthermore, being passive, there is little potential for interference with the environment [60, 99].

Some 3D techniques adopt an overhead viewpoint for the camera stereo rig in order to minimise the occlusion between people that can occur with more oblique camera angles. In [74] a stereo camera is installed above a door, pointing downwards, towards the ground-plane with a view to count shoppers as they enter or exit a retail environment. Techniques using this stereo camera setup have the same disadvantages as 2D techniques that employ this viewpoint. In general, stereo cameras are only applicable to indoor scenarios, which restricts the maximum height at which the camera can be placed. For example, for retail scenarios the ceilings are only 2.5-3 meters high [15]. The field of view can be limited in this short height unless a wide field of view lens is employed. However, this type of lens can result in significant occlusion problems [74]. Therefore, with overhead camera viewpoints a trade-off exists between the field of view and occlusion. A different approach is taken in [75], which introduces another plane-view statistic called the height map. In the height map each ground-plane bin contains the highest point above the ground-level plane that is projected into that bin. It is effectively a simple orthographic rendering of the shape of the 3D point cloud when viewed from overhead [75]. The pedestrian tracking from the height map can be achieved in a similar manner to that of occupancy or volumetric maps, for example [86, 7] simply threshold the height map and use connected component analysis (CCA) to obtain pedestrian regions. However, using this approach the movement of relatively small objects at heights similar to those of peoples' heads, such as when a book is placed on an eye-level shelf, can appear similar to the motion of a person in a height map [74].

There are two broad types of methodologies for tracking algorithms that use 3D information; continuous detect-and-track and single detect-and track. 3D information is mainly used to resolve ambiguities and therefore can be used within both approaches. In tracking pedestrians using 3D information, the position of a person in the current frame is typically used to disambiguate between pedestrians in subsequent frames. This position is defined by some 3D feature point of the object, for example the 3D position of the top, bottom or centroid of the person [102], or the centroid of the person's head region [13, 123]. Some of these features are more robust than others, for example the centroid of a 3D cluster can be more robust than the top or bottom of the cluster region, particularly in outdoor scenes where the object may be far from the camera [17]. However, during occlusion the real and estimated centroids of a full body region may differ significantly, whereas the centroid of the 3D head region can be more robust and remain relatively unaffected.

Using depth information along with several other shape-based features that can describe the differences between pedestrians' bodies and other objects in the scene as the inputs of a learning technique, will result in an accurate pedestrian detection and tracking system. The following subsections describe my approach to detect and track pedestrians using a single optical camera installed on a fixed platform at Wheeler Place.

3.2 Pedestrian Detection and Tracking at Wheeler Place

In this section my pedestrian detection and tracking system is described. The system extracts pedestrian trajectory data from a real-world architectural space called Wheeler Place. As discussed in the previous chapter, Wheeler Place is located between two busy streets in Newcastle, Australia. The place is surrounded by several attractors, which make it a suitable choice for the analysis of pedestrian spatial behaviour.

The first stage in the system is ROIs reduction, which includes foreground detection using motion based features. The second stage is recognising pedestrians using shape-based and depth-based features, and a classification method to distinguish pedestrians from other similar objects. The third and last stage is tracking pedestrians using depth-based and motion-based features. The whole system is developed to work at Wheeler Place using a fixed-platform camera. The camera was installed at Wheeler Place for seven days to collect video data. The video data was analysed to extract pedestrian trajectories. The trajectory data were then used in the proposed behaviour analyser for outlier detection, which will be described in later chapters.

Figure 3.1: Initial background image, where there is no object in the field of interest.

3.2.1 Region of Interest Reduction

The first step in defining the initial ROIs is to extract moving objects from the image. This is known as *background detection*. As discussed in the previous Section, the most common method for detecting background objects using a fixed platform is background subtraction. The same method is employed for background detection in the proposed approach for pedestrian detection. The initial background image, which does not include any moving object, is shown in Figure 3.1. However, changes in general visual features the scene such as changes in daylight or weather conditions cannot be handled using a fixed background image. The result of background detection using one fixed background image is shown in Figure 3.2. To overcome this problem a "dynamic background detection" technique is employed. The technique dynamically upgrades the background image to adopt the scene characteristics in the current frame.

To describe the background detection technique an example is given. As-

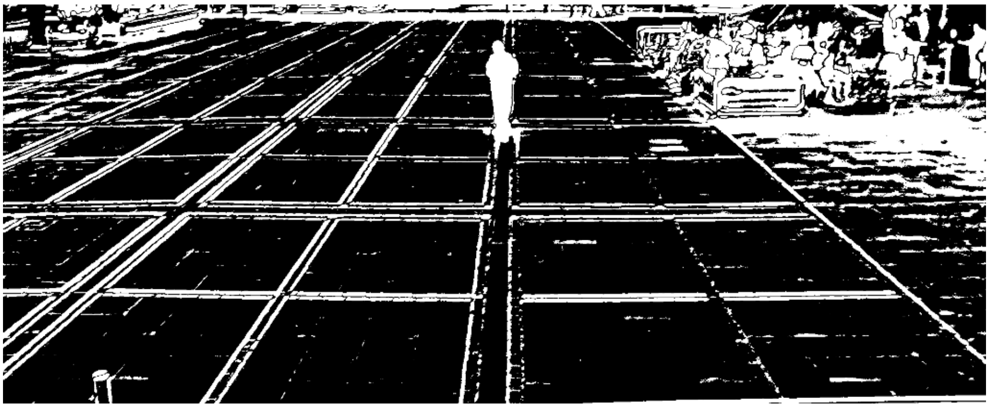

Figure 3.2: Background subtraction using a fixed background image. Use of the fixed background is very sensitive to noise such as illumination changes and small camera movements.

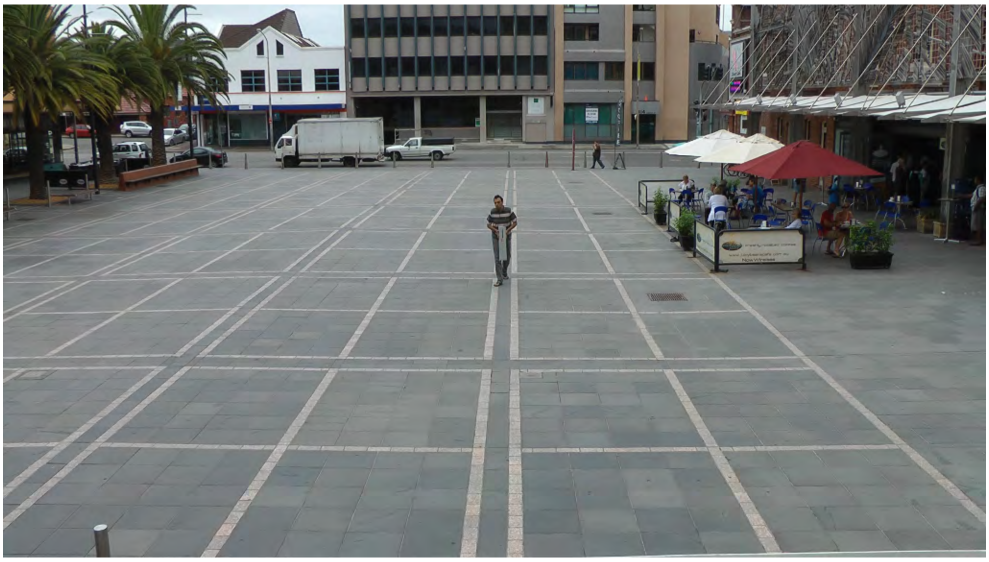

Figure 3.3: An example frame to apply background detection algorithms (first frame in the sequence.

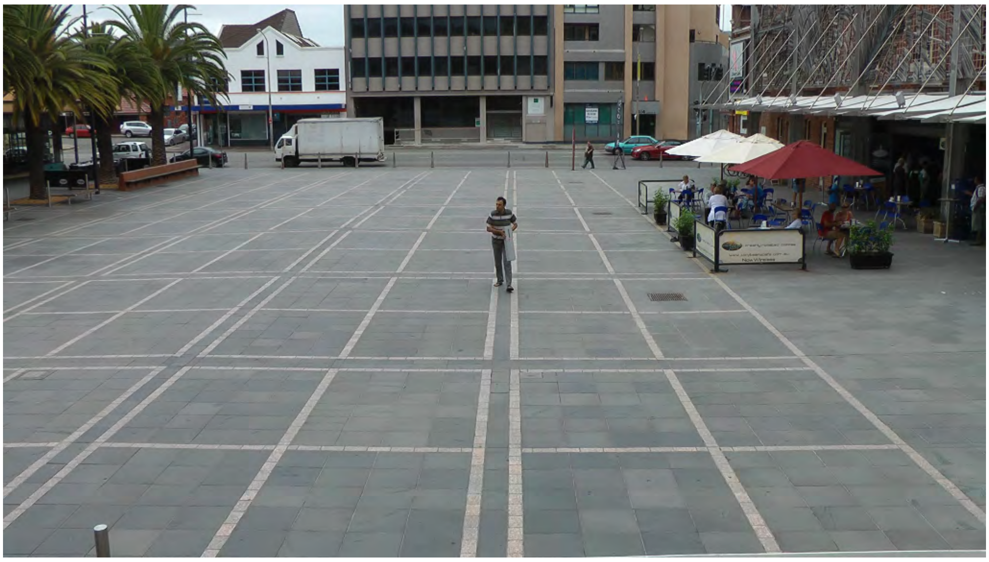

Figure 3.4: An example frame to apply background detection algorithms (second frame in the sequence).

sume we want to detect the background in the frame shown in Figure 3.3 using the initial background image shown in Figure 3.1. To make the edges of the objects smoother and remove all the small artifacts a smoothing technique is employed. This includes applying a circular average filter that convolves the image with a uniform circular averaging filter. Figure 3.5 shows the result of applying this filter on Figure 3.3. As we are interested in pedestrians' behaviour at Wheeler Place only (the grid patterned area), top part of the scene (Hunter Street) is excluded from the images. The same filter is employed on the background image.

Then the current frame and the background image are subtracted. As shown in Figure 3.6 the resulting difference image contains pixels with an intensity that has changed. To make the moving parts more clear, Otsu's method was applied. The method chooses the threshold to minimise the intraclass variance of the black and white pixels, to extract objects from the image [146]. The result of applying Otsu's method on Figure 3.6 is illustrated in Figure 3.7.

A threshold is defined to check whether the background image contains

Figure 3.5: Result of applying the circular average filter on Figure 3.3.

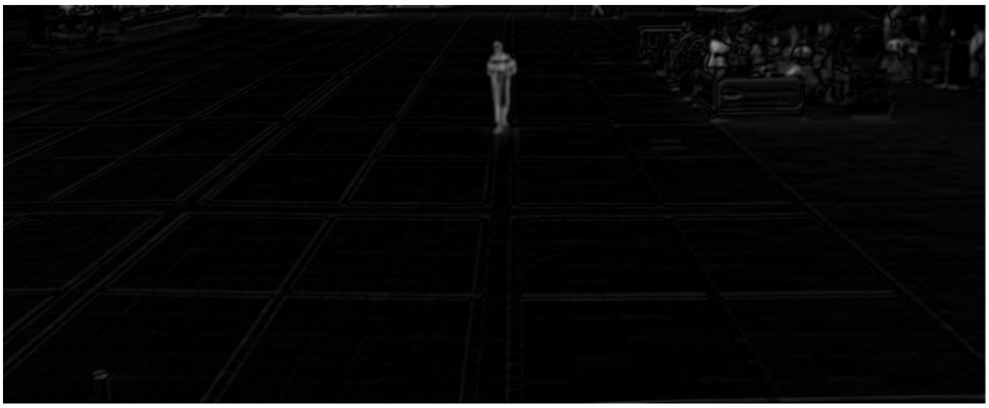

Figure 3.6: Result of image subtraction between the images shown in Figure 3.5 and initial background image.

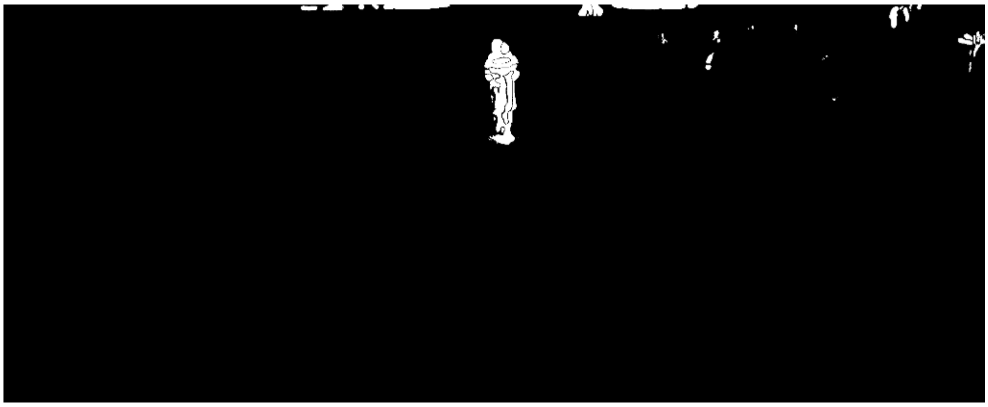

Figure 3.7: Result of applying Otsu's method [146] on Figure 3.6.

only background objects. We call this threshold *white ratio* (W_r). For extracting pedestrians at Wheeler Place, W_r is set to 0.18. This value is chosen experimentally and may vary for other locations with different environmental settings. If the proportion of the total number of white pixels to the number of pixels in the image shown in Figure 3.7 is greater than W_r, then it is likely that the background image needs to be updated. If this happens for five sequential frames in a row then the following algorithm will be utilised to update the background image.

To update the background model a method using Local Binary Pattern (LBP) is employed [76]. In the method, each image block is modelled as a group of weighted adaptive LBP histograms. LBP is invariant to monotonic changes in grayscale. This makes the method robust against light changes. The method compares the histogram of the current frame with the existing K histograms of the background models using a distance measure. The "histogram intersection" is used as the distance measure [76]. The histogram intersection for the normalised histograms x_1 and x_2 is defined as:

$$H(x_1, x_2) = \Sigma_i \min(x_{1,i}, x_{2,i}) \tag{3.1}$$

where i is the bin index of the histogram. Foreground detection is achieved via

Figure 3.8: Connected components, which are obtained using Moore-Neighbor tracing algorithm modified by Jacob's stopping criteria with eight-connected neighbour analysis [65].

comparison of the new block histogram x_t against the existing B background histograms selected at the previous time instant. If a match is not found, the block is considered to belong to the foreground. Otherwise, the block is marked as background [76]. As mentioned before this method is used whenever the background evaluation process indicates that the background model needs to be updated.

The Moore-Neighbor tracing algorithm modified by Jacob's stopping criteria is employed to trace the exterior boundaries of the detected objects, as well as the boundaries holes inside these objects in the binary image shown in Figure 3.7 [65]. An eight-connected neighbour analysis using the tracing algorithm is utilised to obtain the connected component shown in Figure 3.8. Afterwards, small objects that occupy less than five pixels are considered as noise (this threshold is set experimentally). To reduce the ROIs even more, the "height to width" ratio for the human body is employed as a threshold. Regions with a height to weight ratio that falls out of a specific range, defined by the golden ratio, are removed from ROIs. The resulting image is shown in Figure 3.9.

For further reduction of ROIs, an image difference between the current

Figure 3.9: Regions with a height to width ratio that does not occur within a specific range defined by the golden ratio are removed from ROIs. Different components are shown in different colours.

processing frame and its previous frame is calculated. The two frames are shown in Figure 3.3 and Figure 3.4. The images are smoothed with the same technique described before. As shown in Figure 3.10 the resulting difference image contains pixels with an intensity that has been in the second frame. The result of applying Otsu's method on Figure 3.10 is illustrated in Figure 3.11. Regions in ROIs with less than ten percent of their pixels not labelled as "moved" in this image (Figure 3.9), are considered as non-pedestrian regions and are removed from the ROIs. The final resulting ROIs are shown in Figure 3.12.

3.2.2 Pedestrian Recognition

After reducing the number of candidate regions in ROIs, an intelligent classifier is employed to distinguish between pedestrian and non-pedestrian objects. The old approaches of recognising pedestrians have employed a sliding window. The sliding window technique shifts ROI windows of all possible sizes, at all locations over the images while performing feature extraction and pattern classification. This brute-force approach in combination with employing complex classifiers is computationally too intensive [59]. In the proposed pedes-

Figure 3.10: Result of subtraction of two smoothed frames to find the pixels that have changed intensity in the second frame.

Figure 3.11: Result of applying Otsu's method [146] on Figure 3.10 to label the moved pixels.

Figure 3.12: Final three regions in ROIs after applying several reduction steps.

trian recognition system, the number of regions in ROIs have been reduced extensively. This makes it possible to employ a powerful classifier to recognise pedestrians at this stage. The classification module used for pedestrian recognition utilises a set of features to make the distinction between pedestrian and non-pedestrian objects. The classifiers used in this module should be able to provide a complex decision boundary in a high dimensional feature space with a limited training data set. GDTW-P-SVMs are used as the classification technique. Gaussian Dynamic Time Warping (GDTW [8, 175]) is a function that is used as the kernel function in Potential Support Vector Machines (P-SVMs [79]) to construct the new classification technique. A theoretical justification and an experimental comparison will be provided in Chapter 5 to support the idea of the new classification technique.

To make the recognition system invariant to size of pedestrians in the images, the regions are scaled according to their distance from the camera. The distance of regions from the camera in Wheeler Place is calculated using a reference point and the camera Field of View (FOV). This method is also employed to calculate the real-world height of objects, which aids to track pedestrians more accurately. To obtain the distance of objects from the camera we need to calculate the FOV in x-axis and y-axis directions. As shown in Figure 3.13, the field of view of a camera can be simply obtained using Equation 3.3 and

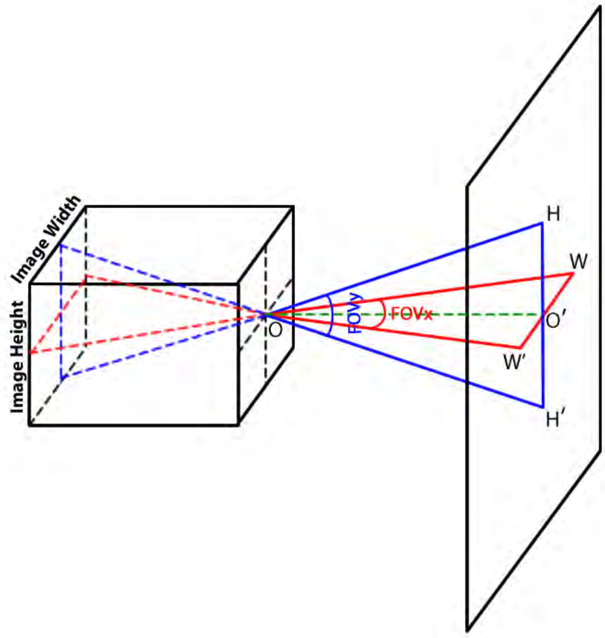

Figure 3.13: Camera field of view calculations

Equation 3.2.

$$FOV_y = 2 \times \tan^{-1}\left(\frac{O'H}{OO'}\right) \tag{3.2}$$

$$FOV_x = 2 \times \tan^{-1}\left(\frac{O'W}{OO'}\right) \tag{3.3}$$

where FOV_y is the camera field of view in the direction of the y axis, and FOV_x is the camera field of view in the direction of the x axis. FOV_y and FOV_x are used to calculate the distance of an arbitrary point in the image plane to the camera. This distance, d, could be obtained by calculating it in x-axis and y-axis directions using the following equations.

$$d_y = \tan(\theta + \phi) \times h \tag{3.4}$$

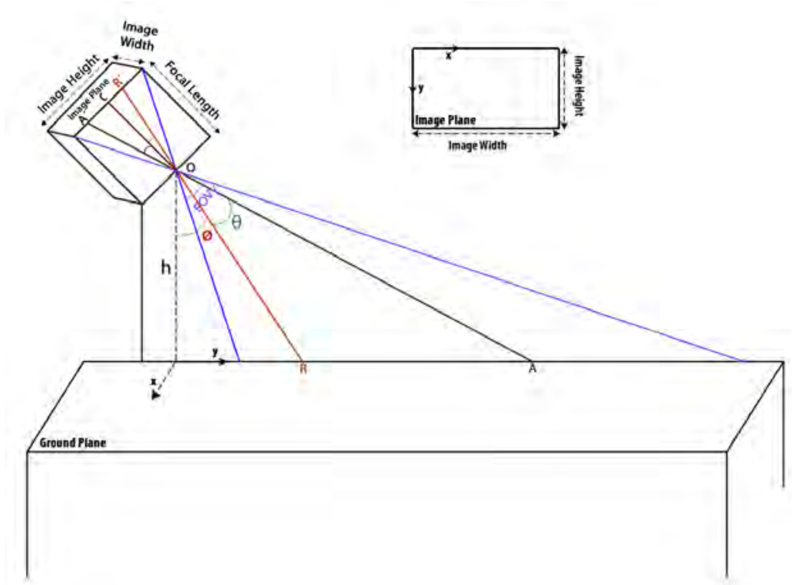

Figure 3.14: Transformation from real-world to 2D-plan coordinates

$$FL = \tan\left(\frac{180 - FOV_y}{2}\right) \times \frac{I_h}{2} \tag{3.5}$$

$$\theta = \tan^{-1}\left(\frac{R'C}{FL}\right) + \tan^{-1}\left(\frac{CA'}{FL}\right) \tag{3.6}$$

$$\phi = \tan^{-1}\left(\frac{d_R}{h}\right) \tag{3.7}$$

where FL is the camera *focal length*, I_h is the image height, d is the distance of point A from the camera, h is the height of the camera from the ground, A is an arbitrary point on the ground, and R is the reference point on the ground with a known distance of d_R from the camera. Extracting depth information from images using a reference point is a popular approach for applications where the physical characteristics of the scene and the camera, such as position of the camera, camera specifications, and distance of a reference point form the camera, are known [204]. The same calculation can be performed using the image width, and FOV_x to obtain d_x and then d.

$$d = \sqrt{d_x^2 + d_y^2} \tag{3.8}$$

$$Object_{Height} = \frac{h}{d} \times (d - d_R) \tag{3.9}$$

To detect pedestrians several features are extracted from different parts of the human body. The legs are the most important part and are very different from other common moving objects that can be seen in urban areas such as pets, and trolleys. The "skeleton" of the legs is extracted using the algorithm described in [114]. This method removes pixels on the boundaries of objects recursively but does not allow objects to break apart. The remaining pixels construct the skeleton of the objects. The skeleton has been used as an important shape-based feature in pedestrian detection applications [95]. Figure 3.15 shows the extracted skeleton from the legs part of some pedestrian image samples.

Canny edge detection [25] is employed to extract the boundaries of the detected objects. The edge detection process serves to simplify the analysis of images by drastically reducing the amount of data to be processed, while at the same time preserving useful structural information about the object boundaries [25]. The algorithm method proposed by Canny is an accurate and robust edge detection algorithm that has been commonly used in pedestrian detection systems [57].

SVM-based classifiers have shown promising results for pattern recognition in images. They have been extensively employed in pedestrian recognition applications [61]. The location of the boundary pixels of the whole pedestrian body (obtained using Canny edge detector), and the skeleton of the legs part are used as the input spaces of two DTW-P-SVMs classifiers. An object will be labelled as pedestrian if both classifiers recognise it as pedestrian.

3.2.3 Pedestrian Tracking

To track pedestrians a new set of features are extracted from the positional or shape information of the detected pedestrians (bounding boxes). The fea-

Figure 3.15: Extracted skeleton from the legs part of some pedestrian image samples [59].

tures include area, width, height, perimeter, distance between box centroids, distance between median pixels[4], distance from the camera, Sum of Squared Differences (SSD) between colour intensities, and the box mean motion vector.

These features are used to assign pedestrians to existing tracks or initialise a new track based on the position of the new detected pedestrians. Euclidean distance (ED) is used as the distance measure to calculate the similarities between pedestrians in the current frame and a stack of previous frames [36]. The first Nearest Neighbor (1-NN) [37] is used to assign the pedestrians to the tracks using the calculated similarities. The combination of ED and 1-NN provides a fast and accurate classification result in pedestrian tracking at Wheeler Place. A user-defined threshold is defined to create a new track for a pedestrian who does not match with others.

Features used to track pedestrians and a description of how they are extracted are listed below:

- *pixel-based features*: These features are simply extracted by counting the number of pixels that the detected pedestrian is occupying. They include area, perimeter, coordinate of box centroids, and percentage of pedestrian pixels in the bounding box.

- *depth-based features*: These features are obtained by mapping the 2D image plane into a 3D real-world plane using the distance of pixels in the image plane from the camera in real-world. They include distance of the pedestrian from the camera, real-world height of the pedestrian, and distance of median pixel from the camera. To obtain the distance

[4]The median pixel for an object is the average of coordinates of the object pixels in x-axis and y-axis directions.

of pedestrians the methods described in Section 3.2.2 were employed. To show the pedestrians' trajectories, the trajectories are mapped from real world-coordinates to the 2D plan of Wheeler Place. The mapping is performed using the depth information and location of the camera in the 2D plan. Figure 3.17 shows the result of mapping of the patterned grid in Wheeler Place to the plan. Red dots in Figure 3.16 are mapped to blue dots in Figure 3.17.

- *motion-based feature*: As pedestrians are more likely to follow their direction of movement in the previous frames, the motion vector is extracted and used as a feature to track pedestrians. A block matching algorithm (BMA) was employed to estimate the motion vector for detected pedestrians. The block size is defined by the size of each bounding box. The algorithm has been proposed in [95]. It searches for the best matching block with minimum distance using a search pattern. The search pattern guarantees a fast and accurate result. The distance measure used to calculate the similarities between each pair of blocks is the mean absolute difference (MAD). It is obtained by calculating the average of the absolute intensity difference between corresponding pixels in two comparing blocks.

3.3 Experimental Results

Wheeler Place is an architectural space located between the two busiest streets in Newcastle (Australia), which features a constellation of attractive objects and offers an excellent test environment for analysing different pedestrian behaviours (Figure 3.18). The space is next to Newcastle's Civic Theatre and City Council Chamber and these two busy buildings have made this area one of the Newcastle's most crowded public spaces.

An optical camera (Panasonic HDC-HS700 [35]) is used for pedestrian detection and tracking at Wheeler Place. They are easily available to the public for a reasonable price and they are not equipped with any advanced recording sensors such as laser scanners or infrared sensors. The camera is capable of

Figure 3.16: Red dots show the location of the patterned grid in Wheeler Place. These locations are then mapped to a 2-D image plan.

recoding at 50 frames per second with a resolution of 1920 by 1080 pixels. It is installed on the balcony of the City Council located at the south part of Wheeler Place at the height of 3.618 meters. Figure 3.19 shows the position of the camera at Wheeler Place. Figure 3.20 shows a sample frame, which has been captured by the camera.

The area of Wheeler Place is 23.35 meters by 48.58 meters and cars are not allowed in this area. Therefore the majority of moving objects are pedestrians and their belongings. I used my height as a reference point to calculate the height of pedestrians in Wheeler Place. All samples are scaled to 18×36 pixels with 4 pixels border to retain contour information.

We assumed that the ground at Wheeler Place is flat. This means the height of the camera with respect to any arbitrary point on the ground plan is the same. This assumption adds a small amount of noise to the mapping results when mapping image plan coordinates to real-world coordinates. This describes why the mapped points in Figure 3.17 do not lay on straight lines.

The extracted features were scaled to the range of $[0, 1]$. A five fold cross-validation technique is used to "tune" the classifier hyperparameters. The tuning method, the classifier configurations and the classification technique will be described more precisely in Chapter 5. The LIBSVM [28] and the P-

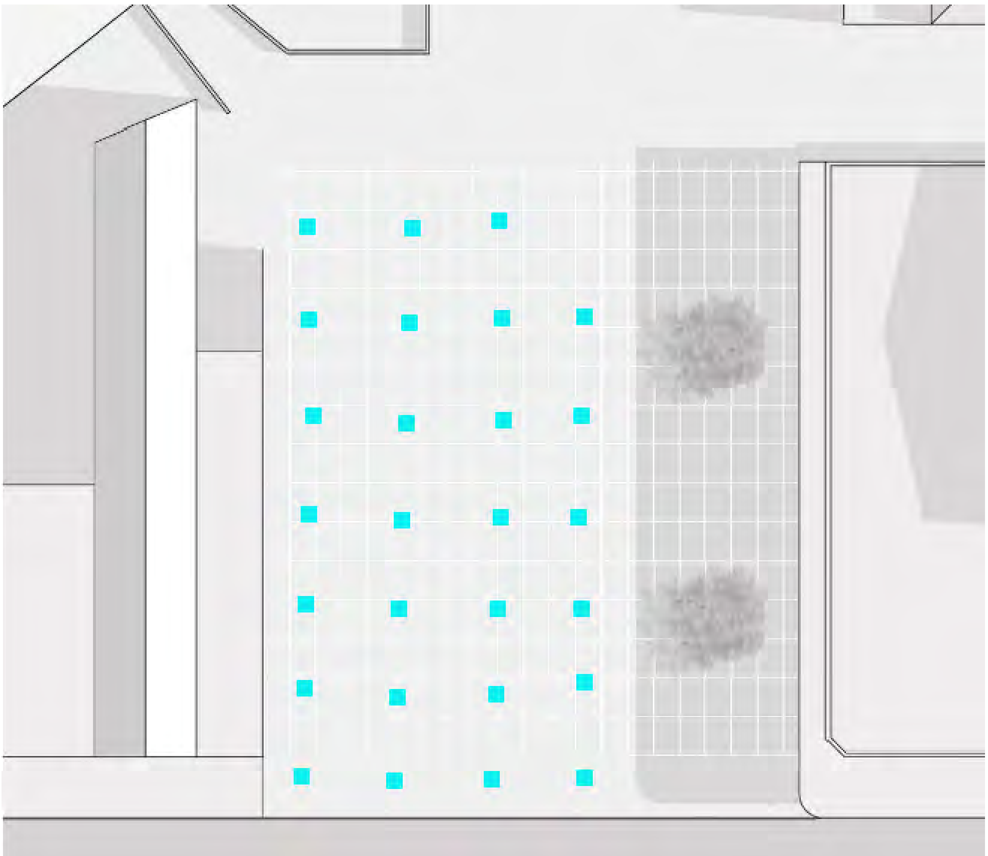

Figure 3.17: The result of mapping of the patterned grid in Wheeler Place to the image plan. The red dots in Figure 3.16 are mapped to blue dots.

Figure 3.18: Wheeler Place

Figure 3.19: Location of the camera

Figure 3.20: Field of view for the camera at Wheeler Place. A pedestrian is looking at the Juicy Bean Cafe.

SVM [79] toolboxes are used for implementing GDTW-P-SVMs. Best values are selected from a generated hyperparameter set to minimise the error rate in the training phase.

The videos were captured during a seven week observation period at Wheeler Place (one day per week). We chose to record pedestrians' behaviour on the same day of every week at the same time (every Thursday from 8am to 5pm). To prepare the training set, pedestrians are labeled manually during their traverse in different frames. As a result, 1917 ROIs are labelled as positive and 212 ROIs are labelled as negative samples with the $0.4s$ sampling rate. This data set is used to train the classifiers against the testing data set, which contains about 77660 positive and negative data samples. Figure 3.21 shows some examples of detected pedestrians at Wheeler Place. 1210 different pedestrians were recognised using the detection system. Among them 830 pedestrians were tracked at Wheeler Place and the rest were outside the area of interest (Wheeler Place), for example walking along Hunter Street.

The attracted trajectories are recognised using our behaviour analysis sys-

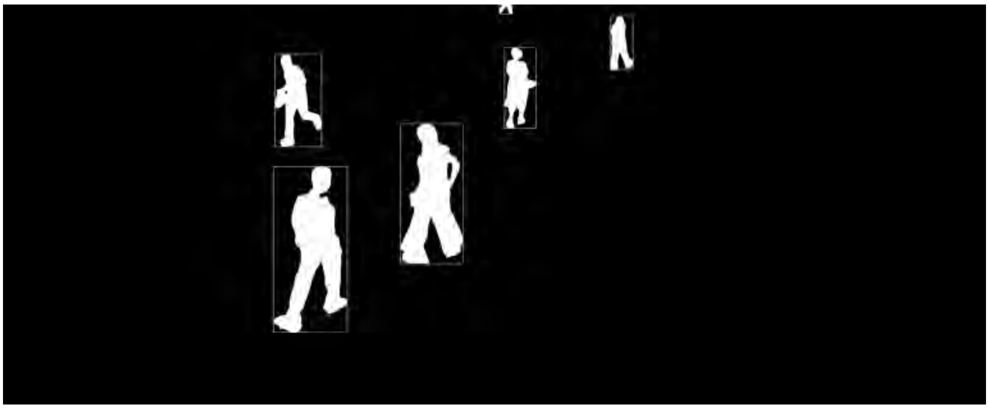

Figure 3.21: Some examples of detected pedestrians using the pedestrian detection and tracking system at Wheeler Place.

tem, which is described in Chapter 5. To balance the number of positive and negative data samples the weight balancing technique suggested by Vapnik is employed [181]. Figure 3.22 shows the trajectories extracted from the video data. Each trajectory is shown in a different colour. The colours are chosen randomly.

Figure 3.23 shows the result of mapping the real-world trajectories to the plan of Wheeler Place. The mapping technique is discussed in Section 3.2.3. In this figure, locations where pedestrians have crossed over more often are shown with darker red colours. As shown in this figure, there are two locations where pedestrians show more interests to go. An opening area that leads to a car, park and the Cafe. The results of analysing the trajectory data are discussed in Chapter 6.

3.4 Summary and Discussion

In this chapter a system for automatic pedestrian detection and tracking at Wheeler Place was presented. The data for trajectory extraction were collected using a single optical camera installed on a fixed platform. The system consists of three parts: *i*) background detection *ii*) pedestrian detection and *iii*)

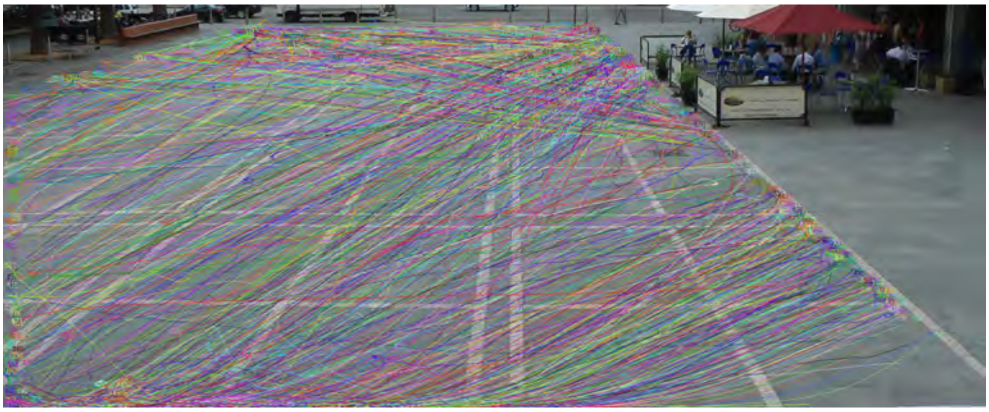

Figure 3.22: Tracked pedestrians using the pedestrian detection and tracking system. Each track is shown in a different colour. The colours are chosen randomly.

pedestrian tracking. In the background detection we employed a background subtraction method with an upgrading strategy. For recognising pedestrians, depth-based, and shape-based features were extracted and used for training the classifiers. The classifiers (GDTW-P-SVMs) were SVM-based and could handle input series with different lengths. More descriptions about the classifier and its trainability will be provided in Chapter 5. Two classifiers were used to recognise pedestrians using the detection of pedestrian leg skeletons and body edges. Pedestrians were tracked using pixel-based, depth-based, and motion-based features.

The presented approach for extracting pedestrians' trajectories has been employed and tested at Wheeler Place. The capability of detecting pedestrians using the discussed system in other urban spaces with environmental features different to those of Wheeler Place is beyond the scope of this PhD research.

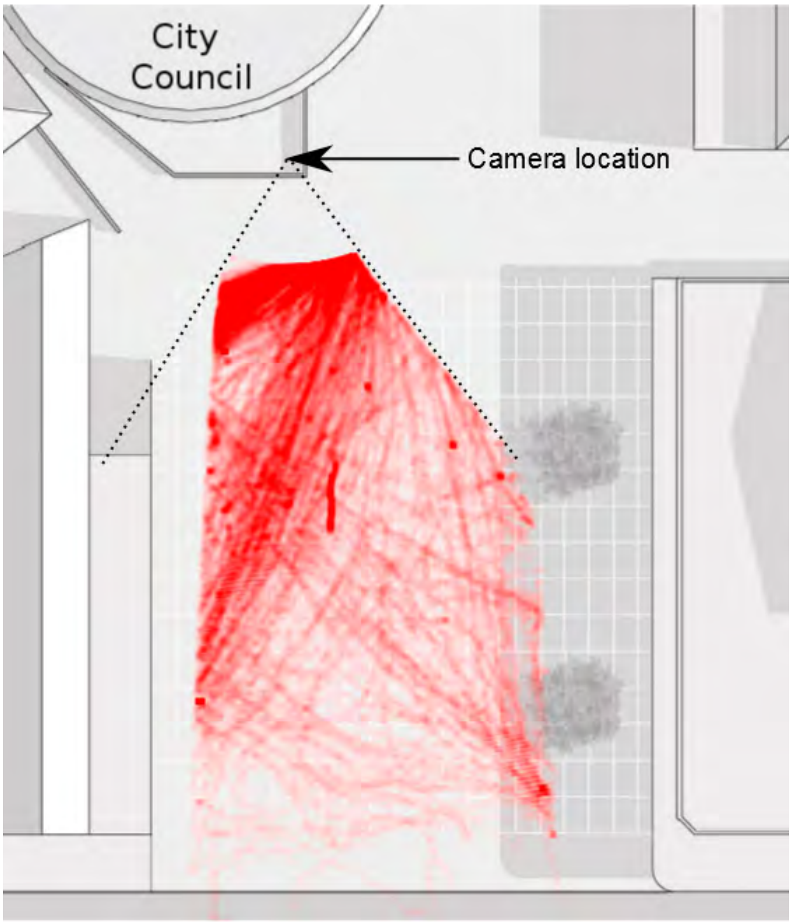

Figure 3.23: Representation of mapping the real-world trajectories to the plan of Wheeler Place. Locations where pedestrians have crossed more often are shown with darker red colours. Dashed lines indicate the field of view of the camera.

Part II

Spatial Behaviour Analysis

Chapter 4

Support vector machines

Contents

This chapter is an introduction to a maximum margin kernel-based classification technique known as Support Vector Machines (SVMs). A variety of support vector classification techniques with a variety of different kernels have been proposed so far [186, 163]. SVMs are learning systems that use a hypothesis space of linear functions in a high dimensional feature space, trained with a learning algorithm from optimisation theory that implements a learning bias derived from statistical theory [38]. This learning strategy introduced by Vapnik and co-workers is a very powerful method that in the few years since its introduction has already outperformed most other systems [181]. They have been employed in a wide variety of applications from time series predication to document analysis to medical and other scientific fields [163].

4.1 Kernel-Based Learning

In supervised learning, the learning technique is given a training set that contains data objects (input space) and their associated labels (classes). To classify a new data object (usually belonging to a test set) a number of sets of hypotheses could be chosen. Among these, linear functions are the simplest to employ [38]. However, complex real-world applications require more expressive hypothesis space than linear functions. Kernel representation is a solution to increase the computational power of linear functions. The kernel maps the data objects from the input space to a high dimensional "feature space", where in the feature space the data objects could be linearly separable.

The difficulty of a learning task depends on the way it is represented. Thus, one common preprocessing step in machine learning involves changing the data representation. This step is equivalent to mapping the input space, $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ to another space, $F = \{\boldsymbol{\phi}(\mathbf{x}) | \mathbf{x} \in X\}$, where $\boldsymbol{\phi}(x)$ is the mapping function and F is called feature space. The values that are used to describe the data is called *feature* and the task of choosing the most suitable representation is known as *feature selection*. Figure 4.1 shows an example of data mapping from non-linearly separable input space to a linearly separable feature space in two dimensions.

The hypothesis space can be expressed as a linear combination of the training points, so that the decision rule can be evaluated using just inner products between the test point and the training points. If we can compute the inner products in feature space directly as a function of original input points, we call such function a kernel function. A *kernel function* is a function K, such that for all $\mathbf{x}, \mathbf{z} \in X$

$$K(\mathbf{x}, \mathbf{z}) = \langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{z}) \rangle \tag{4.1}$$

where $\boldsymbol{\phi}$ is a mapping from X to a feature space. Employing a kernel on the input space makes it possible to map the data to a feature space and train a linear machine in that space [38]. To train a linear classifier using the kernel the only information that is required is the Gram matrix of the training set in the feature space. This matrix is also known as the *kernel matrix* and it is defined as:

$$\mathbf{K} = (K(\mathbf{x}_i, \mathbf{x}_j))_{i,j=1}^{N} \tag{4.2}$$

where N is the number of data objects in the input space. In order to employ a function as the kernel function, it has to satisfy the provided characterisation

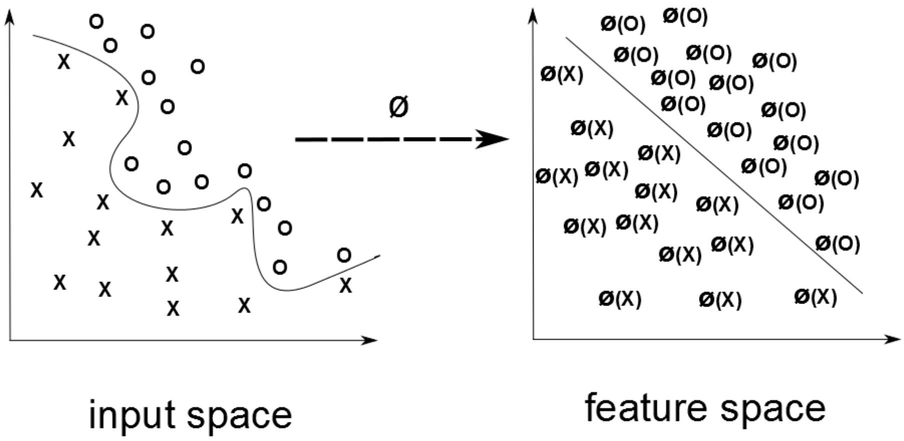

Figure 4.1: Illustration of feature mapping. Data objects in the feature space are linearly separable, which makes the classification task simpler compared to the input space.

in Mercer's theorem [132]:

"Let X be a finite input space with $K(\mathbf{x}, \mathbf{z})$ a symmetric function on X. Then $K(\mathbf{x}, \mathbf{z})$ is a kernel function if and only if its kernel matrix is positive semi-definite."

4.2 Support Vector Classification

Support vector machines (SVMs) are a system for efficiently training linear learning machines in the kernel-induced feature spaces described in the previous section. An important feature of SVMs is that due to Mercer's conditions on the kernels the optimisation problems are convex and hence have no local minima [38]. An important property of SVMs is their simplicity: they are easy to implement and to understand [125].

4.2.1 The Maximum Margin Classifier

A *margin classifier* is a classifier that is able to give an associated distance from the decision boundary, also known as *margin*, for each data object. The margin γ_i with respect to a real-valued decision function, f, of an example (x_i, y_i) is defined to be

$$\gamma_i = y_i f(\mathbf{x}_i) \tag{4.3}$$

The minimum value of the margin with respect to the training set S, that is,

$$min\{y_i f(\mathbf{x}_i) | (\mathbf{x}_i, y_i) \in S\} \tag{4.4}$$

is called *the margin of* f, where

$$S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N) | \mathbf{x}_k \in \mathbb{R}^n, y_k \in \{-1, +1\}\}. \tag{4.5}$$

First, we consider the case of linearly separable data. A training set as defined in Equation 4.5 is called separable by a hyperplane $\mathbf{w}^T\mathbf{x}_k + b = 0$ if there exist both a unit vector $\mathbf{w}$ and a constant b such that the following equalities hold:

$$\mathbf{w}^T\mathbf{x}_k + b \geq +1 | y_k = +1 \tag{4.6}$$

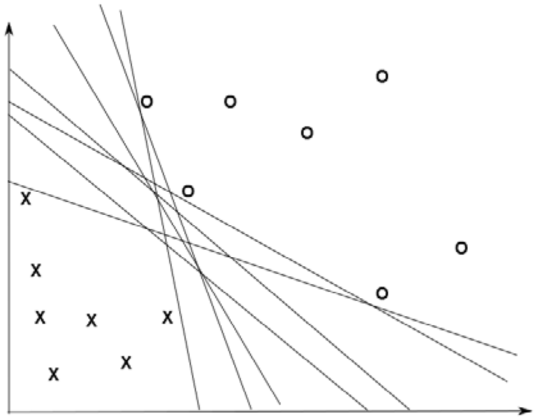

Figure 4.2: Illustration of data classification using a variety of different hyperplanes. Hyperplanes that are too close to the data may increase the chance of false classification.

$$\mathbf{w}^T\mathbf{x}_k + b \leq -1 | y_k = -1 \tag{4.7}$$

The hyperplane is called a *separating hyperplane*.

A hyperplane can separate two classes of data in many possible ways (see Figure 4.2). There is no unique separating hyperplane, unless we add a criterion to decide which is the best or the optimal separating hyperplane. The idea of learning from examples is to recognise the pattern of a class by examining the training points corresponding to that class. New data points are assumed to lie somewhere around the known training data. Therefore, a hyperplane should be chosen such that a small shift of the data does not result in false prediction. If the distance between the separating hyperplane and the training points becomes too small, even test examples very close to the training samples may be classified incorrectly. This will increase the chance of false classification (see Figure 4.3).

Based on this idea, Vapnik and Chervonenkis presumed that the generalisation ability depends on the distance between the hyperplane and the training points. They introduced the *generalised portrait*, a learning algorithm for separable problems, by constructing a hyperplane that maximally separates the

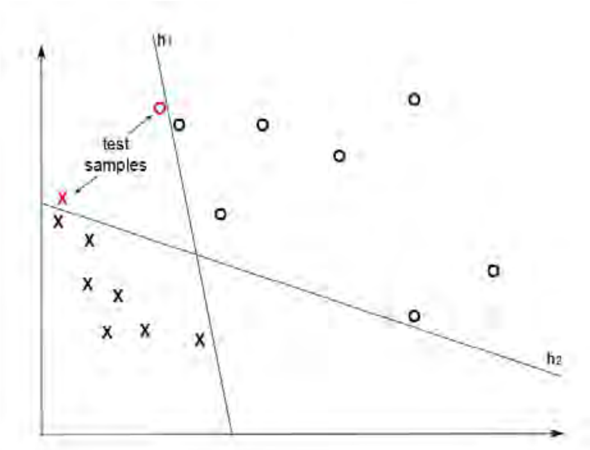

Figure 4.3: Illustration of two false detections using two different hyperplanes. Test samples “red O” and “red X” are miss-classified by the hyperplanes h_1 and h_2 respectively. The hyperplanes are too close to the training data that causes a small *safe margin*.

classes (also known as a classifier with maximum margin) [181]:

$$\max_{\mathbf{w},b} \min \left\{ ||\mathbf{x} - \mathbf{x}_k|| \, |\mathbf{x} \in \mathbb{R}^n, \mathbf{w}^T\mathbf{x} + b = 0, k = 1, \ldots, N \right\} \tag{4.8}$$

The maximal margin classifier is the simplest model of SVMs.

4.2.2 Support Vector Classification

The optimal hyperplane of a training set S is defined by:

$$(\mathbf{w}^*, b^*) = \arg\max_{\mathbf{w},b} \zeta_S(\mathbf{w}, b), \tag{4.9}$$

where $\zeta_S(\mathbf{w}, b)$ is the margin of a set of vectors $S = x_1, \ldots, x_N$ and is defined as:

$$\zeta_S(\mathbf{w}, b) = \min_{\mathbf{x}_k \in S} \zeta_k(\mathbf{w}, b), \tag{4.10}$$

and

$$\zeta_k(\mathbf{w}, b) = y_k(\mathbf{w}^T\mathbf{x} + b). \tag{4.11}$$

Vapnik proves the "uniqueness" of the optimal separating hyperplanes in [181]. To construct the optimal separating hyperplane, the following optimisation problem should be solved

$$\begin{aligned} \max \quad & \zeta_D(\mathbf{w}, b) \\ \textit{subject to} \quad & \zeta_D(\mathbf{w}, b) > 0 \\ & ||\mathbf{w}|| = 1 \end{aligned} \tag{4.12}$$

The problem could be rewritten as

$$\begin{aligned} \min \quad & \tfrac{1}{2} ||\mathbf{w}||^2 \\ \textit{subject to} \quad & \mathbf{w}^T\mathbf{x} + b \geq +1 \quad \textit{for } y_k = +1 \\ & \mathbf{w}^T\mathbf{x} + b \leq -1 \quad \textit{for } y_k = -1 \end{aligned} \tag{4.13}$$

The solutions for Equation 4.13, $\mathbf{w}_0$, and Equation 4.12, $\mathbf{w}^*$, are related as [181]:

$$\mathbf{w}^* = \frac{\mathbf{w}_0}{||\mathbf{w}_0||}. \tag{4.14}$$

The Lagrangian for Equation 4.13 is

$$\mathcal{L}(\mathbf{w}, b, e; \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{k=1}^{N} \alpha_k \left\{ y_k \left[\mathbf{w}^T\mathbf{x}_k + b \right] - 1 \right\} \tag{4.15}$$

where $\alpha_k \geq 0$ are the Lagrange multipliers for $k = 1, \ldots, N$. The solution is characterised by the saddle point of the Lagrangian,

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial \mathbf{w}} = 0 \quad \longrightarrow \quad \mathbf{w} = \sum_{k=1}^{N} \alpha_k y_k \mathbf{x}_k \\ \\ \frac{\partial \mathcal{L}}{\partial b} = 0 \quad \longrightarrow \quad \sum_{k=1}^{N} \alpha_k y_k = 0 \end{cases} \tag{4.16}$$

which results in the following classifier

$$y(\mathbf{x}) = \textit{sign} \left[\sum_{k=1}^{N} \alpha_k y_k \mathbf{x}_k^T \mathbf{x} + b \right]. \tag{4.17}$$

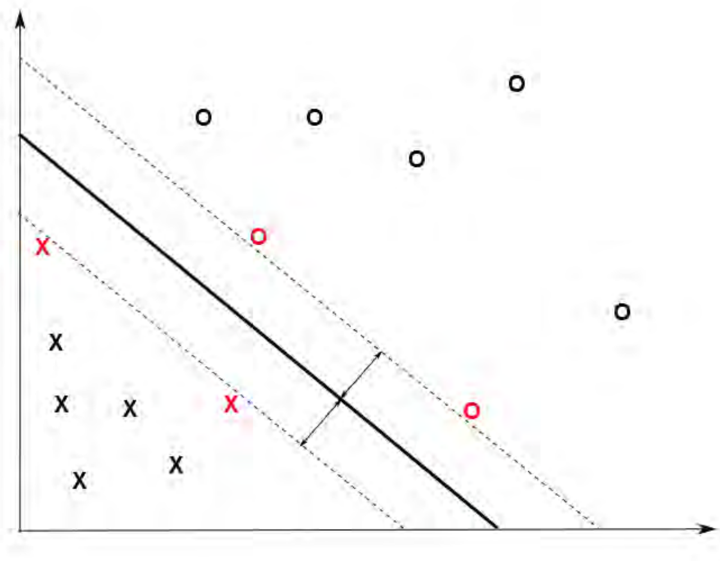

Figure 4.4: Illustration of a separating hyperplane with maximum margin. Support vectors are shown in red.

Equation 4.16 and Equation 4.15 result in the following Quadratic Programing (QP) problem

$$\begin{aligned} \max_{\alpha} \xi_S(\alpha) = & \quad -\tfrac{1}{2} \textstyle\sum_{k,l=1}^{N} y_k y_l \mathbf{x}_k^T \mathbf{x}_l \alpha_k \alpha_l + \sum_{k=1}^{N} \alpha_k \\ \textit{such that} & \qquad \textstyle\sum_{k=1}^{N} \alpha_k y_k = 0 \end{aligned} \tag{4.18}$$

Many of the resulting α_k values are equal to zero. This means that in the resulting classifier the sum should be taken only over the non-zero α_k values (support vectors) instead of all training data points:

$$y(\mathbf{x}) = sign\left[\sum_{k=1}^{\#SV} \alpha_k y_k \mathbf{x}_k^T \mathbf{x} + b\right], \tag{4.19}$$

where $\#SV$ is the number of support vectors. These support vectors are close to the separating hyperplane (Figure 4.4).

For nonlinear problems the "kernel trick" is employed. As a result the dual problem becomes:

$$\begin{aligned} \max_{\alpha} \xi_S(\alpha) = \quad & -\tfrac{1}{2}\textstyle\sum_{k,l=1}^{N} y_k y_l K(\mathbf{x}_k, \mathbf{x}_l)\alpha_k \alpha_l + \sum_{k=1}^{N} \alpha_k \\ \textit{such that} \quad & \textstyle\sum_{k=1}^{N} \alpha_k y_k = 0 \\ & 0 \leq \alpha_k \leq c, k = 1, \ldots, N. \end{aligned} \tag{4.20}$$

Finally the nonlinear SVM classifier becomes:

$$y(\mathbf{x}) = sign\left[\sum_{k=1}^{N} \alpha_k y_k K(\mathbf{x}, \mathbf{x}_k) + b\right]. \tag{4.21}$$

The bias, b, is obtained using Karush-Kuhn-Tucker conditions and the following equation with any data from the training set.

$$y_k\left[\mathbf{w}^T \phi(\mathbf{x}_k) + b\right] - 1 = 0 \ for \ \alpha_k \in (0, c) \tag{4.22}$$

4.2.3 Kernels in Support Vector Machines

The solution to the convex QP problem is again global and unique provided that one chooses a positive definite kernel for $K(.,.)$. This choice guarantees that the matrix involved in the QP problem is positive definite as well, and the kernel trick is applicable. For a positive semi-definite kernel the solution to the QP problem is global but not necessarily unique [181].

Four common choices of kernels are:

1. *Linear Function*: $K(x, z) = x^T z$
2. *Polynomial Function*: $K(x, z) = (\tau + x^T z)^d$, results in polynomial decision function. Mercer's condition holds for all τ.
3. *Radial Basis Function*: $K(x, z) = \exp(-\left\|x - z\right\|_2^2 / \sigma^2)$, gives a Gaussian radial basis function (RBF) classifier. Mercer's condition holds for all σ values. The number of support vectors, the support vectors, the weights, and bias are all produced automatically by the SVM training and give excellent results compared to classical RBF [165].

4. *Neural Networks*[1]: $K(x, z) = \tanh(k_1 x^T z + k_2)$, gives a particular kind of two-layer sigmoid network. The architecture of the network is determined by SVM training. Mercer's condition does not hold for all possible choices of k_1 and k_2 [166].

4.3 Potential Support Vector Machines

Potential support vector machines (P-SVMs) have been proposed by Hochreiter and Obermayer to analyse *dyadic data* where two sets of objects (row and column objects) are characterised by a matrix of numerical values [79]. It is a maximum margin method for construction of classifiers and regression functions for the column objects in a data matrix. They defined a form of representation of data and called it dyadic data. In this form the whole data set can be represented using a rectangular matrix whose entries denote the relationships between the corresponding "row" and "column" objects. Pairwise data representations as a special case of dyadic data can be found for data sets where similarities or distances between objects are measured.

Traditionally, "row" objects have been called "features" and "column" objects have been called "feature vector". To apply SVMs for classification, the data matrix is interpreted as a Gram matrix. In the case of pairwise data the Gram matrix is often symmetric. As discussed in the previous subsection the Gram matrix should be positive semi-definite to obtain a global solution for the QP. To make the SVMs capable of handling non-positive semi-definite data matrices, authors in [79] have suggested to consider column and row objects on an equal footing and interpret the matrix entries as the result of kernel function. This takes a row object, applies it to a column object and outputs a number. As a result, the P-SVM can handle rectangular matrices as well as pairwise data whose matrices are not necessarily positive semi-definite.

P-SVMs have been defined based on a scale-invariant objective function and a new set of constraints. In this subsection the P-SVM classification technique is described. The technique will be used in Chapter 5 to introduce the proposed method for analysing variable-length input series.

[1] Also known as Multilayer Perceptron (MLP)

Consider a set $\{x^i| \leq i \leq L\}$ of objects that are described by feature vectors $\mathbf{x}^i \in \mathbb{R}^N$ and that form a training set $\mathbf{X}_\phi = \left\{\mathbf{x}_\phi^1, \ldots, \mathbf{x}_\phi^L\right\}$. The vectors $\mathbf{x}_\phi^i$ are images of a map ϕ, which is induced either by a kernel or by a measurement function. Both the selection of a classifier using the maximum margin principle and the values obtained for the generalisation error bounds suffer from the problem that they are not invariant under linear transformations [79]. Thus the question arises, which scale factors should be used for classifier selection? The suggested approach by [79] is to scale the training data such that the margin remains constant while the radius of the sphere containing all training data in the feature space, R, becomes as small as possible. This leads to the following objective function, which is an upper-bound for the radius of the sphere containing the scaled data, R^*.

$$\mathbf{w}^T \mathbf{X}_\phi \mathbf{X}_\phi^T \mathbf{w} = \left|\left|\mathbf{X}_\phi^T \mathbf{w}\right|\right|^2 \tag{4.23}$$

The data objects are projected to all P directions of the feature space to make the "complex features". The K_{ij} of such complex feature $\mathbf{z}_w^j$ for an object $\mathbf{x}_\phi^i$ is then given by the dot product

$$K_{ij} = \langle \mathbf{x}_\phi^i, \mathbf{z}_w^j \rangle. \tag{4.24}$$

Let $\mathbf{Z}_w := \left(\mathbf{z}_w^1, \mathbf{z}_w^2, \ldots, \mathbf{z}_w^P\right)$ be the matrix of all complex features. Then we can summarise our knowledge about the objects in X_ϕ using the data matrix

$$\mathbf{K} = \mathbf{X}_\phi^T \mathbf{Z}_w. \tag{4.25}$$

The following quadratic loss function has been defined as a quality measure for the performance of the classifier on the training set.

$$c(y_i, f((x)_\phi^i)) = \frac{1}{2} r_i^2, \tag{4.26}$$

$$r_i = f(\mathbf{x}_\phi^i) - y_i = \langle \mathbf{w}, \mathbf{x}_\phi^i \rangle + b - y_i.$$

Then the mean squared error is

$$R_{emp}\left[f_{\mathbf{w}}, b\right] = \frac{1}{L}\sum_{i=1}^{L} c(y_i, f(\mathbf{x}_\phi^i)). \tag{4.27}$$

The classifier should minimise the R_{emb}, i.e. that

$$\nabla_w R_{emp}\left[f_{\mathbf{w}}, b\right] = \frac{1}{L}\mathbf{X}_\phi \left(\mathbf{X}_\phi^T \mathbf{w} + b\mathbf{1} - \mathbf{y}\right) = 0, \tag{4.28}$$

and

$$\frac{\partial R_{emp}[f]}{\partial b} = \frac{1}{L}\sum_i r_i = b + \frac{1}{L}\sum_i \left(\langle \mathbf{w}, \mathbf{x}_\phi^i\rangle - y_i\right) = 0, \tag{4.29}$$

where the labels for all objects in the training set are summarized by a label vector $\mathbf{y}$. Condition Equation 4.29 implies that the directional derivative should be zero along any direction in feature space, including the direction of the complex feature vectors $\mathbf{z}_w$. This defines the constraints for the classifier.

Based on the objective function (Equation 4.23) and the constraints (Equation 4.29) the P-SVM optimisation problem for classification is defined as

$$\min_{\boldsymbol{\alpha}} \quad \tfrac{1}{2}\boldsymbol{\alpha}^T \mathbf{K}^T \mathbf{K}\boldsymbol{\alpha} - \mathbf{y}^T \mathbf{K}\boldsymbol{\alpha} \tag{4.30}$$

$$\textit{subject to} \quad -C\mathbf{1} \leq \boldsymbol{\alpha} \leq C\mathbf{1}, \tag{4.31}$$

and the P-SVM classification function is

$$f\left(\mathbf{x}_\phi\right) = \sum_{j=1}^{P} \alpha_j K_{(x)j} + b, \tag{4.32}$$

where

$$b = \frac{1}{L}\sum_{i=1}^{L} y_i. \tag{4.33}$$

One of the most crucial properties of the P-SVM procedure is that the dual optimisation problem only depends on $\mathbf{K}$ via $\mathbf{K}^T\mathbf{K}$. Therefore, $\mathbf{K}$ is neither required to be positive semi-definite nor to be square. This allows not only the construction of SVM-based classifiers for matrices $\mathbf{K}$ of general shape but also to extend SVM-based approaches to the class of indefinite kernels operating on the objects' feature vectors [79].

4.4 Summary

Support Vector Machines classification is a method of calculating the optimal separating hyperplane in the feature space. The optimal separating hyperplane is defined as the maximum-margin hyperplane in the higher dimensional feature space. The use of the maximum-margin hyperplane is motivated by statistical learning theory, which provides a probabilistic test error bound that is minimized when the margin is maximised.

The original SVM was a linear classifier. However, Vapnik suggested using the kernel trick. In the kernel trick, each dot product used in a linear algorithm is replaced with a non-linear kernel function. This causes the linear algorithm to operate in a different space. For SVMs, using the kernel trick makes the maximum margin hyperplane fit in a feature space. The feature space is a non-linear map from the original input space, usually of much higher dimensionality than the original input space. In this way, non-linear SVMs can be created. If the kernel used is a radial basis function (RBF), the corresponding feature space is a Hilbert space of infinite dimension. To use the SVM as the classifier the kernel employed to map the input space to feature space should be positive semi-definite (PSD).

Potential support vector machines (P-SVMs) select models using the principle of structural risk minimisation. In contrast to standard SVM approaches, however, the P-SVM is based on a new objective function and a new set of constraints, which lead to an expansion of the classification or regression function in terms of "support features". The optimisation problem is quadratic, always well defined, suited for dyadic data, and neither requires square nor positive definite Gram matrices. The P-SVM suggested a interpretation of dyadic data, where objects in the real-world are not described by vectors. Structures like dot products are induced directly through measurements of object pairs, i.e. through relations between objects.

The concepts described in this chapter will be used in the next chapter to present the proposed classification technique. The technique will be employed to classify behavioural data entries with variable length. The data extraction method has been explained in the previous chapters.

Chapter 5

Variable-length input series analysis

The content of this chapter has been published in [88].

Contents

This chapter describes a new technique for sequential data analysis where each data object is characterised by a series of numerical values that may have different lengths for different data objects. The new technique, called GDTW-P-SVMs, is a maximum margin method for the construction of classifiers with variable-length input series. It employs potential support vector machines (P-SVMs) and Gaussian dynamic time warping (GDTW) to waive the fixed-length restriction of feature vectors in training and test data. As a result, GDTW-P-SVMs enjoy the P-SVM method's properties such as the ability to: i) handle data and kernel matrices that are neither positive definite nor square and ii) minimise a scale-invariant capacity measure. The new technique elaborates on the P-SVM kernel functions, by utilising the well-known DTW algorithm to provide an elastic distance measure for the kernel functions. Benchmarks for classification are performed with several real-world data sets from the UCR Time Series Classification/Clustering page, the GeoLife trajectory data set, and the UCI Machine Learning Repository. The data sets include data with both variable and fixed-length input series. The results show that in the case of fixed-length feature vectors, the new method often performs as good as or even better than the benchmarked standard methods and it is able to classify data sets with variable-length input series significantly better than existing methods.

5.1 Previous Works

Within the context of time series analysis, sequential data classification has received great interest during the last decade. It has been widely applied to various research areas such as financial data mining [184, 194, 121], moving object identification [95, 196], medical data analysis [183, 124, 39], trajectory data analysis [90, 92, 91], time-stamped event data processing [93], and network monitoring [202, 151]. In all applications of sequential data classification using a kernel-based learning approach the data are represented in a new space by a similarity/distance measure. In the new space the data are aligned such that similar features correspond to each other. The features that represent the same property of the data are called "matching features/time stamps". Different representations of the same sequential data could lead to different matching feature sets. This makes "sequential pattern matching", which includes comparing sequences of features for the presence of some pattern, a challenging problem.

Many distance measures have been proposed to solve the above-mentioned problem [84, 85, 98, 130]. In sequential pattern matching two types of distance measures have been employed: "elastic" [43] and "metric" [36] distance measures. According to these two types, one could group distance measures into three categories [130]:

1. Non-elastic metric (Euclidean Distance, l_p-norms and Correlation [43])
2. Elastic non-metric (Dynamic Time Warping [182] and Longest Common Sub-sequence [34])
3. Elastic metric (Edit Distance with Real Penalty [29])

Metric distance measures satisfy the "triangle inequality"[1]. This condition makes possible the efficient pruning of large numbers of time series that deviate too far from a matching pattern [96, 130]. Comprehensive applications have shown that Dynamic Time Warping (DTW) [110, 182] among elastic non-metric distances and the Euclidean Distance among non-elastic metric distances outperformed most of the other distance measures [108, 149, 185, 145, 96].

DTW is a dynamic programming algorithm for measuring the distance between two sets of sequential data, which can be turned into a linear representation in time space (Figure 5.1). DTW has initially been proposed and used in automatic speech recognition [182]. It aims to align two sequences of input series by warping the time axis iteratively until an optimal match between the two sequences is found.

Figure 5.2 shows a warping path for the two sequences in Figure 5.1 obtained using DTW. To find the best match or alignment between the two sequences one needs to find a path through a grid (Figure 5.2). Whenever the path moves horizontally/vertically only, it means that several points from the first/second sequence correspond to one point in the second/first sequence. The path minimises the total distance between the two sequences. DTW finds the minimum matching path by providing non-linear alignments between two sequences.

[1] $d(x, y) \leq d(x, z) + d(z, y)$ ("triangle inequality").

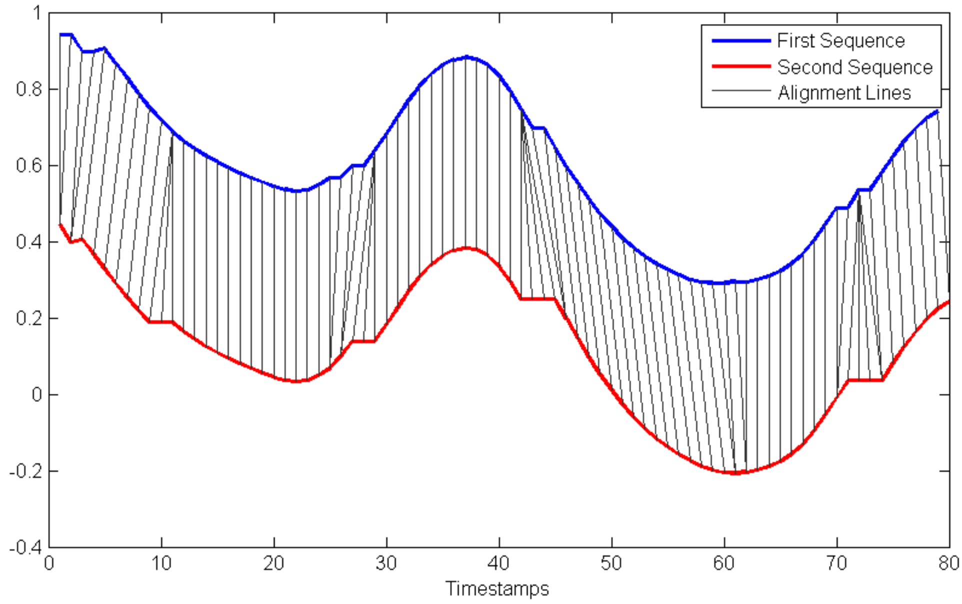

Figure 5.1: Sequence alignment using Dynamic Time Warping [182]; Illustration of comparing points in two sequences using Dynamic Time Warping. As shown, the two sequences have two different lengths. Finding the correspondence between data points has made DTW capable of comparing data objects with different lengths.

DTW is capable of elastic and robust sequential data matching, and it tolerates variable sequence length, which is common in sequential patterns (for example movement trajectories). Sequences are "warped/stretched" non-linearly in the time dimension to determine a measure of their similarity independent of certain non-linear variations in the time dimension. DTW has been widely used as a distance measure for time series classification and clustering. A variety of the DTW algorithms have been proposed for applications such as weighted dynamic time warping [96], derivative DTW [110], multidimensional DTW [192], DTW for pitch determination [51], scaling up DTW [109] and optimised DTW [73]. However, DTW does not account for the relative importance regarding the phase difference between a reference point and a testing point [96]. This may lead to misclassification, especially in applications where the shape similarity between two sequences is a major consideration for accurate recognition [96]. All of the above mentioned algorithms have employed DTW without using a learning algorithm. As it will be shown, combining DTW with a learning algorithm helps to perform the classification of the sequential data with higher accuracy and overcomes DTW problems.

Support vector machines (SVMs), on the other hand, have become a popular approach to pattern classification since they can deliver state-of-the-art performance on a wide variety of real-world classification problems [172, 181, 167, 120]. Many interesting kernels have been proposed for sequential data classification using SVMs [39, 144, 126, 36]. Mutual information kernels have been proposed for a special case of sequential string classification [39, 169]. These kernels are able to solve classification problems in high dimensional space where labelled data are sparse and unlabelled data are abundant [169]. Context-free models or probabilistic suffix tree structures have been employed to construct these kernels for an application of protein classification [39]. For solving general classification problems for variable length data objects, it is very tempting to plug the sequential distance measures into SVM kernels such as the Gaussian kernel [36].

A so-called Gaussian dynamic time warping (GDTW) kernel has been proposed for sequential data classification with applications in online handwriting recognition and speech recognition [8, 175]. The kernel is defined as

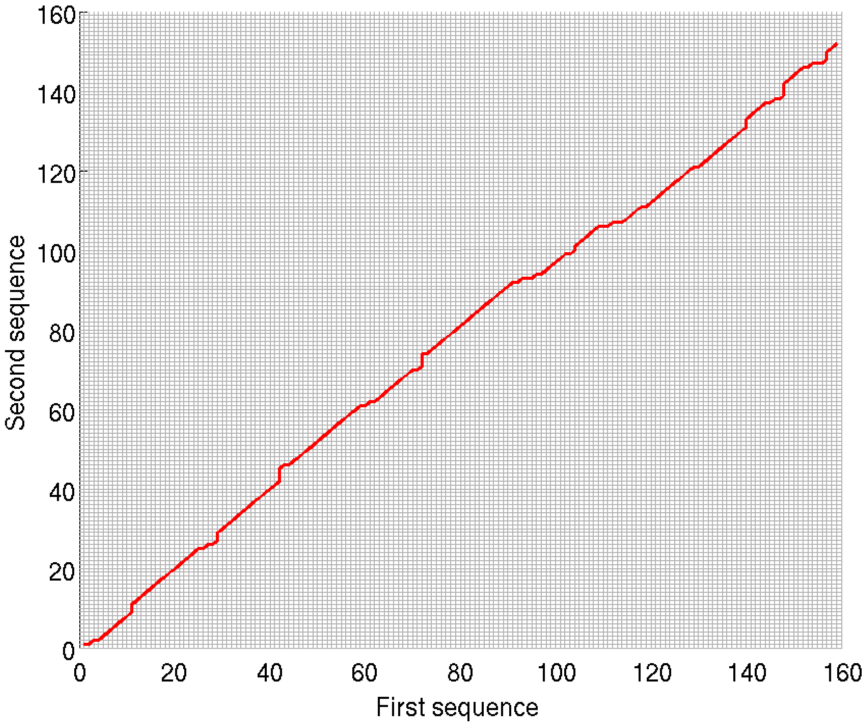

Figure 5.2: Illustration of warping path for the sequences shown in Figure 5.1 obtained using Dynamic Time Warping [182]; DTW finds the minimum matching path by providing non-linear alignments between two sequences. As shown, the path has not been ended at point (160, 160) since the two sequences have two different lengths. In the path, pure vertical movements show that one point in the first sequence is corresponding to several points in the second sequence. Likewise, pure horizontal movements show that one point in the second sequence is corresponding with several points in the first sequence. Finding the correspondence between data points has made DTW capable of comparing data objects with different lengths.

$K(x, y) = exp(-\frac{D(x,y)}{\sigma^2})$, where $D(x, y)$ denotes the DTW distance. Authors in [119] have shown that the GDTW is not a positive semi-definite symmetric function [36] and therefore it does not satisfy the kernel characterisation required by Mercer's Theorem [132]. So GDTW is not a qualified kernel for SVMs, the existence of a "Reproducing Kernel Hilbert Space" [36] is not guaranteed and it is no longer clear what it is that is being optimised [168]. In the following sections it is shown that in general cases it is not clear whether GDTW satisfies Mercer's condition or not. Several approaches proved that GDTW kernels are positive semi-definite under some favorable conditions and they can be tuned effectively for speech recognition [40], but not in the general case of sequential pattern matching.

Recently Sepp Hochreiter and Klaus Obermayer [79] have proposed a technique for the analysis of dyadic data where two sets of objects are characterised by a matrix of numerical values that describe their mutual relationship [79]. They called their method *Potential Support Vector Machines* (P-SVMs). Contrary to standard SVM approaches, the P-SVMs lead to a sparse expansion of the classification and regression functions in terms of the row rather than the column objects and can handle data and kernels that are neither positive definite nor square. Hochreiter and Obermayer have shown that the P-SVMs often perform better than the standard approaches. This provides a good opportunity to cope with the optimisation problem of the GDTW kernels.

In this chapter the advantages and disadvantages of several classification methods for variable-length input series are discussed and a new method is proposed to avoid the shortcomings of others. The feasibility of usage of the GDTW as a Mercer kernel [132] in standard SVMs is discussed. The new method, called GDTW-P-SVMs, utilises the GDTW kernel in P-SVMs for the first time. It benefits from the robustness of DTW as an elastic distance measure in finding similarities between two variable-length data sequences. This makes it a powerful tool to classify trajectory-based data sets such as human spatial behavioural characteristics and animal movement trajectories for behavioural analysis, pen movement trajectories for document analysis and any other collected data for movement-based analysis that can be represented as sequences of time stamped locations.

5.2 Background

5.2.1 Support Vector Machines

Suppose the space X, containing the input data, is referred to as the "input space", while $F = \{\phi(\mathbf{x}) : \mathbf{x} \in X\}$ is an inner product space (Hilbert Space) and is called the "feature space", where $\phi : X \longrightarrow F$ is the feature map from X to F. A *kernel* function [38] is a function $K : X \times X \longrightarrow \mathbb{R}$, such that for all $\mathbf{x}, \mathbf{z} \in X$

$$K(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle \tag{5.1}$$

One of the most common kernel functions is the "Gaussian kernel", which is defined as:

$$K(\mathbf{x}, \mathbf{z}) = \exp\left(-\gamma \, ||\mathbf{x} - \mathbf{z}||^2\right) \tag{5.2}$$

where $\gamma > 0$ is a user-specified shape parameter. Consider classifying a training sample $S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l))$, using the feature space implicitly defined by the kernel $K(\mathbf{x}, \mathbf{y})$ and suppose the parameters α^* solve the following quadratic optimisation problem:

$$\begin{aligned}
&\textit{maximise} \quad W(\boldsymbol{\alpha}) = \textstyle\sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j), \\
&\textit{subject to} \qquad\qquad \textstyle\sum_{i=1}^{l} y_i \alpha_i = 0, \\
&\qquad\qquad\qquad C \geq \alpha_i \geq 0, i = 1, \ldots, l,
\end{aligned} \tag{5.3}$$

where l is the number of training samples, y_i is the label for the ith training sample and C is a real parameter, which is varied through a wide range of values while the optimal performance is assessed using a separate validation set by cross-validation [38]. Let $f(\mathbf{x}) = \sum_{i=1}^{l} y_i \alpha_i^* K(\mathbf{x}_i, \mathbf{x}) + b^*$, where b^* is chosen so that $y_i f(\mathbf{x}_i) = 1$ for any i with $C > \alpha_i^* > 0$. Then the decision rule given by $sgn(f(\mathbf{x}))$ is equivalent to the hyperplane in feature space implicitly defined by the kernel $K(\mathbf{x}, \mathbf{z})$ that solves the optimisation problem (Equation 5.3), where b^* is chosen using the Karush-Kuhn-Tucker conditions [113].

Let $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be a finite input space with $K(\mathbf{x}_i, \mathbf{x}_j)$ a symmetric function on X. Then $K(\mathbf{x}_i, \mathbf{x}_j)$ is a "Mercer kernel" if the matrix

$$\mathbf{K} = (K(\mathbf{x}_i, \mathbf{x}_j))_{i,j=1}^{n} \tag{5.4}$$

is positive semi-definite (PSD)[2] [132, 156]. Three approaches for data classification using SVMs have been proposed:

1. If the data matrix (Equation 5.4) is PSD, it is interpreted as a *Gram matrix* and SVMs are subsequently applied [1].

2. If the data matrix is indefinite but symmetric, the matrix is projected into a subspace spanned by the eigenvectors with positive eigenvalues [67].

3. Another approach for dealing with indefinite data matrices involves flipping the sign of negative eigenvalues [79].

All three approaches guarantee a PSD matrix on the available training set but it may not be PSD on the new test set.

5.2.2 SVMs with GDTW Kernel

The Gaussian function with Euclidean distance measure (Equation 5.2) is among the most commonly used kernels in SVMs. It is well-known that the Gaussian function provides a Mercer kernel [164]. It maps n vectors $v_1, v_2, \ldots, v_n$ into a Hilbert space [36] where $\phi(v_1), \phi(v_2), \ldots, \phi(v_n)$ span an n-dimensional subspace [164]. The Euclidean distance measure used in Equation 5.2 is able to compare two vectors with the same length only. Therefore, the classifier that is using this kernel function is restricted to an input space with fixed-length feature vectors. For instance, sequential data that vary in speed or time (such as pedestrian trajectory data and human voice data with variable recording times) cannot be directly used as the input space for this kernel function.

[2] A positive semi-definite matrix is a Hermitian matrix, all of whose eigenvalues are non-negative [129, 38].

In order to overcome this problem, a Gaussian function can be defined with a DTW distance measure [8, 175]:

$$k_{GDTW}(\mathbf{x}^r, \mathbf{y}^s) = \exp\left(-\frac{D(\mathbf{x}^r, \mathbf{y}^s)}{\sigma^2}\right) \tag{5.5}$$

where $\mathbf{x}^r$ is a time series with discrete time index varying between 1 and r, $\mathbf{y}^s$ is a time series with discrete time index varying between 1 and s, σ is the Gaussian kernel width, and $D(\mathbf{x}^r, \mathbf{y}^s)$ is the DTW distance. It can be calculated recursively as:

$$D(\mathbf{x}^r, \mathbf{y}^s) = ||x_r - y_s||_p + min \begin{cases} D(\mathbf{x}^{r-1}, \mathbf{y}^s) & delete, \\ D(\mathbf{x}^{r-1}, \mathbf{y}^{s-1}) & match, \\ D(\mathbf{x}^r, \mathbf{y}^{s-1}) & insert, \end{cases} \tag{5.6}$$

where $x_r \in \mathbb{R}^d$ is the rth element (last element) of time series $\mathbf{x}^r$, $y_s \in \mathbb{R}^d$ is the sth element (last element) of time series $\mathbf{y}^s$, and $||x_r - y_s||_p$ is the l_p-norm in $\mathbb{R}^d$. For further information about DTW algorithms and their variations please refer to [130, 182, 160].

5.2.3 Two-Step DTW-SVM Classifier

The effective use of SVMs in classification necessitates the appropriate choice of a kernel. Classifying data sets that contain variable-length input series requires the designing problem specific kernels. This involves the definition of a similarity measure, with the condition that the kernels are PSD. An alternative technique is discussed here, which uses a two-step architecture for classifying the data.

In the first step of the classification technique, the data has been represented by the DTW distance measure. DTW is able to find the distance between two input series with different lengths. In the representation each sample is represented by its DTW distances to other data samples. This is shown as a matrix in Figure 5.3 and we call it the *DTW matrix*. In this matrix each row/column represents a transformed data sample. Since $DTW(\mathbf{x}_i^{r_i}, \mathbf{x}_j^{r_j}) = DTW(\mathbf{x}_j^{r_j}, \mathbf{x}_i^{r_i})$, the matrix is symmetric.

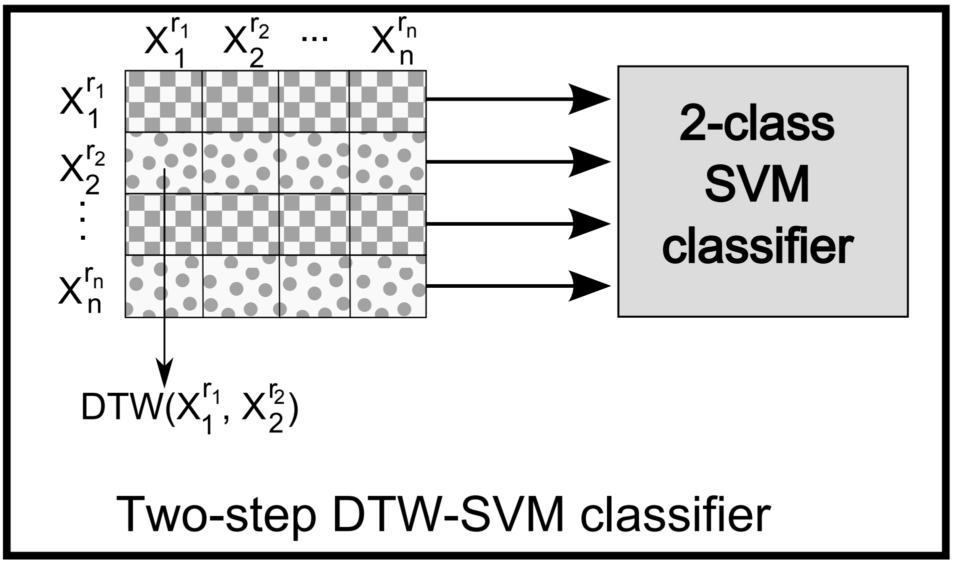

Figure 5.3: Two-step DTW-SVM methodology for classifying two classes of data with variable-length input series. It is used along with the pairwise classification method for multi-class classification problems. $\mathbf{x}_1^{r_1}, \mathbf{x}_2^{r_2}, \ldots, \mathbf{x}_n^{r_n}$ are data samples with different lengths, and $DTW(\mathbf{x}_1^{r_1}, \mathbf{x}_2^{r_2})$ is the DTW distance between $\mathbf{x}_1^{r_1}$ and $\mathbf{x}_2^{r_2}$.

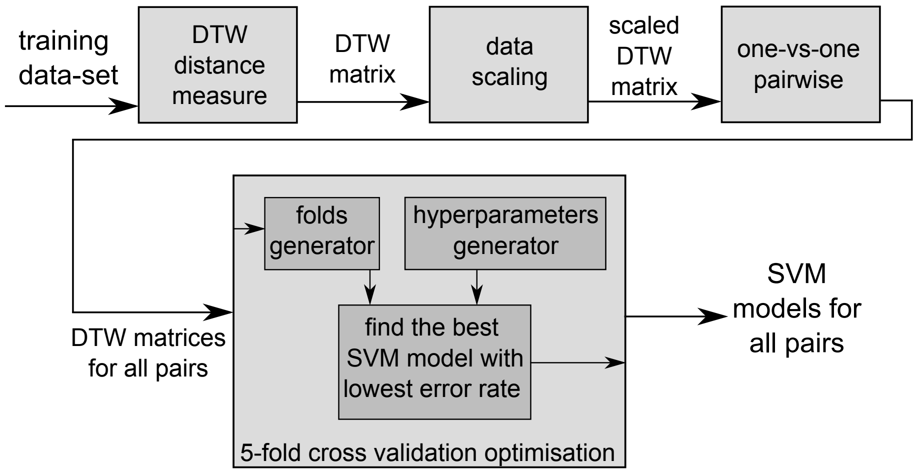

Figure 5.4: Training phase for multi-class classification problems using the idea presented in Figure 5.3; the pairwise classification algorithm (one-vs-one) is employed to train multi-class data, 5-fold cross validation with a leave-one-out policy is utilised for tuning hyperparameters (C, γ and kernel parameters).

In the second step the DTW matrix is used as the input of a standard two-class SVM classifier (as shown in Figure 5.3). It is able to distinguish between two classes only. This technique can be used along with the pairwise classification algorithm to classify multi-class time series.

Figure 5.4 shows different stages in the "training phase" of the multi-class classification method using the two-step DTW-SVMs classifier and a pairwise one-vs-one algorithm. A 5-fold cross validation technique is also employed to "tune" the SVM hyperparameters (C, γ and kernel parameters). As shown, after calculating the DTW distances between all samples the distances are scaled in $[0, 1]$ (The scaling method is described in Section 5.4, Equation 5.18). Then pairs are created from the scaled DTW matrix and class labels that are provided in the training set. Each pair contains two classes of data only (one-vs-one pairwise algorithm). Afterwards, the 5-fold cross validation technique is utilised to tune the hyperparameters. As a result of tuning, one SVM model for each pair of classes will be constructed.

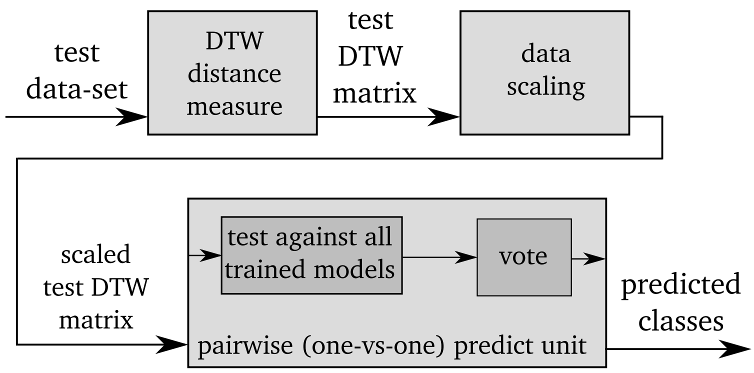

Figure 5.5: Testing phase for trained models obtained using presented architecture in Figure 5.4; a pairwise classification algorithm (one-vs-one) is employed to label multi-class data.

The "testing phase" in this classification technique, as shown in Figure 5.5, is different from the testing phase in commonly used classification methods. In the training phase all training data are used to obtain the DTW matrix and represent the input space for the SVM classifier. In the testing phase each testing object has to be mapped with the same representation as was used in the training phase. This means to classify a test sample, the distances between the test sample and all samples in the training set have to be calculated. The matrix that contains these distances is called the "test DTW matrix" in Figure 5.5. The i^{th} row of the matrix represents the DTW distances between the i^{th} test sample and all samples in the training set.

This technique overcomes the problem of classifying data samples with different lengths using SVMs, and it enjoys the benefits of using the DTW distance measure without suffering from employing non-PSD kernels in SVMs. However, in the testing phase, to calculate the test DTW matrix (as shown in Figure 5.5) the training set as well as the trained models are required for classification. This makes it difficult to distribute trained data, either because they are too big to distribute or in some cases the training data sets are not allowed to be accessed by a testing party. Also, the DTW distances between each test sample and all training samples need to be calculated, which obviously slows down the testing process. Therefore the method is not technically feasible for applications that have a big training set or require real-time classification.

5.3 Proposal of GDTW-P-SVMs

5.3.1 Positive Semi-Definiteness and the GDTW Kernel

As noted by [8] and supported by the proof provided by [119], k_{GDTW} cannot be a PSD function in general. To use $k_{GDTW}(\mathbf{x}^r, \mathbf{y}^s)$ as a kernel function with any given data set, the obtained kernel matrix (Equation 5.7) using the corresponding kernel must satisfy Mercer's condition.

$$\mathbf{K}_{GDTW} = \left[\exp\left(-\frac{D(\mathbf{x}_i^{r_i}, \mathbf{x}_j^{r_j})}{\sigma^2} \right) \right]_{i,j=1}^{n} \tag{5.7}$$

where n is the number of samples in the training set, and $\mathbf{x}_i^{r_i}$ is the ith time series in the training set with a discrete time index varying between 1 and r_i.

$$p_{\mathbf{K}}^{n}(\lambda) = \det(\mathbf{K}_{GDTW} - \lambda \mathbf{I}_n) \tag{5.8}$$

Equation 5.8 shows the "characteristic polynomial" of $\mathbf{K}_{GDTW}$ for n samples, where $\mathbf{I}_n$ is the identity matrix of the same dimension as $\mathbf{K}_{GDTW}$, $\lambda \in \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ is the root of $p_{\mathbf{K}}^{n}(\lambda)$ and $\lambda_1, \lambda_2, \ldots, \lambda_n$ are the eigenvalues for the kernel matrix ($\mathbf{K}_{GDTW}$). In Equation 5.8 all eigenvalues should be non-negative so that $k_{GDTW}(\mathbf{x}, \mathbf{y})$ could be considered as a Mercer kernel [132]. We know that $p_{\mathbf{K}}^{n}(\lambda)$ is a polynomial of degree n, so there exist exactly n eigenvalues. On the other hand, since

$$D(\mathbf{x}_i^{r_i}, \mathbf{x}_j^{r_j}) = D(\mathbf{x}_j^{r_j}, \mathbf{x}_i^{r_i}) \tag{5.9}$$

then $\mathbf{K}_{GDTW}$ is a symmetric matrix and therefore all the eigenvalues are real.

For $n = 3$ in Equation 5.8 the sign of roots (eigenvalues) could be calculated as follows:

$$\begin{aligned}
&p_{\mathbf{K}}^{n=3}(\lambda) = \lambda^3 - 3\lambda^2 \\
&-\lambda\left[\exp(2cD(\mathbf{x}_1^{r_1}, \mathbf{x}_2^{r_2})) + \exp(2cD(\mathbf{x}_1^{r_1}, \mathbf{x}_3^{r_3})) + \exp(2cD(\mathbf{x}_2^{r_2}, \mathbf{x}_3^{r_3})) - 3\right] \\
&+[\exp(2cD(\mathbf{x}_1^{r_1}, \mathbf{x}_2^{r_2})) + \exp(2cD(\mathbf{x}_1^{r_1}, \mathbf{x}_3^{r_3})) + \exp(2cD(\mathbf{x}_2^{r_2}, \mathbf{x}_3^{r_3})) \\
&-2\exp(2cD(\mathbf{x}_1^{r_1}, \mathbf{x}_2^{r_2}))\exp(2cD(\mathbf{x}_1^{r_1}, \mathbf{x}_3^{r_3}))\exp(2cD(\mathbf{x}_2^{r_2}, \mathbf{x}_3^{r_3})) - 1] = 0
\end{aligned} \tag{5.10}$$

where $c = -\frac{1}{\sigma^2}$. For simplicity we assume:

$$l = \exp[2cD(\mathbf{x}_1^{r_1}, \mathbf{x}_2^{r_2})] + \exp[2cD(\mathbf{x}_1^{r_1}, \mathbf{x}_3^{r_3})] + \exp[2cD(\mathbf{x}_2^{r_2}, \mathbf{x}_3^{r_3})] - 3 \tag{5.11}$$

and

$$\begin{aligned}
m = &\exp[2cD(\mathbf{x}_1^{r_1}, \mathbf{x}_2^{r_2})] + \exp[2cD(\mathbf{x}_1^{r_1}, \mathbf{x}_3^{r_3})] + \exp[2cD(\mathbf{x}_2^{r_2}, \mathbf{x}_3^{r_3})] \\
&-2\exp[2cD(\mathbf{x}_1^{r_1}, \mathbf{x}_2^{r_2})]\exp[2cD(\mathbf{x}_1^{r_1}, \mathbf{x}_3^{r_3})]\exp[2cD(\mathbf{x}_2^{r_2}, \mathbf{x}_3^{r_3})] - 1
\end{aligned} \tag{5.12}$$

then we will have

$$p_{\mathbf{K}}^{n=3}(\lambda) = \lambda^3 - 3\lambda^2 - l\lambda + m = 0. \tag{5.13}$$

We know that $D(\mathbf{x}_i^{r_i}, \mathbf{x}_j^{r_j}) \geq 0$ then $0 < exp[2cD(\mathbf{x}_i^{r_i}, \mathbf{x}_j^{r_j})] \leq 1$, therefore $-3 < l \leq 0$ and $-1 < m < 1$. Based on Descarte's sign rule [3], Table 5.1 clarifies the sign of eigenvalues for all possible m and l. In Table 5.1, note that zero roots ($\lambda = 0$) are not included in maximum number of positive eigenvalues. Zero roots are only possible when $m = 0$. Also, if $l = 0$ then $m = 0$ (because in this case $exp[2cD(\mathbf{x}_i^{r_i}, \mathbf{x}_j^{r_j})]$ has to be equal to its maximum value, which is 1).

Figure 5.6 shows eigenvalues relating to l and m, which are obtained from Equations 5.11 and 5.12 respectively. This figure, as well as Table 5.1, confirms that the existence of negative eigenvalues (red dots in Figure 5.6) depends on DTW distances between data samples. Although Table 5.1 and Figure 5.6 show that Equation 5.13 has negative roots only when $m > 0$ and $l < 0$, we cannot expand this idea for the general case of n.

Therefore positive semi-definiteness of the k_{GDTW} depends on the DTW

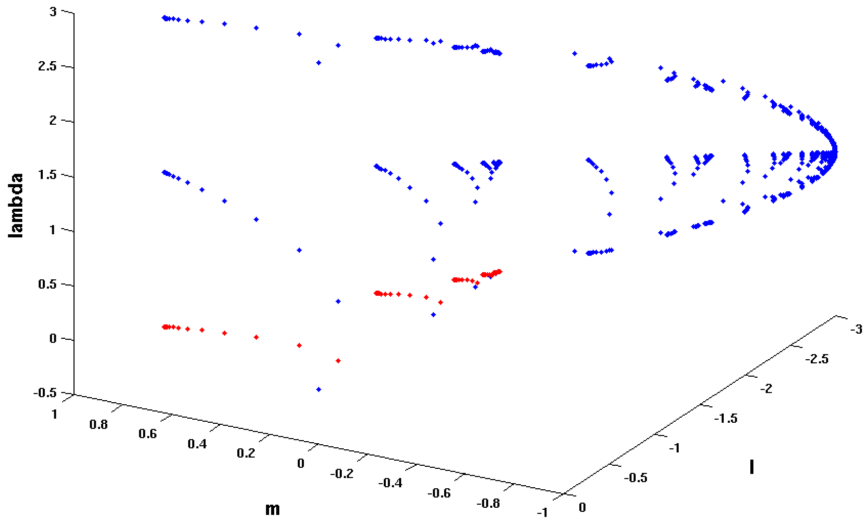

Figure 5.6: Positive (blue dots) and negative (red dots) eigenvalues(λ) obtained from Equations 5.11, 5.12 and 5.13 where $0 \leq D(\mathbf{x}_j^{r_j}, \mathbf{x}_i^{r_i}) < 100; i, j = 1, 2, 3$. Red dots show where the GDTW kernel is not positive semi-definite and blue dots show where the GTDW kernel is positive semi-definite.

Table 5.1: Sign of eigenvalues according to Equation 5.13, maximum number of negative and positive eigenvalues are shown for each set of conditions. Only when $l < 0$ and $m > 0$ may negative eigenvalues exist. Zero roots ($\lambda = 0$) are only possible when $m = 0$.

l	m	maximum #positive eigenvalues	maximum #negative eigenvalues	zero roots possibility
< 0	< 0	3	0	No
< 0	> 0	2	**1**	No
< 0	$= 0$	2	0	Yes
$= 0$	$= 0$	1	0	Yes

distances between the data objects and we cannot say that the GDTW function always satisfies or always does not satisfy Mercer's condition. The trainability of SVMs with the GDTW kernel is compared with the proposed method in Section 5.4.1.1.

Using the GDTW kernel in SVMs may result in a non-PSD kernel and in this case the existence of a Reproducing Kernel Hilbert Space is not guaranteed [168]. Another approach is to apply a transformation to the kernel matrix and make it PSD. This approach can lead to kernel matrices with large diagonal entries, resulting in overfitting [188]. Also it is not clear how this approach can handle new data objects in the test set [69]. As discussed in [69], fixing the diagonal values by subtracting the smallest eigenvalue from the diagonal does not increase the accuracy of the resulting classifier.

5.3.2 Combination of GDTW and P-SVMs

Employing DTW distance as a distance measure in Gaussian Kernel Functions and obtaining a new kernel function (Equation 5.14) called GDTW, is a tempting solution to waive SVM restriction on length of feature vectors. As previously discussed (in Subsection 5.3.1), it is not clear under what conditions the GDTW function (Equation 5.14) satisfies Mercer's conditions and could be considered as a valid kernel function for SVMs.

$$k_{GDTW}(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{D(\mathbf{x}, \mathbf{y})}{\sigma^2}\right) \tag{5.14}$$

On the other hand, the discussed two-step DTW-SVMs classification method has two main shortcomings: i) It needs the training data sets as well as the trained model when testing a new sample, and ii) while online testing is a common requirement of many time series classification problems such as speech recognition and handwriting recognition, testing against a large training data set using the two-step DTW-SVMs classifier (as shown in Figure 5.5) could be a very slow process.

To overcome the shortcomings of the two-step DTW-SVMs, and the shortcomings of using a non-PSD kernel in conventional SVMs, and being able to analyse data sets with different lengths in input series, we propose a new approach called GDTW-P-SVMs. It elaborates on P-SVM kernel functions, by utilising the DTW algorithm to provide an elastic distance measure for the kernel function. It utilises GDTW (Equation 5.14) as the kernel function in potential support vector machines (P-SVMs). In contrast to DTW-SVMs, which calculate the similarities between input series to obtain fixed-length feature vectors (the DTW matrix in Figure 5.3), the GDTW-P-SVM technique employs DTW in its kernel to waive the SVM requirement on fixed-length feature vectors.

Potential support vector machines (P-SVMs) [79] have been proposed by Hochreiter and Obermayer to analyse dyadic data where two sets of objects (row and column objects) are characterised by a matrix of numerical values. It is a maximum margin method for construction of classifiers and regression functions for the column objects in a data matrix. The P-SVM optimisation problem can be summarised as follows:

$$\begin{aligned} \textit{minimise} \quad & \tfrac{1}{2}\|\mathbf{X}_\phi^T \boldsymbol{\omega}\|^2 + C\mathbf{1}^T(\boldsymbol{\xi}^+ + \boldsymbol{\xi}^-) \\ \textit{subject to} \quad & \mathbf{K}^T(\mathbf{X}_\phi^T \boldsymbol{\omega} - \mathbf{y}) + \boldsymbol{\xi}^+ \geq \mathbf{0} \\ & \mathbf{K}^T(\mathbf{X}_\phi^T \boldsymbol{\omega} - \mathbf{y}) - \boldsymbol{\xi}^- \geq \mathbf{0} \\ & \mathbf{0} \leq \boldsymbol{\xi}^+, \boldsymbol{\xi}^- \end{aligned} \tag{5.15}$$

where $\boldsymbol{\omega}$ is a weight vector and $\boldsymbol{\xi}^+$ and $\boldsymbol{\xi}^-$ are slack variables used for the regularisation scheme proposed in [79]. A large value for the slack variables indicates that the particular object only weakly influences the direction of the classification boundary. In Equation 6.3, C is a constant value. If the noise is large, the value of C must be small to remove the corresponding constraints via the slack variables $\boldsymbol{\xi}$. After employing Lagrangian optimisation the following dual optimisation problem will be derived:

$$\begin{array}{ll} \textit{minimise} & \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{K}^T\mathbf{K}\boldsymbol{\alpha} - \mathbf{y}^T\mathbf{K}\boldsymbol{\alpha} \\ \textit{subject to} & \quad -C\mathbf{1} \leq \boldsymbol{\alpha} \leq C\mathbf{1}, \end{array} \tag{5.16}$$

where $\boldsymbol{\alpha}$ is the vector of Lagrange multipliers. Equation 5.16 depends on the data via the kernel or data matrix $\mathbf{K}$ only. One of the most crucial properties of the P-SVM procedure is that the dual optimisation problem depends on only $\mathbf{K}$ via $\mathbf{K}^T\mathbf{K}$. Therefore, $\mathbf{K}$ is not required to be positive semi-definite or square. This allows the construction of SVM-based classifiers for matrices $\mathbf{K}$ of general shape that include indefinite kernels. The offset b^* of the classification function $f(\mathbf{x}) = \sum_{i=1}^{l} y_i\alpha_i K(\mathbf{x}_i, \mathbf{x}) + b^*$ is given by [79]:

$$b^* = \frac{1}{l}\sum_{i=1}^{l} y_i. \tag{5.17}$$

The GDTW-P-SVMs method not only has the advantage of employing the DTW distance measure for comparing input series with different lengths but also overcomes the shortcomings of the two-step DTW-SVMs. As the testing phase for this approach is performed using only the created models, the training set is not required when testing a new data sample. This makes it convenient to distribute the trained data, which are essentially the models. On the other hand, in the testing process it is not required to compare each test sample against the entire training set. This makes it an appropriate method for problems that demand real-time classification results.

5.4 Experimental Results

To practically evaluate the effectiveness of the two-step DTW-SVMs classifier and GDTW-P-SVMs, a set of standard benchmark classification tasks is used for time series. The experiments used all time series available at the UCR repository [107], the character trajectory data set available at UCI machine learning repository [53] and GeoLife human trajectory data sets [209].

One common method to evaluate classification techniques is comparing results obtained from n-fold cross-validation optimisation. The optimisation involves tuning the hyperparameters (C, γ and kernel parameters) to minimise the error rate. We note that to obtain comparable results, whenever a tuning takes place, every adjustment should be considered as a separate independent experiment. The recommended procedure is to use "cross validation tuning" entirely within the training set and use a separate test set for evaluating the classification method [162]. When doing comparative evaluations, everything that is done to modify or prepare the algorithms must be done in advance of seeing the test data [162, 115]. In the experiments, to follow this recommendation a training subset and a testing subset are either pre-defined by the data set providers or a separate tuning set is defined to tune the hyperparameters.

A pairwise strategy with a one-vs-one policy is utilised for multi-class problems. For model selection, a five-fold cross-validation with a leave-one-out policy is performed on each pair of data. In the n-fold cross-validation technique, $n = 5$ and $n = 10$ are the two most commonly used values for number of folds. In the experiments some data sets have only a few samples of some classes. In these cases the number of samples in a pair can be less than the number of folds and therefore some folds may remain empty. To reduce the frequency of occurrence of empty folds $n = 5$ is used as the number of folds. If the number of samples in a pair is still less than the number of folds, then a sample repetition technique is used to ensure there is at least one sample in each fold. In essence, the sample repetition technique repeats the existing samples with consideration of the balance of data for both classes in the pair.

A shuffling technique is also applied to the data prior to fold generation. The technique helps to maintain a balance of the number of classes in each

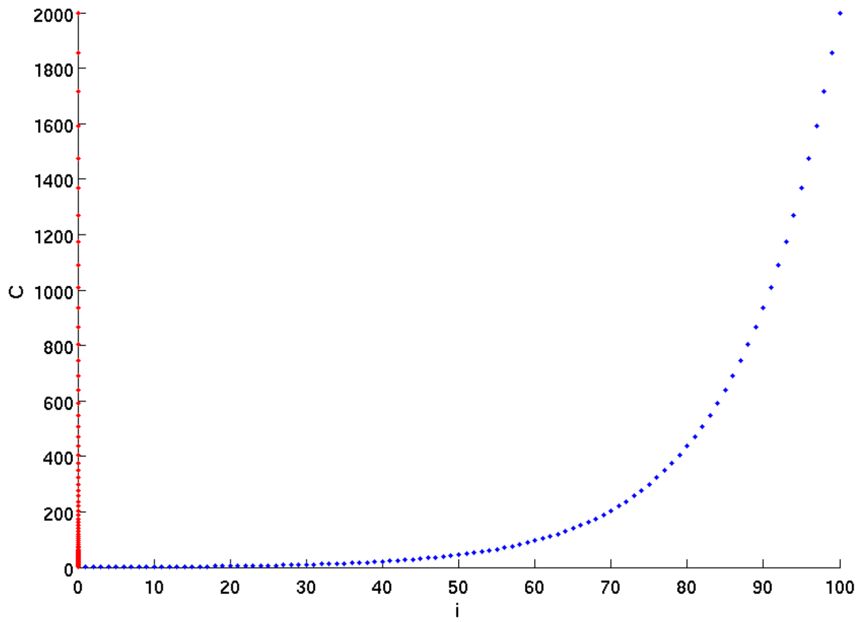

Figure 5.7: Exponential increment of C values for $p = 100$ and $C_{max} = 2000$; The best C values (Equation 5.3) are selected for each fold among a predefined set of values $\{C_1, C_2, \ldots C_p\}$ where $C_i = exp(i \times \frac{\ln(C_{max})}{p}); i = 0, 1, \ldots, p$.

fold. The accuracy of a model cannot be judged using an unbalanced testing set where the majority of its data belong to one class only. The shuffling technique runs over the data a hundred times to find the folds with balanced training and testing sets.

To perform data classification the LIBSVM [28] and the P-SVM [79] toolboxes are used for implementing the two-step DTW-SVMs and the GDTW-P-SVMs respectively. To ensure a fair comparison, the hyperparameter selection procedure was equal in all methods. Best values are selected from a generated hyperparameter set to minimise the error rate in the training phase. More precisely, the settings for the GDTW-P-SVMs and the two-step DTW-SVMs are listed below:

- *two-step DTW-SVMs*: The Gaussian kernel (Equation 5.2) is used as the kernel function for SVM learning. The best C values (Equation 5.3) are selected for each fold among a predefined set of values, $\{C_1, C_2, \ldots C_{p_c}\}$. A logarithmic distribution is used for C with higher density close to zero. The values in the set are obtained using $C_i = exp(i \times \frac{\ln(C_{max})}{p_c}); i = 0, 1, \ldots, p_c$, where p_c and C_{max} are two constant values that indicate number of values and the maximum value for C respectively (Figure 5.7). The same strategy has been employed for selecting the best γ among $\gamma_i = exp[\ln(\gamma_{min}) + i \times (\frac{\ln(\gamma_{max}) - \ln(\gamma_{min})}{p_\gamma})]; i = 0, 1, \ldots, p_\gamma$, where p_γ is the number of values for γ, γ_{min} and γ_{max} are the minimum and maximum values for γ, respectively (Figure 5.8).

- *GDTW-P-SVMs*: DTW Gaussian function is used (Equation 5.14) as the kernel function for P-SVM classification. C (see Equation 5.16) and γ ($\gamma = 1/\sigma^2$ in Equation 5.14) values are selected using the same methods as described for the DTW-SVMs.

All possible permutations of hyperparameters are used to find the minimum classification error rate for k folds. Then for each pair we have $s \geq k$ selected sets (some sets result in the same minimum error rate for the same fold). Among the s sets, the most frequent set with the lowest error rate is determined as the best hyperparameter set for that particular pair. If there is more than one set with that feature then the set that contains the biggest value for C will be selected as the best set.

Testing a new data sample is performed against all trained models (one for each pair). A predicted label with the highest number of votes will be selected as the class of the new data sample. If the highest number of votes belongs to more than one class then the same voting algorithm will be performed on the pairs consisting of selected classes only. This routine continues until eventually one class wins the competition. If the voting algorithm fails to find the winner, the label with the lowest class number among selected classes will be chosen as the predicted class for that particular data sample.

The standard DTW algorithm has quadratic time and space complexity that limits its use to only small time series data sets. To overcome this problem, the DTW algorithm described in [161] is used. The algorithm provides

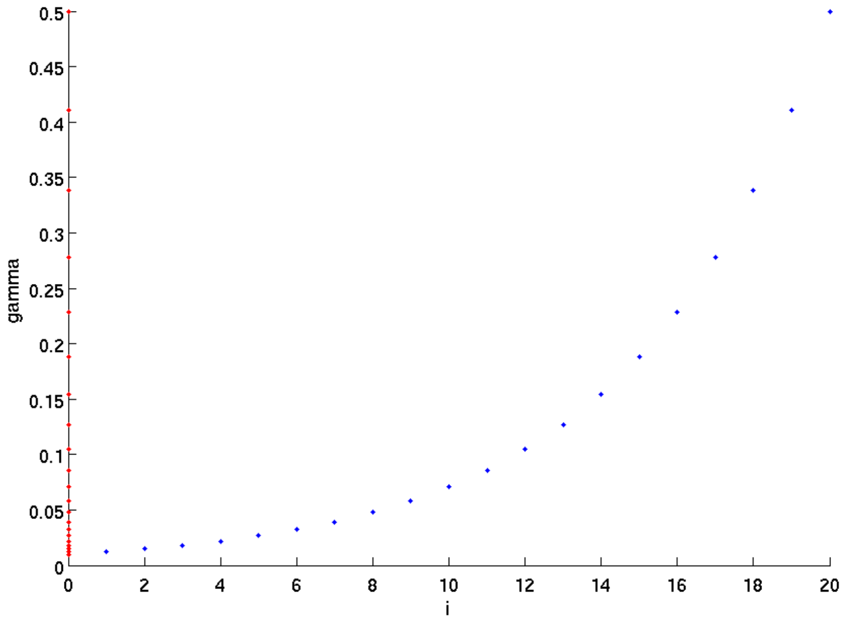

Figure 5.8: Exponential increment of γ values for $p = 20$, $\gamma_{min} = 0.01$ and $\gamma_{max} = 0.5$; The best γ values (Equation 5.2 and $\gamma = 1/\sigma^2$ in Equation 6.1) are selected for each fold among a predefined set of values $\{\gamma_1, \gamma_2, \ldots \gamma_p\}$, where $\gamma_i = exp[\ln(\gamma_{min}) + i \times (\frac{\ln(\gamma_{max}) - \ln(\gamma_{min})}{p})]; i = 0, 1, \ldots, p$.

the DTW alignments with linear time and space complexity. It uses a multilevel approach that recursively projects a solution from a coarser resolution and refines the projected solution. This makes it possible for the proposed classification technique, GDTW-P-SVMs, to have the same complexity as the P-SVMs with Euclidean distance. As discussed in [20], regardless of the exact algorithm used, the computational cost of solving the SVM Quadratic Problem grows at least like n^2 when C is small and n^3 when C gets large. It depends on the number of samples (n), the number of support vectors, and the hyperparameters (C and γ).

Large margin classifiers are known to be sensitive to data normalisation. The accuracy of a SVM can be severely degraded if the data is not normalised [68]. The main advantage of normalising is to avoid attributes in greater numeric ranges dominating those in smaller numeric ranges. Another advantage is to avoid numerical difficulties during the calculation. Because kernel values usually depend on the inner products of feature vectors, e.g. the linear kernel and the polynomial kernel, large attribute values might cause numerical problems [28]. The normalisation could be performed on input space (on the data sets) and feature space (in the kernel function). The RBF-based kernels normalise the feature space themselves [2]. This does not mean that input space normalisation is not required [68]. In the selected data sets for this experiment, there exist many features. Each of these features may be measured in a different scale and has a different range of possible values. In this case it is beneficial to scale all the features to a common range in each data set [2]. This method is also known as standardisation. For scaling the data, a "min-max" method is employed to scale the training data to the common range, $[0, 1]$:

$$\hat{x} = \frac{x - x_{min}}{x_{max} - x_{min}}, \tag{5.18}$$

where x_{min} and x_{max} are minimum and maximum values of the scaling data set, $\hat{x}$ is the scaled data, and x is the raw data sample. As previously discussed in section 5.2.3, the DTW-SVMs classification technique requires the training set as well as the trained models to test data samples. Maximum and minimum values of the training set are used to scale the test set to the desired range.

The proposed classification method is examined with two types of data sets:

1. *Fixed-length feature vectors*: Data sets where the data have a certain number of features and the values of all features for all samples are provided.

2. *Variable-length input series*: Data sets where the data do not have a fixed number features, or values of some features for some samples are not available. Trajectory-based data sets are one of the most common examples of this type, such as character trajectory data sets and human trajectory data sets.

The next two subsections discuss classification results obtained using both types of data.

5.4.1 Fixed-Length Feature Vector Classification

This section describes experiments that compare various classification techniques using data sets with fixed-length feature vectors. The classification performance of GDTW-P-SVMs is compared with two groups of classifiers. The first group contains classifiers that use the first nearest neighbour (1NN) technique along with a selection of common distance measures to classify UCR data sets. In the second group we compare the proposed method with kernel-based classifiers that employ DTW as a distance measure in their kernel. The outcome is a pairwise comparison of these classifiers with respect to their classification accuracy.

5.4.1.1 UCR data sets

In this section the UCR time series data sets [107] are used to benchmark the proposed methods and compare them with some other previously presented classification techniques. An overview of the UCR data set and its properties is given in Table 5.2. The benchmark represents a wide variety of practical classification problems including speech recognition, face recognition, motion tracking, data analysis, and electrocardiography data classification. The length of time series varies from 60 to 637 time steps and the data sets contain 24,009 time series in total. Each of the 20 data sets comes with a training set and a testing set.

Table 5.3 shows the results obtained for different classification methods. The methods, which are discussed in this table, employed the 1-nearest neighbor classifier with a distance measure. The 1-nearest neighbor classifier is used because the 1-nearest neighbor classifier with DTW showed very competitive performance and it has been widely used for time series classification [96]. In the table, for space concerns, the acronyms of the methods' names are used: 1NN ED (first nearest neighbour with Euclidean Distance), DTW (classic Dynamic Time Warping [182]), ODTW (Optimised Dynamic Time Warping [153]), LCSS (Longest Common Sub-sequence [34]) and ERP (Edit distance with Real Penalty [29]).

Figure 5.9 compares the proposed classification technique with the other methods. Red dots and blue dots show error rates for GDTW-P-SVMs and DTW-SVMs respectively for each data set. The black line in each diagram is representative of the case where both methods undergoing comparison would have equal error rates. More blue/red dots above the black line means the DTW-SVMs/GDTW-P-SVMs have lower classification error rates than the comparing method. As shown the GDTW-P-SVMs (red dots) have lower error rates in most cases even when comparing with powerful distance measures such as LCSS and ERP (last two diagrams in Figure 5.9). The DTW-SVMs technique also has a lower error rate than most other techniques, but it always has a higher error rate than GDTW-P-SVMs.

Figure 5.10 shows Receiver Operating Characteristic (ROC) curves of the GDTW-P-SVMs and SVMs with ED-Gaussian Kernel classifiers for the five data sets from the UCR repository that have two classes (binary classification problems) [50]. The False Positive Rate (FPR) is defined as the fraction of the false negatives out of total negatives, and the True Positive Rate (TPR) is defined as the fraction of the true positives out of total positives. The Area Under Curve (AUC) when using GDTW-P-SVMs for ECG, Gun-Point, Wafer, Coffee, and Yoga data sets are 0.825, 0.985, 0.916, 1.000, and 0.894 respectively and when using SVMs they are 0.891, 0.837, 0.688, 0.713, and

Table 5.2: UCR time series data set properties [107]

Database Name	Number of classes	Size of training set	Size of testing set	Length
Synthetic control	6	300	300	60
Gun-Point	2	50	150	150
CBF	3	30	900	128
Face(all)	14	560	1690	131
OSU Leaf	6	200	242	427
Swedish Leaf	15	500	625	128
50 Words	50	450	455	270
Trace	4	100	100	275
Two Patterns	4	1000	4000	128
Wafer	2	1000	6174	152
Face(four)	4	24	88	350
Lightning-2	2	60	61	637
Lightning-7	7	70	73	319
ECG	2	100	100	96
Adiac	37	390	391	176
Yoga	2	300	3000	426
Fish	7	175	175	463
Beef	5	30	30	470
Coffee	2	28	28	286
Olive oil	4	30	30	570

Table 5.3: UCR time series classification error rates, 1NN: first nearest neighbour, ED: Euclidean distance, ODTW: optimised DTW, LCSS: longest common sub-sequence, ERP: edit distance with real penalty. GDTW-P-SVMs show better results for automatic time series classification with fixed-length feature vectors. Best result(s), i.e. lowest error rate, for each data set are shown in bold.

Database Name	1NN ED	1NN ODTW	1NN DTW	1NN LCSS	1NN EPR	DTW SVM	GDTW P-SVM
Synthetic control	0.12	0.017	0.007	0.047	0.036	0.007	**0.000**
Gun-Point	0.087	0.087	0.093	0.013	0.040	0.200	**0.000**
CBF	0.148	0.004	0.003	0.009	0.003	**0.000**	**0.000**
Face (all)	0.286	0.192	0.192	0.201	0.201	0.256	**0.102**
OSU Leaf	0.483	0.384	0.409	**0.202**	0.397	0.355	0.330
Swedish Leaf	0.213	0.157	0.210	0.117	0.120	0.184	**0.094**
50 Words	0.369	0.242	0.310	**0.213**	0.281	0.264	0.222
Trace	0.24	0.01	**0.000**	0.20	0.170	**0.000**	**0.000**
Two Patterns	0.090	0.0015	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
Wafer	0.005	0.005	0.020	**0.000**	0.009	0.010	**0.000**
Face (four)	0.216	0.114	0.170	0.068	0.102	0.079	**0.023**
Lightning-2	0.246	**0.131**	**0.131**	0.180	0.148	0.197	0.164
Lightning-7	0.425	0.288	0.274	0.452	0.301	0.370	**0.260**
ECG	0.120	0.120	0.230	**0.100**	0.130	0.150	**0.100**
Adiac	0.389	0.391	0.396	0.452	0.378	0.371	**0.289**
Yoga	0.170	0.155	0.164	**0.137**	0.147	0.151	0.147
Fish	0.217	0.233	0.267	**0.091**	0.120	0.206	0.194
Beef	**0.467**	**0.467**	0.500	0.533	0.500	0.500	0.500
Coffee	0.250	0.179	0.179	0.214	0.250	0.179	**0.000**
Olive oil	**0.133**	0.167	**0.133**	0.800	0.167	**0.133**	**0.133**
Average Rank	4.650	3.000	3.650	2.800	3.000	3.100	1.400

0.568[3]. The ROC curves show that GTDW-P-SVMs have higher accuracy in classifying positive and negative samples than SVMs with Gaussian kernel. They also support the classification error rates presented in Table 5.3.

As seen in Figure 5.9 and the last column of Table 5.3, GDTW-P-SVMs clearly outperform other classification methods; in most cases the accuracy of GDTW-P-SVMs is higher than that of others. The experimental results for fixed-length feature vectors indicate that the proposed method (GDTW-P-SVMs) is promising for automatic time series classifications with fixed-length feature vectors.

Table 5.4 compares classification results obtained using kernel-based classification techniques that use DTW as the distance measure in their kernel function. In the table, for space concerns, the acronyms of the methods' names are used: ppfSVM-NDTW (pairwise proximity function SVM [66] with negated DTW kernel [69]), ppfSVM-GDTW (pairwise proximity function SVM with GDTW kernel), SVM-NDTW (conventional SVM with negated DTW kernel), SVM-GDTW (conventional SVM with GDTW kernel).

Figure 5.11 compares the proposed classification technique with other methods, which were discussed in Table 5.4. In this figure, the same representation as in Figure 5.9 is used. As shown the GDTW-P-SVMs (red dots) have lower error rates in all cases even when compared with pairwise proximity function SVM with GDTW and NDTW kernels (last two diagrams in Figure 5.9).

The last rows of Table 5.3 and Table 5.4 show the "average rank" of each classifier using the Friedman test [54]. The average rank is calculated for each group of classifiers separately. To obtain the average rank initially the classifiers were ranked on each data set separately. Then for each data set the classifier with the lowest error rate (highest performance) is assigned rank 1, the second best rank 2, and so on. In the case of ties, average ranks are assigned for that data set. Then the ranks are averaged over all data sets in each group.

Critical value for the two-tailed Bonferroni-Dunn test [46] with $\alpha = 0.05$

[3]Closer the value of AUC to 1, higher the accuracy of classification.

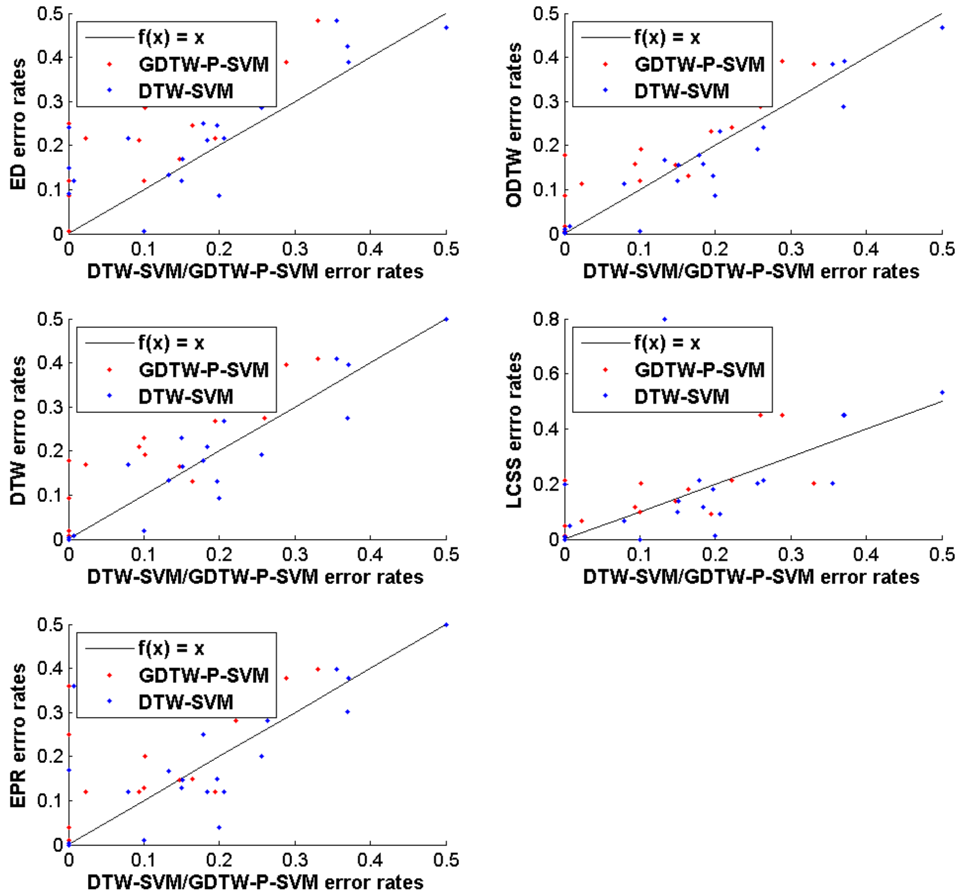

Figure 5.9: Comparison of time series classification methods (presentation method adopted from [130]). $x-axes$ represent error rates for DTW-SVM and GDTW-P-SVMs with blue and red dots respectively. $y-axes$ show error rates for the other five classification methods, which were compared in Table 5.3. Black lines represent $f(x) = x$. More blue/red dots above the black line means that the DTW-SVM/GDTW-P-SVMs have lower classification error rates than the comparing method. As shown the GDTW-P-SVMs (red dots) have lower error rates in most cases even when compared with powerful distance measures such as LCSS and ERP (last two diagrams in the figure).

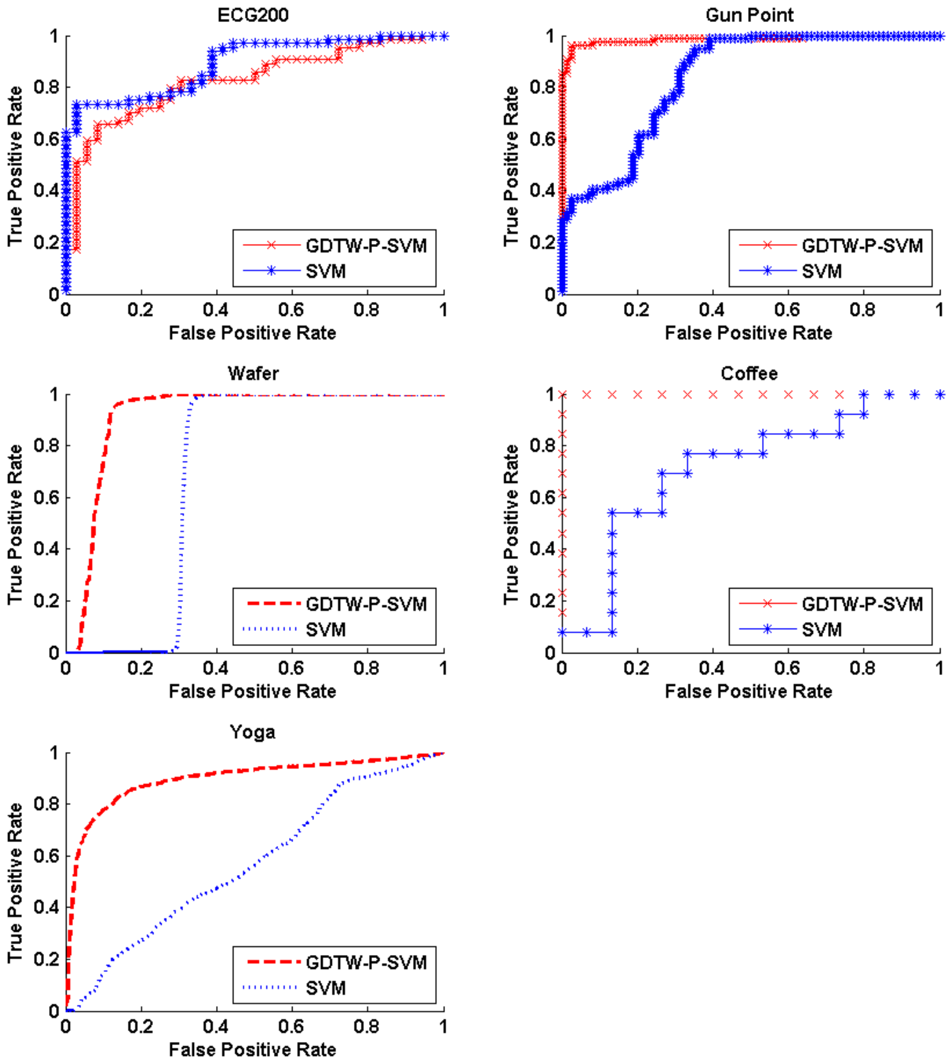

Figure 5.10: Comparison of Receiver Operating Characteristic (ROC) curves for SVMs with Gaussian Kernel and GDTW-P-SVMs for five UCR data sets with two classes; False Positive Rate (FPR) is defined as the fraction of the false negatives out of total negatives. True Positive Rate (TPR) is defined as the fraction of the true positives out of total positives. The Area Under Curve (AUC) when using GDTW-P-SVMs for ECG, Gun-Point, Wafer, Coffee, and Yoga data sets are 0.825, 0.985, 0.916, 1.000, and 0.894, respectively, and when using SVM they are 0.891, 0.837, 0.688, 0.713, and 0.568 (the closer the value of the AUC is to 1, the higher the accuracy of classification).

Table 5.4: UCR time series classification error rates using DTW-based kernel function classifiers. ppfSVM/NDTW: pairwise proximity function SVM with negated DTW kernel, ppfSVM/GDTW: pairwise proximity function SVM with GDTW kernel, SVM/NDTW: conventional SVM with negated DTW kernel, SVM/GDTW: conventional SVM with GDTW kernel. GDTW-P-SVMs show promising results for automatic time series classification with fixed-length feature vectors. Best result(s) for each data set is shown in bold. The classifiers were ranked on each data set according to their performance and the ranks averaged over all data sets (lower rank indicates better performance.)

Database Name	ppfSVM NDTW	ppfSVM GDTW	SVM NDTW	SVM GDTW	DTW SVM	GDTW P-SVM
Synthetic control	0.013	0.013	0.013	0.023	0.007	**0.000**
Gun-Point	0.047	0.140	0.460	0.127	0.200	**0.000**
CBF	0.003	0.001	0.010	0.046	**0.000**	**0.000**
Face(all)	0.237	0.226	0.170	0.265	0.256	**0.102**
OSU Leaf	0.405	0.355	0.706	0.401	0.355	**0.330**
Swedish Leaf	0.147	0.155	0.363	0.382	0.184	**0.094**
Trace	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
Two Patterns	**0.000**	0.001	0.007	**0.000**	**0.000**	**0.000**
Wafer	0.010	0.015	0.181	0.034	0.010	**0.000**
Face(four)	0.148	0.114	0.102	0.114	0.079	**0.023**
Lightning-2	0.328	0.316	0.492	**0.115**	0.197	0.164
Lightning-7	0.301	0.315	0.219	0.301	0.370	**0.260**
ECG	0.160	0.220	0.440	0.170	0.150	**0.100**
Adiac	0.325	0.343	0.512	0.560	0.371	**0.289**
Yoga	0.227	0.177	0.534	0.219	0.151	**0.147**
Fish	**0.189**	0.240	0.240	0.297	0.206	0.194
Beef	0.567	0.533	0.633	0.600	**0.500**	**0.500**
Coffee	0.107	0.179	0.500	0.179	0.179	**0.000**
Olive oil	0.167	0.267	**0.133**	0.267	**0.133**	**0.133**
Average rank	3.421	3.868	4.737	4.500	3.052	1.421

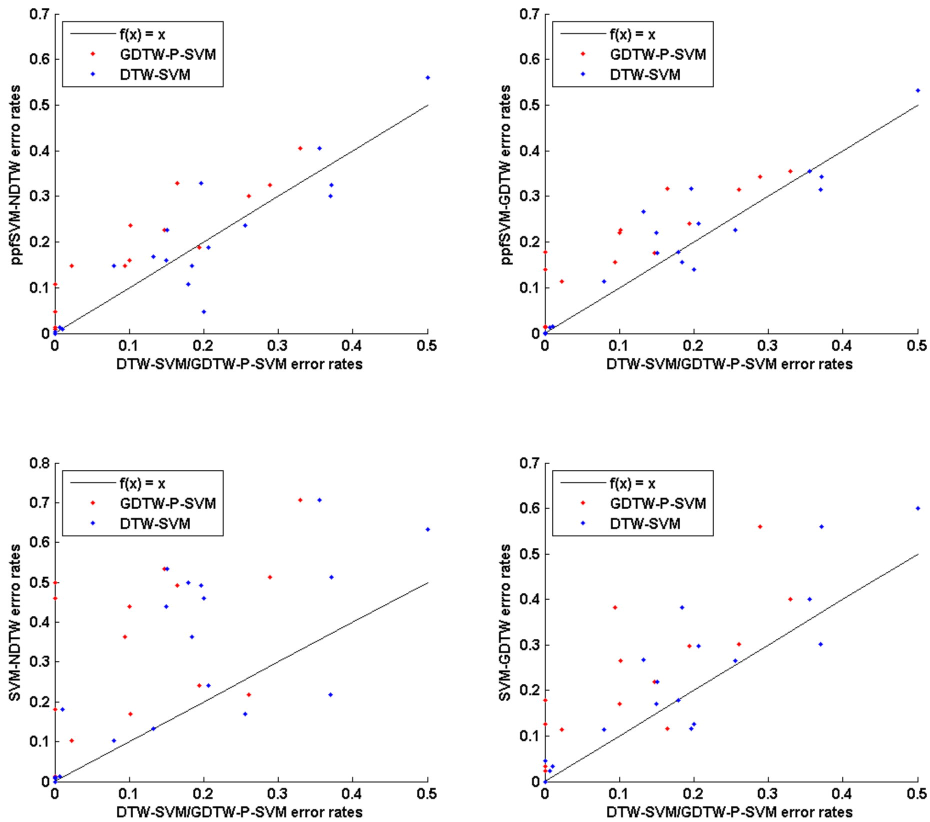

Figure 5.11: Comparison of time series classification methods (presentation method adopted from [130]). $x - axes$ represent error rates for DTW-SVM and GDTW-P-SVMs with blue and red dots respectively. $y - axes$ show error rates for the other four classification methods, which were compared in Table 5.4. Black lines represent $f(x) = x$. More blue/red dots above the black line means that the DTW-SVM/GDTW-P-SVMs have lower classification error rates than the comparing method. As shown the GDTW-P-SVMs (red dots) have lower error rates in most cases even when compared with ppfSVM-NDTW and ppfSVM-GDTW (the first two diagrams in the figure).

for Table 5.3 is $q_\alpha = 2.638$. Critical difference (CD)[4] for this table is obtained as:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} = 1.8021 \tag{5.19}$$

Where k is the number of classification techniques and N is the number of data sets. Critical value for the two-tailed Bonferroni-Dunn test with $\alpha = 0.05$ for Table 5.4 is $q_\alpha = 2.576$. Critical difference (CD) is obtained as:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} = 1.5636 \tag{5.20}$$

A pairwise comparison of the average rank of classifiers and the obtained critical value for Table 5.3 and Table 5.4 using the Bonferroni-Dunn test are presented in Table 5.5 and Table 5.6, respectively. Bold values in Table 5.5 and 5.6 show that the corresponding classifier on the left performs significantly better than the corresponding classifier on top. The values in these two tables are obtained by subtracting the average ranks of corresponding classifiers. If the value is more than the critical difference then the difference between the compared classifiers is significant [42].

The differences between the rank of GDTW-P-SVMs and the ranks of other classification methods in the first group of classifiers are always greater than the CD (obtained in Equation 5.19). Therefore GDTW-P-SVMs perform significantly better than the other classification methods that are discussed in Table 5.3. Although the difference between GDTW-P-SVMs and 1NN-LCSS is just above the CD, GDTW-P-SVMs still have statistically significantly better performance. In the second group of compared classifiers, the rank of GDTW-P-SVMs showed a greater difference compared to others. As shown in Table 5.6, the differences between the rank of GDTW-P-SVMs and the rank of other classification methods are always greater than the CD obtained in Equation 5.20. Therefore GDTW-P-SVMs perform significantly better than other classification methods that are shown in Table 5.4.

We have not used the ANOVA [52] to evaluate the proposed classifcation

[4] The performance of two classifiers is significantly different if the corresponding average ranks differ by at least the critical difference [42].

Table 5.5: A pairwise comparison of the average rank of classifiers discussed in Table 5.3 using Bonferroni-Dunn test. The classifiers employed 1-NN with a selection of common measure distances. Bold values in the table show that the corresponding classifier on the left performs significantly better than the corresponding classifier on top. The $CD = 1.802$ for this group of classifiers is obtained using Equation 5.19.

Classifiers	1NN ED	1NN ODTW	1NN DTW	1NN LCSS	1NN EPR	DTW SVM	GDTW P-SVM
1NN NDTW	N/A	-1.75	-1.225	-2	-1.65	-1.625	-3.825
1NN ODTW	1.75	N/A	0.525	-0.25	0.1	0.125	-2.075
1NN DTW	1.225	-0.525	N/A	-0.775	-0.425	-0.4	-2.6
1NN LCSS	2	0.25	0.775	N/A	0.35	0.375	-1.825
1NN EPR	1.65	-0.1	0.425	-0.35	N/A	0.025	-2.175
DTW SVM	1.625	-0.125	0.4	-0.375	-0.025	N/A	-2.2
GDTW P-SVM	**3.825**	**2.075**	**2.6**	**1.825**	**2.175**	**2.2**	N/A

Table 5.6: A pairwise comparison of the average rank of classifiers with DTW-based kernel (discussed in Table 5.4) using Bonferroni-Dunn test. Bold values in the table show that the corresponding classifier on the left performs significantly better than the corresponding classifier on top. The $CD = 1.5636$ for this group of classifiers is obtained using Equation 5.20.

Classifiers	ppfSVM NDTW	ppfSVM GDTW	SVM NDTW	SVM GDTW	DTW SVM	GDTW P-SVM
ppfSVM NDTW	N/A	0.447	1.316	1.079	-0.368	-2
ppdfSVM GDTW	-0.447	N/A	0.868	0.631	-0.816	2.447
SVM NDTW	-1.316	-0.868	N/A	-0.237	-1.684	-3.316
SVM GDTW	-1.079	-0.631	0.237	N/A	-1.447	-3.079
DTW SVM	0.368	0.816	1.684	1.447	N/A	-1.632
GDTW P-SVM	**2**	**2.447**	**3.316**	**3.079**	**1.632**	N/A

technique because:

1. ANOVA assumes that the classification error rates (performance) are drawn from a normal distribution, which is not always the case in general.

2. ANOVA requires that random variables have equal variance. Neither learning algorithms nor data sets can satisfy this condition [42].

In the next two subsections the proposed classification methods are tested against two trajectory-based data sets with variable-length input series.

5.4.2 Variable-Length Input Series Classification

The comprehension of phenomena related to movement – not only of people and vehicles but also of animals and other moving objects – has always been a key issue in many areas of scientific investigation and social analysis. Collected data for movement based analysis are called *trajectory data* and it can be represented as sequences of time stamped locations. Trajectory data are normally obtained from location-aware devices that capture the position of an object at a specific time interval. Since object movements occur at different speeds the trajectory data are variable-length input series, which makes them suitable data sets for the GDTW-P-SVMs. In this section the classification results for the character [53] and human [208] trajectory data sets are presented.

5.4.2.1 Character trajectory data set

The character trajectory data set consists of labelled samples of pen tip trajectories recorded whilst writing individual characters. All samples are from the same writer, for the purposes of primitive extraction. Only characters with a single pen-down segment were considered. The data consist of 2858 character samples with different lengths. Each sample is a 3-dimensional velocity trajectory (x, y, and pen tip force). The data has been numerically differentiated and Gaussian smoothed, with a sigma value of 2 [53, 171]. The classification task is to recognise characters in the data set using trained models.

Table 5.7 represents the classification error rates resulting from the experiments. Three data representations are used for character classification in [127]:

1. *Likelihood*: Employs the label information that is available for the objects in the training data and represents the data using maximum likelihood [127].

2. *Fisher kernel*: The Fisher kernel is defined as the inner product of the directions of gradient ascent, i.e., the inner product of the natural gradients. It simply uses the gradients as features, without any further rescalings or normalisations [171].

3. *Fisher Kernel learning (FKL)*: Trains the model in such a way that objects with the same class induce gradients that are similar, whereas objects with different classes induce log-likelihood gradients that are dissimilar [127].

In the proposed classification method, a simple data projection for representing the data is used. It projects 3D trajectory data (2D coordinates and pen force value) into 1D variable-length sequential data samples. As seen in Table 5.7 the proposed method has lower error rates compared to the other methods.

5.4.2.2 Human Trajectory Data Sets

The rise of GPS and broadband-speed wireless devices has led to tremendous excitement about a range of applications broadly characterized as "location based services". These applications will provide users with information that is targeted and personalised to their location, whether it be nearby stores, friends, or traffic conditions, etc.

The human trajectory data set that is used in the experiments was collected in the Geolife project by 167 users in a period of over three years. A GPS trajectory of this data set is represented by a sequence of time-stamped points, each of which contains the information of latitude, longitude and altitude. This data set contains 17,355 trajectories with a total distance of about 1 million

Table 5.7: Classification error rate (in percent) on the handwritten character data set for four different classification methods presented in [127] as well as our classification methods. The data set includes data object with variable-length input series. While no feature presentation method is applied to the data for the GDTW-P-SVM and DTW-SVM methods, they have shown the lowest classification error rates.

Classifier	Feature Representation	Error rate
Bayes	Likelihood	12.46
Softmax	Likelihood	8.14
	Fisher	8.23
	Fisher Kernel Learning	6.95
SVMs	Likelihood	7.91
	Fisher	7.64
	Fisher Kernel Learning	6.91
DTW-SVM	–	**5.450**
GDTW-P-SVM	–	**3.010**

kilometers and a total duration of 48,000+ hours. These trajectories were recorded by different GPS loggers and GPS-phones. This data set recorded a broad range of users' outdoor movements, including not only life routines like going home and going to work but also some entertainment and sports activities, such as shopping, sightseeing, dining, hiking, and cycling [208].

The classification task defined here is based on supervised learning to automatically recognise users' transportation modes, such as driving, walking, taking a bus, riding a bike and traveling on a train, from raw GPS logs. 59 users have labeled their trajectories with transportation mode. The total distance and duration of transportation modes are listed in Table 5.8. Trajectories with unknown and airplane transportation mode were excluded from the data set.

Table 5.9 shows the classification accuracy for our approach as well as 4 other classification methods described in [207] over the training set and testing set. In [207] two segmentation methods, by length and by duration, along with a classifier were used to recognise the transportation mode. As the proposed method is able to handle data samples with different lengths this step can

Table 5.8: Total distance and duration of transportation modes in the GeoLife data set [208].

Transportation Mode	Distance (km)	Duration (hours)	#train	#test
Walk	11,457	5,126	1,586	1,910
Bike	6,335	2,304	274	602
Bus	21,931	1,430	930	629
Car and Taxi	34,127	2,349	318	324
Train	74,449	459	412	870
Total	18,7679	12,041	3,520	4,335

be waived and the raw GPS trajectory data can be used to recognise the transportation mode. Here again a dimension projection that projects 3D (latitude, longitude, and altitude) into 1D variable-length sequential data is used. As seen in Table 5.9 the accuracy of the proposed approach (GTDW-PSVMs) is higher than the other approaches.

5.5 Discussion and Future Work

This new technique couples a SVM-based classification technique with an indefinite kernel. The proposed coupling combination was compared with a number of other combinations that have been recently proposed (SVMs-NDTW, SVMs-GDTW, ppfSVMs-NDTW, and ppfSVMs-GDTW). In addition to those combinations, there exist a variety of indefinite kernels and classification techniques that can be coupled, such as [144, 126]. The kernels do not satisfy Mercer's condition and they induce associated functional spaces called Reproducing Kernel Krein Space (RKKS), which is a generalisation of Reproducing Kernel Hilbert Space (RKHS). This chapter emphasised the importance of such couplings by giving GDTW-P-SVMs as an example with a classification accuracy that is significantly higher than existing methods for wide varieties of benchmarked data sets. Experimenting with indefinite kernels and other combinations of kernel-based classification techniques that can handle indefinite kernels is a possible direction of future research.

Table 5.9: Classification accuracy for the GeoLife data set. The data set includes data objects with variable-length input series. While no segmentation method is applied to the data for the GDTW-P-SVM and DTW-SVM methods, they have shown the highest classification accuracy.

Classifier	Segmentation Method	Accuracy (%)
Decision Tree	by length	70
	by duration	75
SVMs	by length	57
	by duration	62
Bayes net	by length	69
	by duration	71
CRF	by length	53
	by duration	40
DTW-SVM	–	**79**
GDTW-P-SVMs	–	**81**

Although in the experiments the GDTW-P-SVMs were employed to solve classification problems, the ability of GDTW-P-SVMs to handle variable length data objects can be utilised for time series segmentation. For example, an energy-based model for unsupervised factorisation has been employed for unsupervised time series segmentation with fixed-length using SVMs [140, 139]. A similar approach could be applicable for segmenting time series with variable length using GDTW-P-SVMs.

5.6 Summary

A new classification technique, GDTW-P-SVMs, was introduced for sequential data analysis where each data object is characterised by a series of numerical values that may have different lengths for different data objects. The new technique is a maximum margin method for the construction of classifiers with variable-length input series. The well-known DTW algorithm was utilised to provide an elastic distance measure that is able to compare variable-length input series. We compared GDTW-P-SVMs with the two-step DTW-SVM method where training data were required in the testing phase as well as the trained models. Although DTW-SVMs were able to classify trajectory data with acceptable error rates, they are not able to provide the classification results in real-time as the testing phase for this technique is too slow for problems that have a big training set. GDTW-P-SVMs were proposed to overcome the shortcomings of the DTW-SVM by altering the kernel function in P-SVM using DTW. As a result, GDTW-P-SVMs could handle data and kernel matrices that were neither positive definite nor square, and it could also be applied to data with variable-length input series. Benchmarks for classification were performed with several real-world data sets from the UCR Time Series Classification/Clustering page, the GeoLife trajectory data set, and the character trajectory from the UCI repository. The data sets included data with both variable and fixed-length input series. Classification error rates and ROC curves showed that GDTW-P-SVMs can converge into the optimal separating hyperplane with maximum margin in classification problems with fixed-length feature vectors. We also discussed the reason why DTW was not used as a kernel distance measure in standard SVMs. In the case of variable-length data samples, GDTW-P-SVMs significantly outperformed other existing methods by two main advantages: *i*) the proposed method had significantly lower classification error rates and *ii*) at the same time it waived the need for data representation to provide fixed-length feature vectors. The second advantage is important when the extraction of fixed-length feature vectors is not feasible or when using fixed-length segments of data objects fails to properly describe the relationship between data objects.

The proposed classification techniques will be used to distinguish between different classes of behavioural data. The data include time-stamped spatial characteristics with different lengths. The data were extracted/generated at Wheeler Place using the proposed simulation software and pedestrian detection and tracking system. The next chapter will provide more details and experimental results of utilising the classification technique.

Chapter 6

Spatial behaviour analysis

The content of this chapter has been presented in [90, 94, 89].

Contents

The analysis of pedestrians' reactions to their immediate surroundings in indoor and outdoor areas (pedestrian spatial behaviour) is an important facet of architectural and urban design. A simulated environment is used to generate pedestrian behavioural characteristics data. The simulation allows us to investigate spatial visual behaviour without the difficulties of real-world behavioural feature extraction. While a range of simulation software has been proposed, the spatial behaviour simulator (described in Chapter 2) is used [92, 93]. The selected simulation software presented in [92] and [93] focuses on the visible static and dynamic properties of the physical environment, such as attractors with different conspicuity areas, crowd attraction with variable levels of attraction, and the impact of the crowd gaze vector on the behaviour of others. While a range of names have been proposed for attractive objects like billboards or display stands, in urban spaces the term ‘attractor’ is simply used [93, 91, 90, 92]. The simulator is also able to model advanced spatial behavioural characteristics such as gaze vector, speed of movement, useful field of view, aim of movement, goal-driven attention, interpersonal distance and stimulus-driven attention. It provides rich information on how pedestrians react to their surroundings and also to other pedestrians in an architectural urban space (agent-to-agent reaction).

In addition to the simulated characteristics, real-world pedestrians’ trajectories are also used for analysing spatial behaviour. The real-world trajectories are extracted using the proposed pedestrian detection and tracking system described in Chapter 3.

The behaviour analysis system employs a special version of support vector machines (SVMs), called GDTW-P-SVM (described in Chapter 5) that is capable of handling input series that might have different lengths. The analysis system learns pedestrians’ behaviour patterns based on the characteristics observed/generated in an architectural environment. Instead of using a fixed-length segment of a behavioural sequence, the whole sequence for each individual pedestrian is used as an input to the classifier. This chapter presents an analysis system that can describe the relationship between pedestrians’ behavioural characteristics and their flow dynamics in an urban space.

6.1 Previous Works

Analysing the impacts of attractors on pedestrian spatial behaviour will help us to improve the planning of pedestrian environments. This type of analysis has been conducted previously in a large number of studies such as architecture, cognitive science and environmental research [189, 141, 6]. For instance, William H. Whyte analysed the behaviour of pedestrians in complex urban environments and concluded that pedestrians’ flow dynamics were governed by desires to move to some particular points in space such as urban space attractors [189]. In another approach, Suzuki et al. proposed a computational method to learn motion patterns and detect anomalies by human trajectory analysis [176]. They employed hidden Markov models (HMMs) to model time-series features of human positions. Using a similarity matrix of HMM mutual distances and k-means clustering, they acquired features of human motion patterns [176].

Some of the relevant studies used agent-based models or computer simulations [58, 6] while others dealt with real humans [207, 159]. However, analysing the spatial behaviour of pedestrians using an automated intelligent system has rarely been investigated. This is either because extracting such an amount of information from observers was a time consuming process, or an accurate classifier able to work with variable-length data objects was not available [208, 58, 207]. There are still many open questions about how the aesthetics of the environment interact with human pedestrians or users of space, and how it compares to other, more functional factors, such as path widths, the availability of open space, and the presence of obstacles or attractive objects [193].

The analysis of trajectory data has been conducted in many research areas including scientific investigation and social analysis. This kind of analysis can be applied to any moving object and is not limited to pedestrian movement. Collected data for movement-based analysis are called trajectory data and they can be represented as sequences of time stamped locations. The recent advances and price reduction of technologies for collecting spatial data such as satellite images, cellular phones, sensor networks, and GPS devices have facilitated the collection of data referenced in space and time. These huge collections of data often contain important information that conventional systems are unable to discover.

Trajectory based classification can be based on simple rules such as decision making according to length of trajectory, and number of stops [45, 72, 205], or using learning methods to exploit statistical regularities such as neural networks [157], Bayesian networks [208, 207, 195] and the HMM [135, 136]. Since the object movements occur at different speeds, the trajectory data have variable lengths. Conventional classifiers such as Support Vector Machines (SVMs) and Artificial Neural Networks (ANNs), can only classify data objects with fixed-length feature vectors. Zheng and his team [208, 207] collected 17,355 GPS trajectories from 167 users in a period of over three years. They applied two segmentation methods (by length and by duration) on this data set to obtain fixed-length feature vectors. They employed SVMs, Bayesian Networks, and Decision Trees to recognise users' transportation modes. Fixed-length segments (features) may fail to contain the information needed to properly describe samples in the input space. Therefore, in classifying trajectory data the accuracy of the classifier using fixed-length feature vectors is very dependent on the segmentation or feature selection algorithm. Instead of using some fixed-length segments (features) to describe the trajectory data, the proposed analysis system considers the entire trajectory for each user as a single input to the system. To achieve this, two classification methods, DTW-SVM and GDTW-P-SVMs, which were previously described, are employed to classify trajectories [88]. The classifiers are capable of learning data entries with different length input series.

Spatial behaviour analysis is not only limited to pedestrian movement, but can also be expanded to other behavioural characteristics such as gaze dynamics, speed of head movement and speed of pedestrian movement over time. Especially when it comes to analysing people's attentions in public places, sudden changes in gaze vector are important behavioural characteristics of individuals. While attractive objects take up a significant proportion of visual external stimuli, other people's gaze direction could be considered as another important category of external stimuli to attract a pedestrian's gaze [92]. Detecting these sudden changes in pedestrian movement dynamics caused by environmental settings and differentiating them from normal/casual movements can extract valuable information from trajectory data. The main focus of this chapter is to introduce a new system that is able to recognise any abnormality in pedestrians' trajectory-based data. The system employs an intelligent method to distinguish the casual behaviours from the attracted ones. The classifiers model an input space using the entire sequence of data for individuals instead of modeling a set of selected features or fixed-length segments of the sequence. The new aspect, which is focused on in this research, is the use of GDTW-P-SVMs to classify simulated and real-world trajectory data sets into several predefined behavioural categories.

6.2 Spatial Characteristics Selection

In the simulation software each agent is presented as a series of behavioural features. The following features are used to analyse the agent's spacial behaviour [93]:

- (x_t^i, y_t^i): The location of the ith agent in a 2D plan at time t.
- $(\frac{dx}{dy}, \frac{dy}{dt})$: The speed of movement of agents at time t.
- (α_t): The angle between the direction of movement (also known as speed vector in the simulation) and the gaze vector of an agent at time t.
- $(\frac{d\alpha}{dt})$: The derivative of the angle in respect to t.

The simulation software is capable of modelling virtual attractive objects with a dynamic level of attraction. One example of these types of attractors is a *social group*. The social group is a fundamental and universal feature of human social life. The dynamic level of attraction reflects the impacts of the size of the group such that larger groups are able to draw more attention than smaller groups. The location and level of attraction of a virtual attractive object (VAO) vary according to the location and number of agents in the group [92]. Figure 6.1 shows how the simulation software models the impact of crowd attraction on agents' spatial behaviour.

An image of the agents as modelled in [93] is shown in Figure 6.2. The simulation is a multi-agent-based simulator and each agent represents pedestrian spatial behaviour in an urban space. As agents move towards their destination they might be distracted by several environmental features such as attractors, other pedestrians, and their purpose of walking. Therefore even agents with the same speed and destination are likely to have trajectories with different lengths. Comparing these trajectories to find the differences in agents'

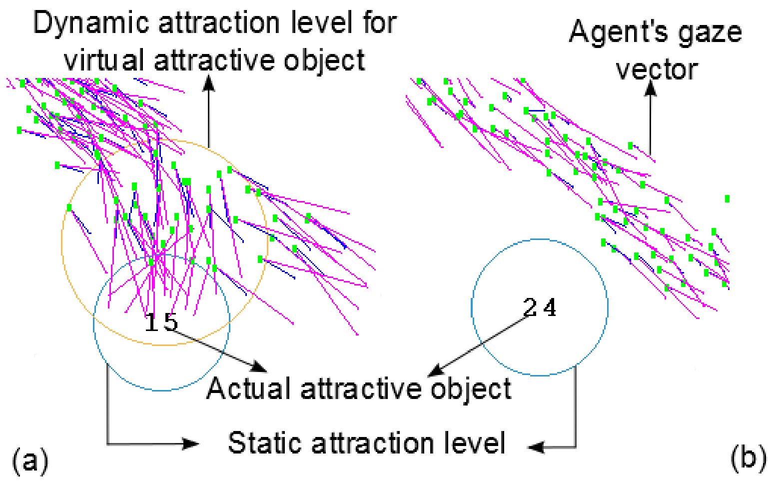

Figure 6.1: Agent's behavioural characteristics (a) with and (b) without the impact of a virtual attractive object [92]. By considering crowd attraction in the simulation, as the attracted crowd around an attractor grows bigger more agents become attracted to the attractor from further distances.

behaviour is a key feature in analysing spatial behaviour. The conventional approach to compare trajectories with different lengths is to use a window-based method to make segments with fixed-length and then apply a distance measure to compare the segments [208]. However, analysing only a segment of this trajectory may not accurately reflect the agent's behavioural differences.

6.3 Analysis System

One approach to cope with the problem of trajectories having different lengths is to use distance measures that can compare signals with different lengths, such as Dynamic Time Warping (DTW). DTW is a well-known elastic distance measure that uses dynamic programming for obtaining the distance between two sequences of sequential data [182]. Figure 6.3 shows how DTW can compare two trajectories with different lengths. The trajectories are warped or stretched non-linearly in time dimension to determine a measure of similarity independent of non-linear variation in the time dimension. To describe the be-

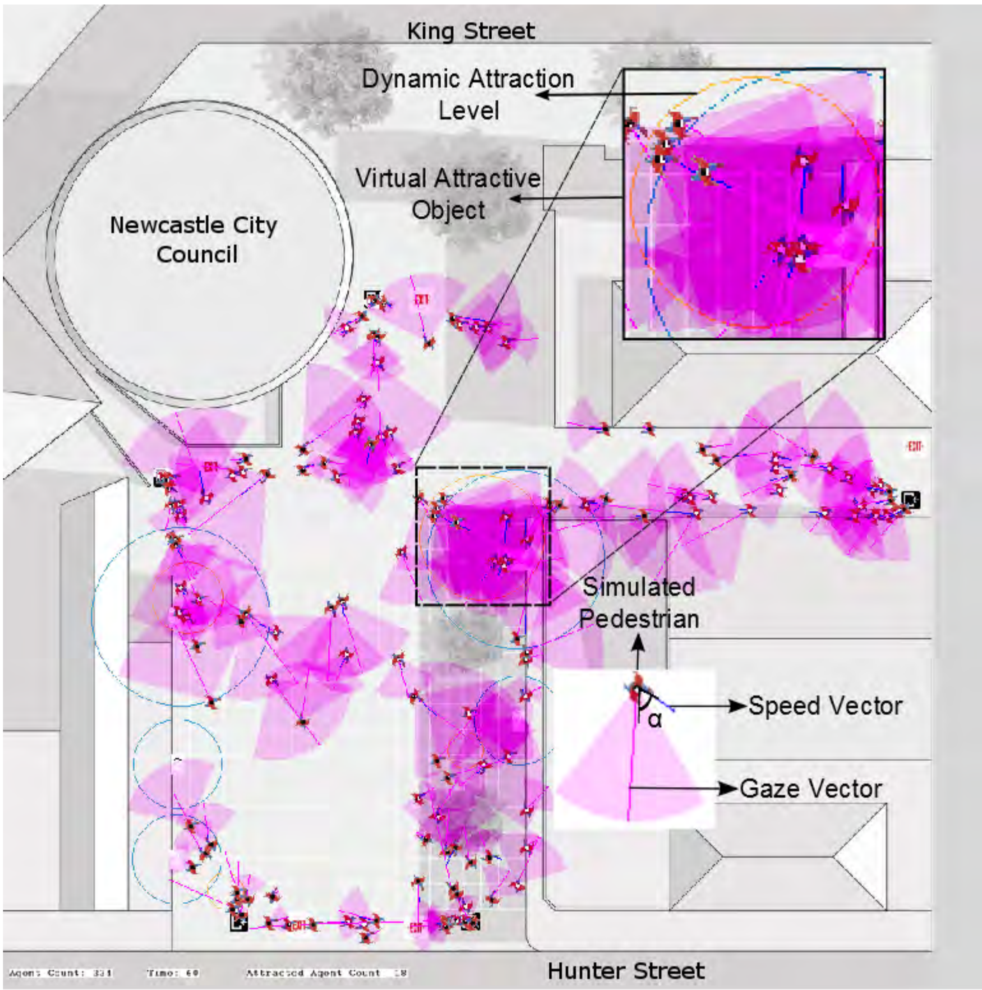

Figure 6.2: Pedestrian spatial behaviour simulation on the Wheeler Place 2D plan [92]. Level of attractions for attractors is shown with blue circles. The dynamic attraction level for virtual attractive objects is shown with orange circles. The pink area in front of each agent shows the useful field of view (UFOV) for that agent.

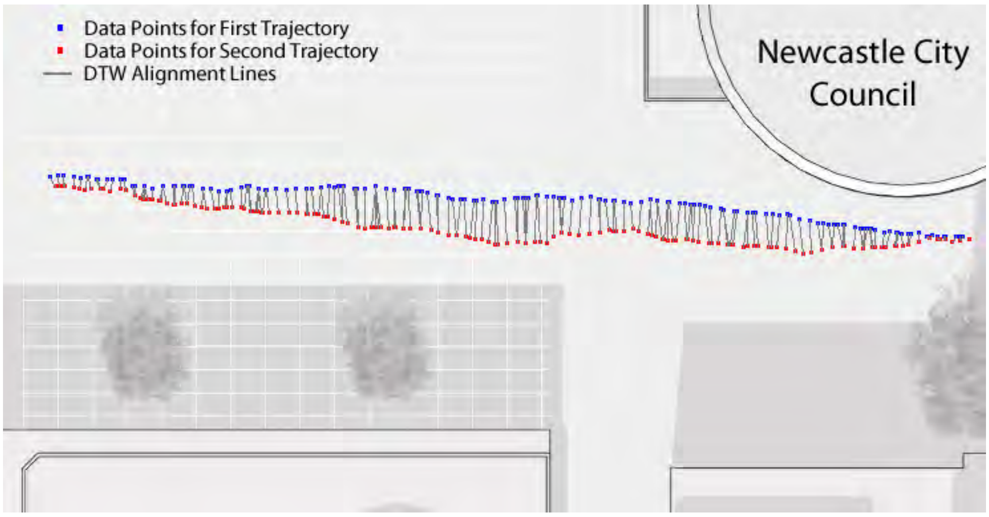

Figure 6.3: Trajectory alignment using the Dynamic Time Warping (DTW) technique; Illustration of the comparison of data points in two trajectories produced by simulation software presented in [93, 92].

haviour that is reflected by trajectories, the distance measure can be used alone or it could be employed in a kernel-based learning approach such as Support Vector Machines (SVMs). The ability of DTW in comparing variable-length input series makes it a tempting feature to use DTW in SVMs as a kernel function. In this regard, the Gaussian Dynamic Time Warping (GDTW) kernel function (as defined in Equation 6.1) has been used in SVMs as a kernel function [8].

$$k_{GDTW}(\mathbf{x}^r, \mathbf{y}^s) = \exp\left(-\frac{D(\mathbf{x}^r, \mathbf{y}^s)}{\sigma^2}\right) \tag{6.1}$$

where $\mathbf{x}^r$ is a time series with discrete time index varying between 1 and r, $\mathbf{y}^s$ is a time series with discrete time index varying between 1 and s, σ is the kernel width, and $D(\mathbf{x}^r, \mathbf{y}^s)$ is the DTW distance between the two time series,

$\mathbf{x}$ and $\mathbf{y}$, and it can be calculated recursively as [130, 182, 160]:

$$D(\mathbf{x}^r, \mathbf{y}^s) = ||x_r - y_s||_p + min \begin{cases} D(\mathbf{x}^{r-1}, \mathbf{y}^s) & delete, \\ D(\mathbf{x}^{r-1}, \mathbf{y}^{s-1}) & match, \\ D(\mathbf{x}^r, \mathbf{y}^{s-1}) & insert, \end{cases} \quad (6.2)$$

where $x_i \in \mathbb{R}^d$ is the ith element of time series x, $\mathbf{y}_i \in \mathbb{R}^d$ is the ith element of time series $\mathbf{y}$, and r and s are the length of $\mathbf{x}$ and $\mathbf{y}$ respectively.

Employing DTW in kernel-based learning methods is a challenging problem since the kernel function that uses DTW as its distance measure would not always be positive semi-definite (please refer to Chapter 5 for more details).

6.4 Analysing Using DTW-SVM

As discussed in Chapter 5, the resulting SVM classifier that uses GDTW as the kernel function may converge to a non-optimal separating hyperplane in comparing trajectories. To overcome this problem the DTW-SVM method is used to classify trajectory data. The DTW-SVM is a two-step classification method that finds the DTW distances between all trajectories in the first step (DTW matrix) and models the distances using standard SVMs in the second step.

The DTW matrix in DTW-SVM contains the DTW distances between all possible pairs of trajectories. For instance, in the matrix the value stored in the ith row and jth column is the DTW distance between the ith and jth trajectories. Each row of this matrix is used as an input point for a standard SVM classifier. Since the number of data points in the training set is known, the length of new data points obtained from the DTW matrix is fixed. Therefore, the DTW-SVM is capable of classifying data objects with variable length input series [88].

Although, the DTW-SVM can model trajectory data with different lengths, it encounters another problem. In the training phase of the DTW-SVM technique the input space of the SVMs classifier was represented using the the DTW distances between training trajectories. In the testing phase we still need to apply the same representation on each data sample in the test set. Thus for each trajectory in the test set we need to calculate the DTW distance between the testing trajectory and all trajectories in the training set. This makes the testing phase very slow and renders the DTW-SVM an unsuitable method for applications that need real-time classification results. To overcome the shortcomings of this method, GDTW-P-SVMs are used to classify trajectories.

6.5 Analysing Using DTW-P-SVMs

The GDTW-P-SVMs were presented as a new technique for sequential data analysis where each data object is characterised by a vector of numerical values that may have different lengths for different data objects. It employs Potential Support Vector Machines (P-SVM) [79] and GDTW to waive the fixed-length restriction of feature vectors in training and test data. The P-SVM optimisation problem was defined as Equation 6.3 and it can handle data and kernels that are neither positive definite nor square.

$$\begin{aligned} &\textit{minimise} && \tfrac{1}{2}\|\mathbf{X}_\phi^T\boldsymbol{\omega}\|^2 + C\mathbf{1}^T(\boldsymbol{\varepsilon}^+ + \boldsymbol{\varepsilon}^-) \\ &\textit{subject to} && \mathbf{K}^T(\mathbf{X}_\phi^T\boldsymbol{\omega} - \mathbf{y}) + \boldsymbol{\varepsilon}^+ \geq \mathbf{0} \\ &&& \mathbf{K}^T(\mathbf{X}_\phi^T\boldsymbol{\omega} - \mathbf{y}) - \boldsymbol{\varepsilon}^- \geq \mathbf{0} \\ &&& \mathbf{0} \leq \boldsymbol{\varepsilon}^+, \boldsymbol{\varepsilon}^- \end{aligned} \tag{6.3}$$

As a result, GDTW-P-SVMs do not suffer from the shortcomings of DTW-SVM and benefit from the DTW advantage of being able to compare trajectories with different lengths (Chapter 5).

6.6 Data Sets

6.6.1 Simulated Data Sets

The simulation software was run over the 2D plan of Wheeler Place in Newcastle (Australia). It contains several attractors with different levels of attraction

Table 6.1: Data sets details

Class	Attractor Existence	Crowd Attraction	Agent Behaviour		#Samples	
			See	Stop	Test	Train
1	NO	N/A	NO	NO	1500	465
2	YES	NO	YES	NO	1500	482
3	YES	NO	YES	YES	1500	492
4	YES	YES	YES	YES	1500	504
5	YES	YES	YES	NO	1500	497

such as the City of Newcastle Information Centre and Climate Meter, Juicy Beans Restaurant and Internet Cafe, a big public art work, the Civic Theatre and the Civic Theatre Restaurant (Figure 6.2). The entrances and exit points are limited to a few points. Each agent chooses one of these points as the start point and another one as the main destination (Figure 6.6)[93, 92].

In the scenario where an agent is close enough to an attractor, simultaneous changes in both α and its derivative ($d\alpha/dt$) signals can be observed ($R4$ in Figure 6.5). Whenever an attractor is close by, the agent's head rapidly corrects its position so that its gaze vector is in the direction of the attractor. As a result of this correction, a sudden change in α velocity ($d\alpha/dt$) can also be observed. If the category of the attractor is matched with the agent's need vector then the agent will move towards the attractor with the highest possible speed for the agent ($R2$ in Figure 6.5). The highest speed is defined for each agent in the agent's speed category. When the agent reaches the attractor it will focus on the attractor for a short period of time ($R3$ in Figure 6.5), before continuing its normal behaviour ($R1$ in Figure 6.5). The majority of data collected in the experiments with distractions showed distinct changes in $d\alpha/dt$ when the agent's head engaged a visually attractive object.

Table 6.1 describes a data set that is used to analyse simulated pedestrian spatial behaviour. The data set is obtained using the simulator software with different environmental and behavioural configurations. Five different classes are defined according to the configurations. In the first class the simulated pedestrians (agents) cannot see any attractor either because the attractors are not close enough to the flow of agents or there is no attractor in the

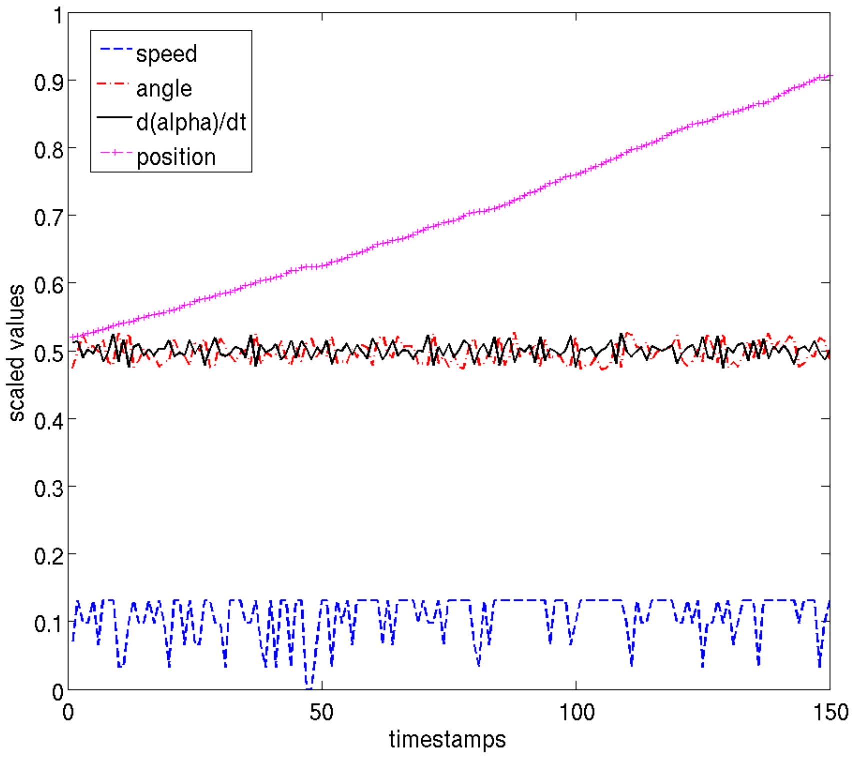

Figure 6.4: Spatial behavioural characteristics of an agent where there is no attractor close by. The values for each curve are scaled between 0 and 1. The agent's positions are projected from 2D to 1D. The curves of the angle and its derivative show a smooth behaviour with a small oscillation in these values where there is no attractor around the agent.

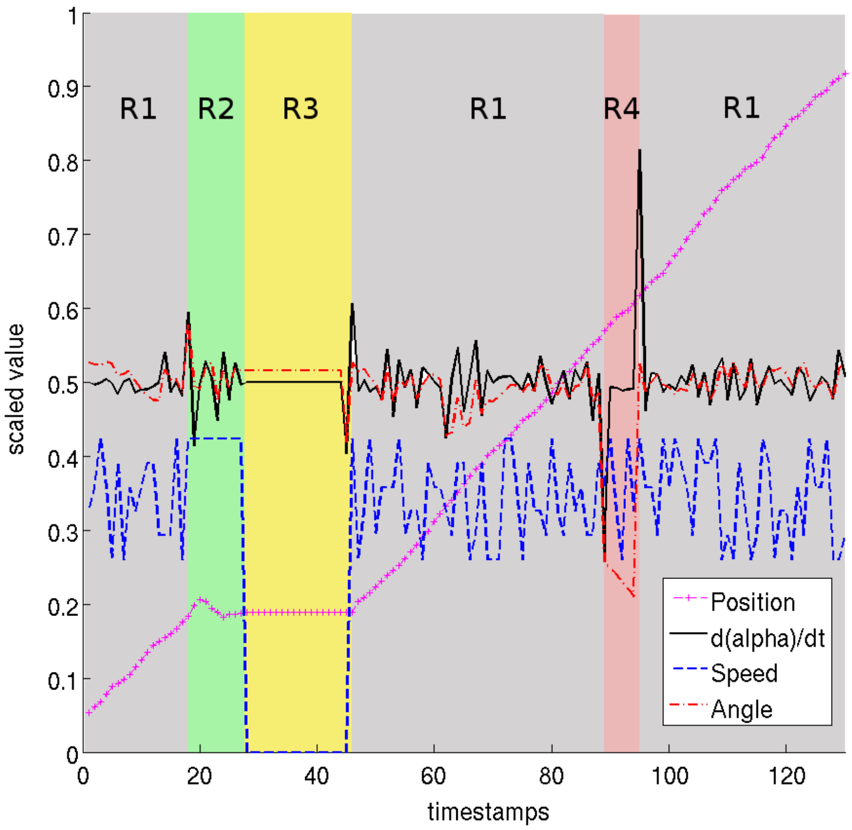

Figure 6.5: Spatial behavioural characteristics of an agent with the existence of some attractors. Different behaviours are shown in different colours and are labeled with $R1, R2, R3$, and $R4$. $R1$s show regions where the agent cannot see any attractor and it shows casual behaviour. $R2$s show regions where the agent is becoming attracted to an attractor. $R3$s show regions where the agent is attracted to an attractor. $R4$s show regions where the agent is not becoming attracted to an attractor but it pays visual attention to the attractor. Spikes in curves of the angle and its derivative clearly distinguish them from other curves in $R4$.

Figure 6.6: Some examples of simulated trajectories mapped onto the real-world image. Each trajectory is shown in a different colour. The colours are chosen randomly.

experimental space. In previous chapters this class of behaviours is referred to as *normal* behaviours and the rest as attracted/abnormal behaviours. In this section, however, the simulated behaviours are more precisely categorised into four additional classes as well as the normal behaviour. The new classes not only describe agents' behaviour in the presence or absence of attractive objects, but they can also describe the impacts of crowd attraction on agents' spatial behaviour. The classes show the impact of an agent's need vector by distinguishing between agents that can only see an attractor and those that get attracted to an attractor and spend some time reading/visiting the attractor ("stop" in Table 6.1).

The following settings for the simulation software were used to generate the simulated agents:

- Maximum number of agents in the scene: 50
- Number of object categories: 5
- Number of agent's need categories: 5
- Number of agent's speed categories: 5
- Number of agent's sight categories: 4
- Average agent's speed distribution: Normal
- Environmental configurations: Scenario two (Figure 6.2) as described in Chapter 2.

6.6.2 Real-World Data Sets

To collect real-world trajectory data sets, the pedestrian detection and tracking system described in Chapter 3 is used. An optical camera is installed on a fixed-platform at Wheeler Place to collect video data that contain pedestrians' movements. The videos are captured during a seven week observation period (every Thursday from 8am to 5pm). The camera is installed on the same place with the same horizontal and vertical angels for all the experiments. During the recording period 1200 pedestrians were detected. Among them 830 pedestrians

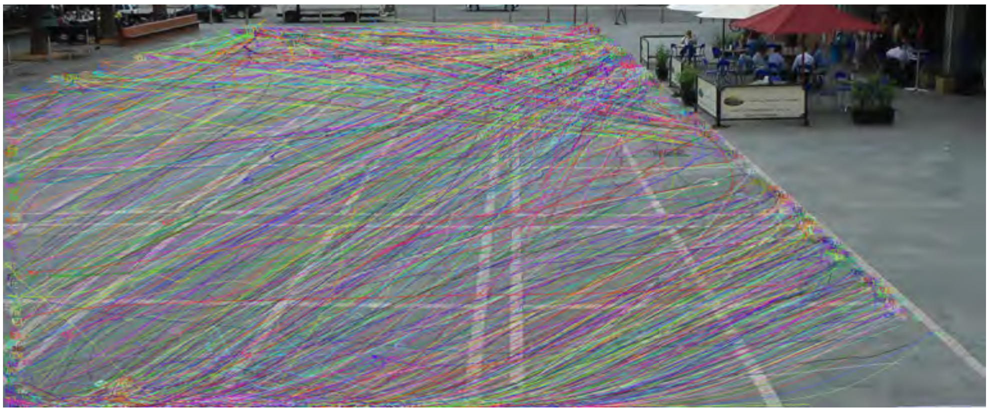

Figure 6.7: Tracked pedestrians using the pedestrian detection and tracking system. Each track is shown in a different colour. The colours are chosen randomly.

were tracked at Wheeler Place and the rest were outside the area of interest. either walking along Hunter Street (Figure 6.12) or not entering the main grid of Wheeler Place. Figure 6.7 shows the collected trajectories.

6.7 Experimental Results

6.7.1 Simulated Data Classification

To analyse simulated pedestrian spatial behaviour, the data set described in Table 6.1 is used. The data set is split into two data sets according to the crowd attraction activation. The first data set contains class 1, class 2 and class 3, for which crowd attraction is disabled, and the second data set contains class 1, class 4, and class 5 with enabled crowd attraction.

Table 6.2 and Table 6.3 show classification error rates for simulated behavioural characteristics. In both tables the classification error rates of three trajectory analysis methods are compared. To perform the data classification the LIBSVM toolbox is used and a one-vs-one strategy along with a classifier is employed. A five-fold cross-validation optimisation technique with a leave-one-out policy is utilised to tune the hyperparameters (C, γ and kernel

Table 6.2: Classification error rates without the impact of crowd attraction. Three classes of behaviour are used; class 1: the attractors are far away from the agents' flow dynamics, class 2: the agents only see the attractors, and class 3: the agents not only see but also get attracted to the attractors. Columns labelled with 'see' include class 1 and class 2, columns labelled with 'stop' include class 1 and class 3, and columns labelled with 'total' include class 1, class 2 and class 3 of behaviour. GDTW-P-SVMs have the lowest classification error rate when analysing the angle and its derivative, which are shown in bold.

Features	DTW (1NN)			GDTW-P-SVMs			DTW-SVM		
	See	Stop	Total	See	Stop	Total	See	Stop	Total
Trajectory	N/A	0.153	0.448	N/A	0.014	0.318	N/A	0.157	0.449
Speed	N/A	0.138	0.442	N/A	0.030	0.333	N/A	0.083	0.400
Angle(α)	0.049	0.421	**0.211**	0.015	0.037	**0.021**	0.085	0.207	**0.150**
$d(\alpha)/d(t)$	0.159	0.201	**0.160**	0.019	0.021	**0.021**	0.037	0.262	**0.121**

Table 6.3: Classification error rates with the impact of crowd attraction. Three classes of behaviour are used; class 1: the attractors are far away from the agents' flow dynamics, class 4: the agents only see the attractors, and class 5: the agents not only see but also get attracted to the attractors. Columns labelled with 'see' include class 1 and class 4, columns labelled with 'stop' include class 1 and class 5, and columns labelled with 'total' include class 1, class 4 and class 5 of behaviour. GDTW-P-SVMs have the lowest classification error rate when analysing the angle and its derivative, which are shown in bold.

Features	DTW (1NN)			GDTW-P-SVMs			DTW-SVM		
	See	Stop	Total	See	Stop	Total	See	Stop	Total
Trajectory	N/A	0.292	0.501	N/A	0.031	0.341	N/A	0.114	0.408
Speed	N/A	0.284	0.471	N/A	0.087	0.372	N/A	0.128	0.421
Angle(α)	0.085	0.195	**0.142**	0.015	0.093	**0.082**	0.043	0.297	**0.167**
$d(\alpha)/d(t)$	0.159	0.191	**0.174**	0.019	0.042	**0.031**	0.000	0.290	**0.167**

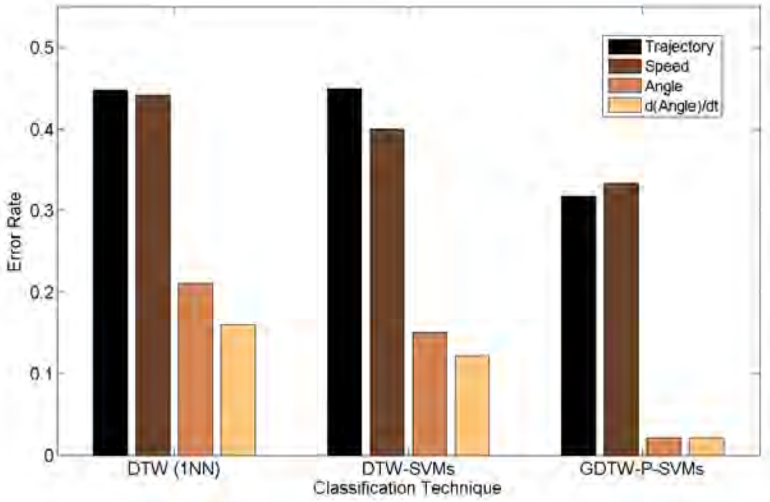

Figure 6.8: Classifier comparison without the impact of crowd attraction. Classifying the angle and its derivative results in lower classification error rates. The GDTW-P-SVM classification technique has the highest accuracy among other classification techniques when analysing spatial behavioural characteristics.

parameters). The test sets and training sets are two completely separate sets of data. The training sets are used for obtaining the models and tuning hyper-parameters. The test sets are only used to obtain the classification error rates reported in Table 6.3 and Table 6.2.

The classification error rates are shown for four behavioural characteristics, which include: trajectory, speed, the angle between the movement direction and gaze vector (α), and the derivative of α with respect to time (t). The error rates are reported for different situations:

- *See*: Agents in this situation could only see an attractive object and this does not attract their their interest. The error rates reported in "See" columns are the ratio of the "number of these agents that classified correctly" to "the total number of these agents plus those that showed a normal behaviour". In other words, the training set and testing sets include only class 1 and class 2 data sets for Table 6.2 and class 1 and

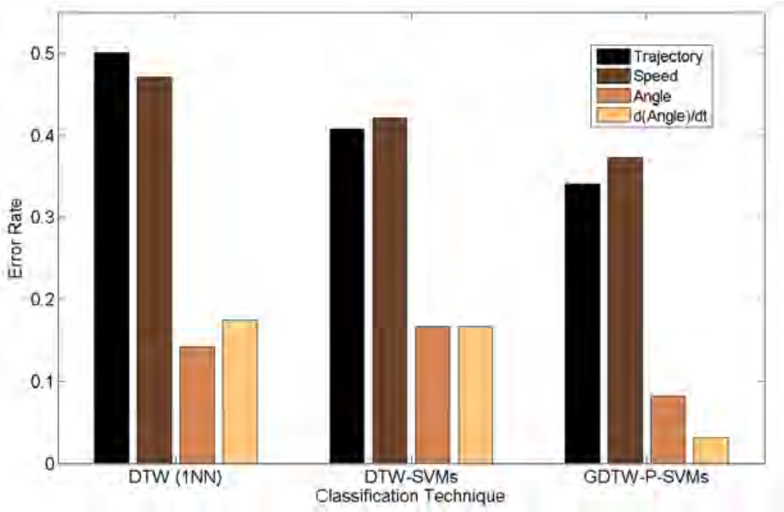

Figure 6.9: Classifier comparison with the impact of crowd attraction. Crowd attraction implies noise on data and this results in higher error rates compared to Figure 6.8. The GDTW-P-SVM classification technique still has the lowest error rate among the comparing classifiers.

class 5 for Table 6.3, as described in Table 6.1.

- *Stop*: Agents in this situation were not only able to see an attractor but also get attracted to the attractor and spend some time examining the attractor. The error rates reported in "Stop" columns are the ratio of the "number of these agents that classified correctly" to "the total number of these agents and those that showed a normal behaviour". In other words, the training set and testing sets include only class 1 and class 3 data sets for Table 6.2 and class 1 and class 4 for Table 6.3, as described in Table 6.1.

- *Total*: These columns report the total classification error rates for all three classes of agents ("only see", "get attracted by an attractor", and "the normal behaviour when there is no attractor close by"). The total error rates are the ratio of the "number of agents that were classified correctly" to "the total number of agents". Here, the training set and testing sets include classes 1, 2 and 3 data sets for Table 6.2 and classes 1, 4 and 5 for Table 6.3, as described in Table 6.1.

In the simulation software it is assumed that if the object is not of interest to an agent, the agent only sees the object without getting attracted by it. Thus no significant change in the agent's trajectory and speed will be observed. In this case, analysing trajectory or speed data cannot reveal the required information about the agent's spatial behaviour. This behaviour is shown by "N/A" in Table 6.3 and Table 6.2. On the other hand, crowd attractions require virtual attractive objects (defined in [92]) that themselves need attracted agents. Since in class 1 we assumed that attractors are far away from agent flow, crowd attraction cannot be happening and this situation is also shown as "N/A" in Table 6.1.

While analysing the trajectory and speed data cannot reveal the desired information about the agent's spatial behaviour, the angle and its derivative can distinguish between agents that saw an attractor and those with normal spatial behaviour. This is shown in both Table 6.2 and Table 6.3. The *total* columns in the tables contain classification error rates for all three classes (see, stop and normal). In this column, all three classification techniques have lower error rates for classifying the angle and its derivative. This is because in the scenes with a high density of agents, a collision avoidance algorithm that was used in the simulator alters agent's trajectories. In cases where the scene is too crowded the simulator also implies changes to agents' speed. Consequently, the obtained speed and location trajectories do not purely result from the influence of attractors on the agent's behaviour.

By enabling the crowd attraction in the simulator, agents not only follow their needs but also get attracted by other agents who are forming a social group. Although the frequency of social group occurrence is low, social groups imply noise on the trajectory data. This explains why most of the error rates in Table 6.3 are higher than those in Table 6.2.

Figure 6.8 and Figure 6.9 are two other representations of the total columns. As shown, GDTW-P-SVMs have the lowest error rate among the classifiers. Classifying the angle and its derivative have shown lower error rates compared to trajectory and speed of movement in all three classifiers. Including the impact of crowd attraction in the simulation leads to higher error rates than when omitting it.

6.7.2 Real-World Data Classification

The trajectories extracted from videos are used as the input space of the GDTW-P-SVMs to learn pedestrians' patterns of movements. To analyse pedestrians' reactions to the attractive objects the following two classes are learnt by the classifier.

1. *Casual/normal behaviour*: Includes the trajectories of pedestrians who have entered into and exited from Wheeler Place without being distracted.

2. *Attracted behaviour*:Includes the trajectories of pedestrians who have become attracted by an attractive object and are distracted from their casual direction of movement.

Figure 6.10 shows the trajectories with the casual/normal behaviour in black and attracted trajectories in white. Table 6.4 shows the characteristics of the training and testing data sets. The number of negative and positive samples in the data set is not balanced. To cope with this problem the weight balancing technique suggested by Vapnik is employed [181].

Figure 6.11 compares Receiver Operating Characteristic (ROC) curves for GDTW-SVMs, DTW-SVMs and GDTW-P-SVMs using the real-world trajectory data set. The False Positive Rate (FPR) is defined as the fraction of the false negatives out of total negatives, and the True Positive Rate (TPR) is defined as the fraction of the true positives out of total positives. The Area Under the Curve (AUC) for GDTW-SVMs, DTW-SVMs, and GDTW-P-SVMs are 0.6401, 0.7780 and 0.8915 respectively (the closer the value of AUC to 1, the higher the accuracy of classification). The ROC curves show that GTDW-P-SVMs have higher accuracy in classifying positive and negative samples than other methods.

Figure 6.10: Tracked pedestrians using the pedestrian detection and tracking system. White trajectories belong to pedestrians who have been attracted to the Cafe or the Opening area. The start point of each trajectory is shown with its number.

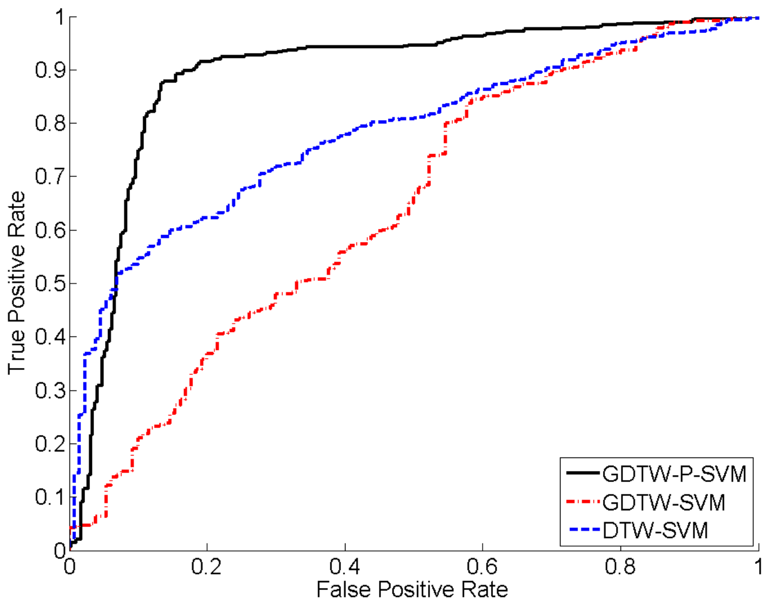

Figure 6.11: Receiver Operating Characteristic (ROC) curves for trajectory classification of real-world data using GDTW-SVM, DTW-SVMs and GDTW-P-SVMs. The Area Under the Curve (AUC) for GDTW-SVMs, DTW-SVMs, and GDTW-P-SVMs are 0.6401, 0.7780 and 0.8915 respectively.

6.7.3 Simulated vs. Real-World Trajectory Analysis

To compare the classification results using the real-world and simulated trajectories, the simulation software is employed to generate new trajectory data sets that are compatible with the real-world restrictions. The following configurations are altered in simulation to generate the new data sets:

- The tracking system cannot track pedestrians in the Cafe so trajectories are extracted outside the cafe only. In the simulation the same restriction is applied. When an agent gets attracted to an attractive object the corresponding simulated trajectory will be finished.
- The Cafe and the opening area are the only attractive objects with extractable attracted trajectories. Trajectories that are attracted by other attractive objects are not extractable because they are occluded with an obstacle, for example, trees, benches and seating areas for the cafe.
- The trajectories generated by the simulation software are cropped to match the field of view of the camera installed at Wheeler Place.
- Number of generated trajectories in each class of behaviour in the training and test sets are controlled by the *object category* and the *agent need* vectors (described in [92]).

Figure 6.12 shows the the result of mapping the real-world trajectories to the plan of Wheeler Place. In this figure, locations where pedestrians have shown more interest to cross over are shown with a darker red colour. As shown in this figure, there are two locations where pedestrians have shown more interests to go. An opening area that leads to a car park, and the Cafe. Figure 6.13 illustrates the simulated trajectories using the same representation method. As shown in these two figures, while the generated trajectories are smooth and do not contain noise, the extracted trajectories from real-world video contain pedestrian detection and tracking noise. It will be shown that the proposed method for behavioural analysis can handle noisy as well as non-noisy data sets.

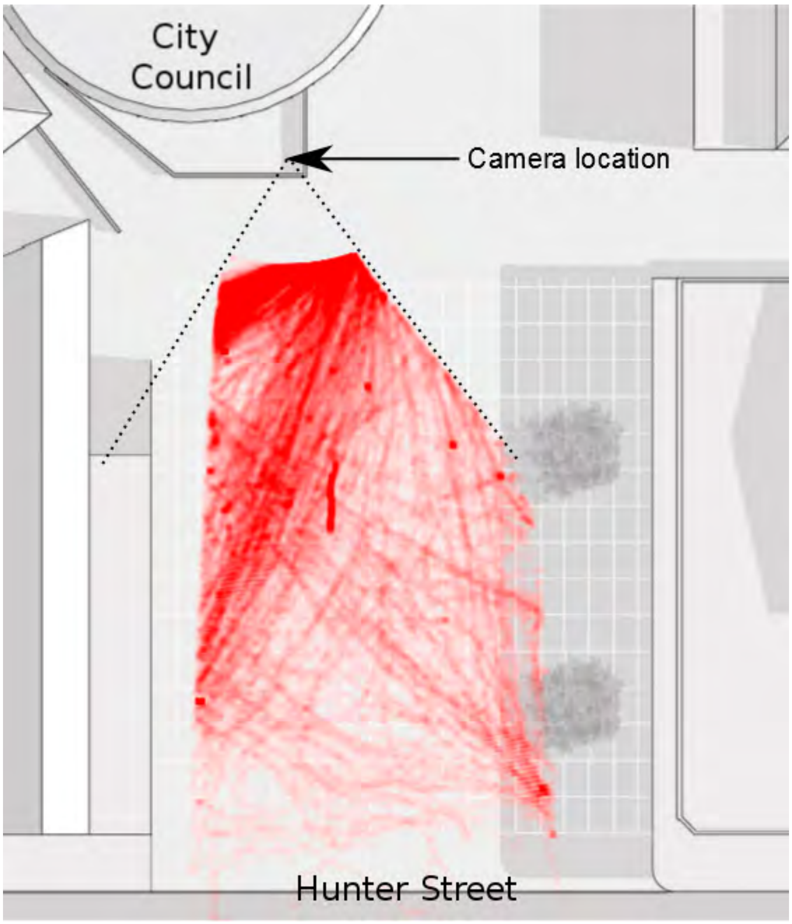

Figure 6.12: Representation of mapping the real-world trajectories to the plan of Wheeler Place. Locations where pedestrians have crossed more often are shown with darker red colours. Dashed lines indicate the field of view of the camera.

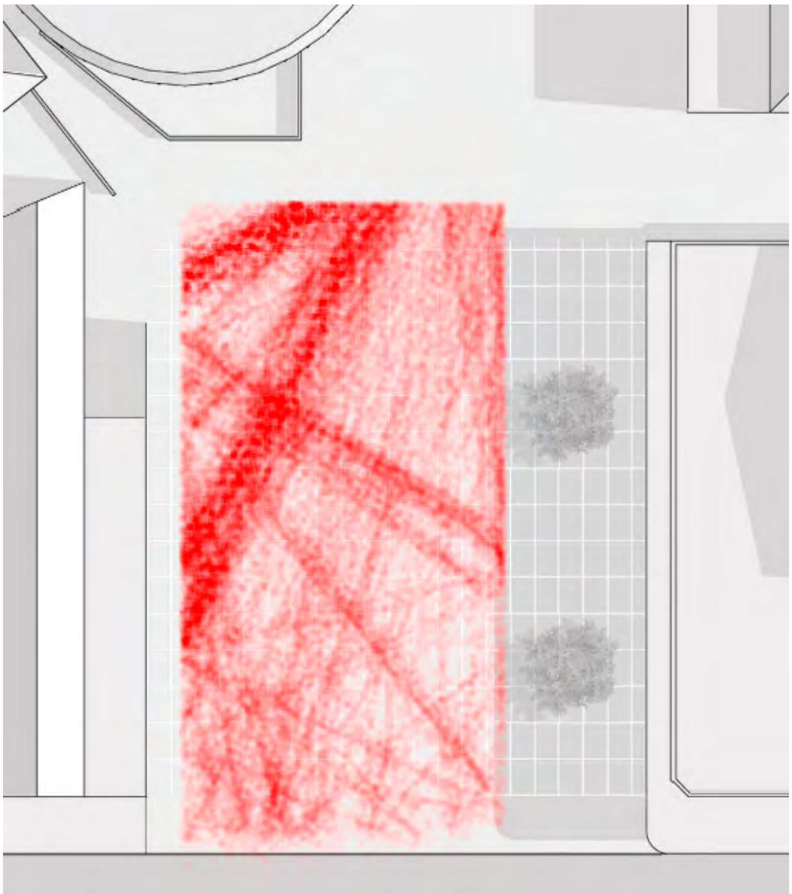

Figure 6.13: Representation of the simulated trajectories at Wheeler Place. Locations where simulated pedestrians have crossed more often are shown with darker red colours.

Table 6.4: The characteristics of training and testing data sets used in the analysis of pedestrian behaviour at Wheeler Place.

Collection	Dataset	#Normal	#Attracted	Total
Real-World	Train	148	85	233
	Test	391	206	597
	Total	539	291	830
Simulation	Train	150	85	235
	Test	380	215	595
	Total	530	300	830

To perform data classification the LIBSVM [28] and the P-SVM [79] toolboxes are used for implementing the two-step DTW-SVM and the GDTW-P-SVMs respectively. To ensure a fair comparison, the *hyperparameter* selection procedure was equal in all methods. Best values are selected from a generated hyperparameter set to minimise the error rate in the training phase. For classifying data the Radial Basis Function (RBF) kernel with Euclidean Distance (ED) is employed in SVMs and the Gaussian Dynamic Time Warping (GDTW) function is used in GDTW-P-SVMs. The five-fold cross-validation technique is employed to *tune* the classifier hyperparameters (C and γ). Two separate data sets are used for training and testing. The tuning is performed using the training set only. More details about the tuning are provided in Chapter 5.

Figure 6.14 compares the accuracy of the classification of the trajectories using 1-nearest neighbour (1NN) DTW, GDTW-SVMs, DTW-SVMs and GDTW-P-SVMs for simulated and real-world data sets described in Table 6.4. The GDTW-P-SVMs classification technique shows higher accuracy than other techniques. The classification accuracy is always higher when using simulated trajectory data except for GDTW-SVMs. As discussed in [88], the positive definiteness of the GDTW depends on the training data sets and therefore the convergence of GDTW-SVMs to the optimal solution is not always guaranteed. In the experiments the GDTW-SVMs showed the lowest classification accuracy.

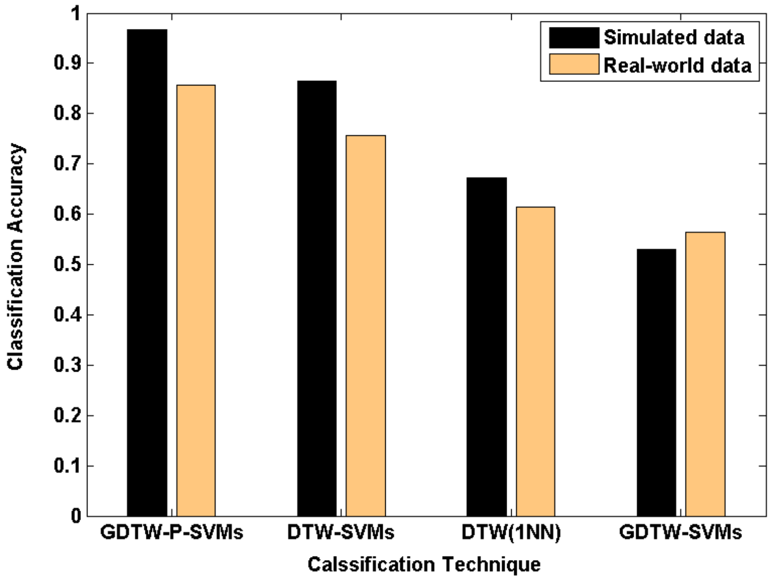

Figure 6.14: Classifier comparison using real-world and simulated trajectory data sets. The classification of simulated data is always more accurate than the real-world data except for GDTW-SVMs. The GDTW-P-SVMs classification technique shows the highest accuracy.

6.8 Summary and Discussion

A system to analyse pedestrians' spatial behaviour in an architectural environment was developed. The system utilised a new classifier that is capable of handling input series with different lengths. The classifier was presented in Chapter 5. Different spatial behaviours were described using simulated and extracted characteristics. The simulated characteristics were generated using the proposed method described in Chapter 2. Instead of segmenting each trajectory data sample into several fixed-length segments prior to analysing them, the method analysed the whole trajectory sequence as a single input. Several behaviour patterns were learnt using the simulated and real-world data sets. GDTW-P-SVMs, DTW-SVMs, DTW(1NN) and DTW-SVMs classification techniques were employed to distinguish between the behaviour patterns. Although DTW-SVMs were able to classify trajectory data with acceptable error rates, they are not able to provide the classification results in real-time. The shortcomings of using DTW-SVMs, discussed in Chapter 5, were also highlighted using the behavioural data sets. It was argued that the classification accuracy using the real-world data is lower than using the simulated data. GDTW-P-SVMs classified the behavioural data sets with the highest accuracy compared to the other algorithms and had the benefit of a fast testing phase.

Part III

Conclusions

Chapter 7

Conclusions and future works

Contents

This chapter presents the main conclusions drawn from this work. Evidence is provided for achievement of the aims, which were first introduced in Chapter 1. Future work is proposed to address the limitations of this research.

7.1 Conclusions

This thesis has presented an analysis method for pedestrians' spatial behaviour in urban spaces. The method is capable of distinguishing between different behavioural classes based on characteristics extracted from simulated and real-world pedestrians. Simulation software for modelling spatial behaviour, a new classification method to handle input series with different lengths, and an analysis approach using the classification method were presented in this thesis.

7.1.1 Behavioural Features Collection

Simulation software for modelling pedestrian spatial behaviour in a 2D plan of an urban space was presented in Chapter 2. The simulation is multi-agent-based, which adds two unique features to the conventional model: the gaze vector and attractor objects. The benefits of this new approach become especially clear in cases of complex spatial arrangements where changing the configuration of walking environments (and thus adapting designs) was possible. A software system was developed to read plans, extract boundaries and obstacles, and generate urban and pedestrian models. The simulation was then run using the models, and the analysis of the behavioural characteristics of simulated agents was performed.

To model advanced spatial features of the pedestrians and their environments, the dependency of attention and visual gaze direction was discussed. An innovative model is proposed to simulate goal-driven attention and stimuli-driven attention with the "agent's need vector" (ANV) and the "object level of attraction". The proposed model includes group formation and the effects of group size on directing a pedestrian's attention. The simulation was run for a real-world space, Wheeler Place in Newcastle. Different scenarios with different configurations, and the impacts of considering crowd attraction on pedestrian behaviour, were presented and discussed in Chapter 2. The experimental results demonstrated that the new system could provide significant support for understanding how changes in the configuration of the physical/visual built environment are reflected by measurable changes in agents' behaviour.

To obtain behavioural characteristics from real-world video data, a pedestrian detection and tracking system was presented in Chapter 3. The system employed an optical camera installed on a fixed platform to extract trajectory data at Wheeler Place. The system consisted of three parts: *i*) background detection *ii*) pedestrian detection and *iii*) pedestrian tracking. In the background detection we employed a background subtraction method with an upgrading strategy. For recognising pedestrians, depth and shape features were extracted and used for training classifiers. For tracking the detected pedestrians, pixel, depth, and motion features were used to find the same pedestrian in a stack of frames. The presented approach showed the capability of GDTW-P-SVMs in classifying extracted features for pedestrian detection. The simulated and extracted behavioural characteristics have been used in the proposed analysis system.

7.1.2 Behaviour Analysis

The simulated behavioural data and the real-world extracted trajectories are used to classify agent/pedestrian behaviour. To classify trajectories a kernel-based classification technique was proposed and used. Support Vector Machines are a kernel-based classification technique that is capable of calculating the optimal separating hyperplane in the feature space. The original SVM was a linear classifier. However, Vapnik suggested the use of the kernel trick. If the kernel used is a radial basis function (RBF), the corresponding feature space is a Hilbert space of infinite dimension. To use the SVM as the classifier the kernel employed to map the input space to feature space should be positive semi-definite (PSD). The standard SVMs with a PSD kernel can only classify feature vectors with fixed-length.

The behavioural features generated and extracted using the simulation and tracking systems may have different lengths. Therefore the standard SVMs with a PSD kernel defined in Hilbert Space cannot be used with the behavioural features. To solve this problem a new classification technique, GDTW-P-SVMs, was introduced for sequential data analysis where each data object is characterised by a series of numerical values that may have different lengths for different data objects. The new technique is a maximum margin method for the construction of classifiers with variable-length input series. The well-known DTW algorithm was utilised to provide an elastic distance measure that is able to compare variable-length input series. GDTW-P-SVMs were compared with the two-step DTW-SVM method, where training data were required in the testing phase as well as in the trained models. Although DTW-SVMs were able to classify trajectory data with acceptable error rates, they are not able to provide the classification results in real-time as the testing phase for this technique is too slow for problems that have a big training set.

GDTW-P-SVMs were proposed to overcome the shortcomings of the SVMs (with RBF kernel) and DTW-SVM by altering the kernel function in P-SVMs using DTW. As a result, GDTW-P-SVMs could handle data and kernel matrices that were neither positive definite nor square, and it could also be applied to data with variable-length input series. A comparison of the performance of GDTW-P-SVMs and several other classification techniques was performed with several real-world data sets from the UCR Time Series Classification/Clustering page, the GeoLife trajectory data set, and the character trajectory from the UCI repository. The data sets included data with both variable and fixed-length input series. Classification error rates and ROC curves showed that GDTW-P-SVMs can converge into the optimal separating hyperplane with maximum margin in classification problems with fixed-length feature vectors. A statistical evaluation was provided in Chapter 5 showing that GDTW-P-SVMs improved the classification accuracy significantly.

Chapter 5 discussed the reasons why DTW cannot be used as a kernel distance measure in standard SVMs. In the case of variable-length data inputs, GDTW-P-SVMs significantly outperformed other existing methods by two main advantages, it has significantly lower classification error rates and works directly with the variable-length input series. The second advantage becomes more important when the extraction of fixed-length feature vectors is not feasible or when using fixed-length segments of data objects fails to describe the relationship between data objects properly.

A pilot system to analyse pedestrians' spatial behaviour in an architectural environment was developed. The system utilised a new classifier that is capable of handling input series with different lengths. The classifier was presented in Chapter 5. Different spatial behaviours were described using simulated and extracted characteristics. The simulated characteristics were generated using the proposed method described in Chapter 2. GDTW-P-SVMs, DTW-SVMs, DTW(1NN) and DTW-SVMs classification techniques were employed to distinguish between the behaviour patterns. Although DTW-SVMs were able to classify trajectory data with acceptable error rates, they are not able to provide the classification results in real-time. The shortcomings of using DTW-SVMs, discussed in Chapter 5, were also highlighted using the behavioural data sets. It was argued that the classification accuracy using the real-world data is lower than using the simulated data.

The proposed classification technique was used in a behaviour analysis system. The system was developed to analyse pedestrian spatial behaviour in an architectural environment. The system employed the GDTW-P-SVMs classifier that is capable of accepting data objects with different length input series, such as trajectory-based sequential data sets. Instead of segmenting each trajectory data sample into several fixed-length segments prior to analysing them, the method analysed the whole trajectory sequence as a single input. Several behaviour patterns were learnt using the simulated and real-world data sets. It was argued in Chapter 6 that the classification accuracy using the real-world data is lower than using the simulated data. GDTW-P-SVMs classified the behavioural data sets with the highest accuracy compared to the other algorithms and had the benefit of a fast testing phase.

7.2 Main Research Contributions

The main contributions of this research are made towards achieving the aims described in Chapter 1. The following lists the main contribution regarding each of the aims:

1. *Behaviour simulation*: The main purpose of this aim was to introduce a simulation system that can be used for evaluating an urban design based on pedestrians' reactions to their immediate surroundings. To this end, a new simulation system is proposed that includes a multi-agent-based simulation procedure, an urban model and a pedestrian spatial behaviour model. The models introduce the simulation of several new physical attributes of visually attractive objects in an urban space. These include Virtual Attractive Objects, dynamic level of attraction, crowd attraction, group formation based on visual desires, and goal-driven and stimulus-driven attentions. The experimental results using the simulation software system demonstrated that the new system can provide significant support for understanding how changes in the configuration of the physical/visual built environment are reflected by measurable changes in agent behaviour.

   Therefore, the main contribution of the simulation software presented in Chapter 2 is introducing a multi-agent-based spatial behaviour simulation software that provides a unique support to architects for assessing the impacts of planned urban spaces on pedestrian behaviour. Furthermore, the simulation software extends the literature by introducing new urban and pedestrian models that can generate complex behavioural characteristics to fulfill the requirements of assessing an architectural design. The contributions of the simulation software have been published in [90, 91, 92, 93].

2. *Variable-length time series classification*: The main purpose of this aim was to develop a classification technique for time series with different length feature series. To achieve this the existing algorithms, which use Support Vector Machines (SVMs) and different distance measures, were examined. In particular, the usage of Dynamic Time Warping (DTW) as a distance measure in Radial Basis Kernels (also known as Gaussian kernels) was examined. The kernel with DTW is called Gaussian Dynamic Time Warping (GDTW). DTW was used as it can measure the similarities between two input series with different lengths. The use of GDTW in standard SVMs and its positive definiteness was discussed. It was shown that the GDTW in SVMs may result in a non-PSD kernel and therefore the existence of a Reproducing Kernel Hilbert Space is not guaranteed. To cope with this challenge a new method that uses GDTW in an existing classification technique called Potential Support Vector Machines (P-SVMs), which can handle non-positive kernel matrices was proposed. The proposed method is called GDTW-P-SVMs and was compared to several benchmarked data sets with fixed-length and variable-length data objects. In Chapter 5 it was shown that the proposed coupling method significantly improved the classification accuracies using the data sets.

   Therefore, the major contribution regarding the time-series classification presented in Chapter 5 was introducing GDTW-P-SVMs as a new coupling technique that is capable of classifying variable-length and fixed-length data objects with a significantly improved accuracy. The trainability of GDTW-P-SVMs and their performance compared to existing methods has been presented in [88, 94].

3. *Analysis of spatial behaviour in urban spaces*: The aim was to develop a system that automatically analyses behavioural characteristics in urban spaces. To achieve this the GDTW-P-SVMs were employed on spatial behaviour data generated and extracted using the simulation and tracking software systems. The classification technique learns pedestrians' spatial behaviour patterns in a simulated and real-world urban space. Chapter 6 showed that GDTW-P-SVMs can provide the highest classification accuracy using the data sets when compared with other existing methods. As a result of using GDTW-P-SVMs, the system waived the need for using any segmentation method that was required to generate fixed-length feature vectors.

   The main contribution toward this aim was the introduction of a spatial behaviour analysis system that learns patterns of behaviour using the whole sequence of data series as a single input to increase the classification accuracy. The spatial behaviour analysis for simulated and real-world data sets has formed the basis of the following publications: [94, 89].

Overall, the novel contributions of this thesis are a pedestrian spatial behaviour simulation software with the application of architectural design evaluation, a classification technique that has an improved classification accuracy, and an analysis system that learns patterns of pedestrians' spatial behaviour using the classification technique in a simulated and real-world urban space.

7.3 Future Works

7.3.1 Simulation and Feature Extraction

To understand the generalisability of the simulation software to model spatial behaviour in urban spaces other than Wheeler Place, it could be applied to range of urban spaces. Furthermore it could also be applied to other architectural spaces including shopping malls, exhibitions, indoor and outdoor areas. The research should give consideration to the quality of the approach through assessment against the criteria used in this thesis. Taking future work in this direction would extend the analysis of the approach's performance to these different spaces, thus augmenting the guidance available to architects and designers who need to find the impacts of attractors on human spatial behaviour.

In the simulation it is assumed that the physical characteristics of an environmental setting influence our attitudes and actions more than biological or cultural traits. Including the cultural impacts on human behaviour can improve the simulation results. This includes finding ways to measure socio-cultural variables and analysing the impacts of the spatial environment on these variables. By comparing the resulting models and the proposed model, the impact of socio-cultural variables on spatial behaviour can be simulated.

Another opportunity to enhance the urban and pedestrian models may be to investigate the characteristics of attractive objects and their conspicuity area. The research could include an investigation of different classes of attractive objects based on the needs of pedestrians. The classes can be defined using a set of features that describe the needs. This will provide more accurate information on the agents' reactions to attractive objects.

7.3.2 Analysis Approach and GDTW-P-SVMs

A future step to compare the GDTW-P-SVMs is to adopt other elastic distance measures in the Radial Basis Function (RBF) and employ the new kernel in potential support vector machines. The distance measures may include Longest Common Sub-Sequence (LCSS) and Edit Distance with Real-Penalty. Furthermore, the research could contain an experimental justification for trainability of the proposed classifiers, which include testing the new classifiers using benchmarked variable and fixed length data sets.

Future work may include applying the proposed classification method for other possible classification problems that deal with data objects of different lengths than the trajectory analysis. This may result in simplifying feature extraction in pattern recognition in digital images. The research should investigate the classification accuracy of the proposed approach by adopting it in different pattern recognition tasks in digital images.

A future work may include further investigation in the analysis system by adopting the system in different urban spaces with a wider variety of attractive objects such as shops, open areas, and architectural and historical buildings. Furthermore, the investigation may include adopting the analysis system in indoor as well as outdoor spaces such as museums, shopping malls and exhibitions. Potential applications of the analysis system may include the analysis of pedestrians' behaviour as a group of people for managing crowds, predicting human behaviour in emergency situations, analysis of customers' reactions to a particular object in a shop.

Bibliography

[1] Mathias M. Adankon and Mohamed Cheriet. Optimizing resources in model selection for support vector machine. *Pattern Recognition*, 40(3):953–963, 2007. (Cited on page 108.)

[2] Shawkat Ali and Kate Smith-Miles. Improved support vector machine generalization using normalized input space. In Abdul Sattar and Byeong-ho Kang, editors, *AI 2006: Advances in Artificial Intelligence*, volume 4304 of *Lecture Notes in Computer Science*, pages 362–371. Springer-Verlag Berlin Heidelberg, 2006. (Cited on page 123.)

[3] Bruce Anderson, Jeffrey Jackson, and Meera Sithara. Descartes' rule of signs revisited. *The American Mathematical Monthly*, 105(5):447–451, 1998. (Cited on page 114.)

[4] Gianluca Antonini, Michel Bierlaire, and Mats Weber. Discrete choice models of pedestrian walkig behavior. *Transportation Research Part B: Methodological*, 40(8):667–687, 2006. (Cited on page 19.)

[5] Michael Argyle and Janet Dean. Eye-contact distance and affiliation. *Sociometry*, 28(3):289–304, 1965. (Cited on page 33.)

[6] Amy H. Auchincloss and Ana V. Diez Roux. A new tool for epidemiology: the usefulness of dynamic-agent models in understanding place effects on health. *American Journal of Epidemiology*, 168(1):1–8, 2008. (Cited on pages 143 and 144.)

[7] S. Bahadori, L. Iocchi, G.R. Leone, and D. Nardi. Real-time people localization and tracking through fixed stereo vision. *Applied Intelligence*, 26:83–97, 2007. (Cited on page 62.)

[8] Claus Bahlmann, Bernard Haasdonk, and Hans Burkhardt. On-line handwriting recognition with support vector machines - a kernel approach. In A. Denise Williams, editor, *The 8th International Workshop*

on Frontiers in Handwriting, IWFHR'02, pages 49–54, Washington, DC, USA, 2002. IEEE Computer Society. (Cited on pages 72, 104, 109, 113 and 149.)

[9] Mark Baldassare. Human spatial behavior. *Annual Review of Sociology*, 4:29–56, 1978. (Cited on page 33.)

[10] K. Ball and C. Owsley. The useful field of view test: a new technique for evaluating age-related declines in visual function. *Journal of American Optometric Association*, 64(1):71–79, 1993. (Cited on pages 26 and 34.)

[11] M. Ballerini, N. Cabibbo, R. Candelier, A. Cavagna, E. Cisbani, I. Giardina, and V. Lecomte. Interaction ruling animal collective behaviour depends on topologocal rather than metric distance. In *Proceeding of National Academy of Sciences*, volume 105, pages 1232–1237, 2008. (Cited on page 33.)

[12] R. Barlovic, T. Huisinga, A. Schadschneider, and M. Schreckenberg. Open boundaries in a cellular automaton model for traffic flow with metastable states. *Physical Review E*, 66(4):046113, 2002. (Cited on page 20.)

[13] Jorge Batista. Tracking pedestrians under occlusion using multiple cameras. In Aurélio Campilho and Mohamed Kamel, editors, *Image Analysis and Recognition*, volume 3212 of *Lecture Notes in Computer Science*, pages 552–562. Springer Berlin / Heidelberg, 2004. (Cited on page 62.)

[14] Wolfgang Becker. *Saccades*, chapter 5, pages 95–137. Eye movements. CRC Press, FL, USA, 1st edition, 1991. (Cited on page 34.)

[15] David Beymer. Person counting using stereo. In *Human Motion, 2000. Proceedings. Workshop on*, pages 127–131, Washington, DC, USA, 2000. IEEE Computer Society. (Cited on page 62.)

[16] Ofer Biham, A. Alan Middleton, and Dov Levine. Self-organization and a dynamical transition in traffic-flow models. *Physical Review A*, 46:R6124–R6127, Nov 1992. (Cited on page 20.)

[17] James Black and Tim Ellis. Mutli camera image tracking. *Image and Vision Computing*, 24(1):1256–1267, 2006. (Cited on page 63.)

[18] V. J. Blue and J. L. Adler. Cellular automata model of emergent collective bi-directional pedestrian dynamics. In Mark A. Bedau, McCaskill John S., Norman H. Packard, and Steen Rasmussen, editors, *Artificial Life VII, Proceedings of the Seventh International Conference on Artificial Life*, pages 437–445, USA, 2000. The MIT Press. (Cited on page 19.)

[19] Eric Bonabeau. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences of the United States of America*, 99(3):7280–7287, 2002. (Cited on page 20.)

[20] Antoine Bordes, Seyda Ertekin, Jason Weston, and Leon Bottou. Fast kernel classifiers with online and activate learning. *Journal of Machine Learning Research*, 6:1579–1619, 2005. (Cited on page 123.)

[21] A. Broggi, P. Cerri, S. Ghidoni, P. Grisleri, and Ho Gi Jung. Localization and analysis of critical areas in urban scenarios. In *Intelligent Vehicles Symposium, 2008 IEEE*, pages 1074 –1079, Washington, DC, USA, June 2008. IEEE Computer Society. (Cited on pages 60 and 61.)

[22] JM Burnfield and CM Powers. *Normal and pathologic gait.* Orthopaedic Physical Therapy Secrets. Hanley and Belfus Medical Publishers, 2nd edition, 2006. (Cited on pages 31 and 38.)

[23] C. Burstedde, K. Klauck, Schadschneider A., and J. Zittartz. Simulation of pedestrian dynamics using a two-dimensional cellular automaton. *Physica A: Statistical Mechanics and its Applications*, 295(3):507–525, 2001. (Cited on pages 20 and 43.)

[24] Francesca Camillen, Salvatore Capre, Cesare Garofalo, Matteo Ignaccolo, Giuseppe Inturri, Alessandro Pluchino, Andrea Rapisarda, and Salvatore Tudisco. Multi agent simulation of pedestrian behaviour in closed spatial environments. In *Science and Technology for Humanity (TIC-STH)*

IEEE Toronto International Conference, pages 375 – 380, Washington, DC, USA, 2009. IEEE Computer Society. (Cited on pages 20, 21 and 43.)

[25] John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8(6):679 –698, 1986. (Cited on page 75.)

[26] D Carwright and A Zander. *Group dynamics: Research and Theory.* American public health association, New York, 3rd edition, 1968. (Cited on page 36.)

[27] Nurhan Cetin, Kai Nagel, Bryan Raney, and Andreas Voellmy. Large-scale multi-agent transportation simulations. *Computer Physics Communications*, 147(1-2):559–564, 2002. (Cited on pages 6 and 21.)

[28] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/$\sim$cjlin/libsvm`. (Cited on pages 78, 120, 123 and 167.)

[29] Lei Chen and Ng Raymond. On the marriage of lp-norms and edit distance. In Mario A. Nascimento, M. Tamer Ozsu, Donald Kossmann, Jose A. Blakeley Renee J. Miller, and Berni Schiefer, editors, *Proceedings of the 30th Very Large Databases (VLDB) Conference*, pages 792–803, San Francisco, CA, USA, 2004. Morgan Kaufmann Publishers. (Cited on pages 102 and 125.)

[30] F.-C. Cheng and S.-J. Ruan. Accurate motion detection using a self-adaptive background matching framework. *Intelligent Transportation Systems, IEEE Transactions on*, 13(2):671–679, June 2012. (Cited on pages 57 and 58.)

[31] Fan-Chieh Cheng, Shih-Chia Huang, and Shanq-Jang Ruan. Illumination-sensitive background modeling approach for accurate moving object detection. *Broadcasting, IEEE Transactions on*, 57(4):794–801, 2011. (Cited on page 57.)

[32] Simon Chih-Feng Shu. Housing layout and crime vulnerability. *Urban Des Int*, 2000. (Cited on page 7.)

[33] Bastien Chopard. *Cellular automata modeling of physical systems.* Cambridge University Press, 1998. (Cited on page 19.)

[34] Vaclav Chvatal and David Sankoff. Longest common subsequences of two random sequences. *Journal of Applied Probability*, 12(2):306–315, 1975. (Cited on pages 102 and 125.)

[35] Panasonic Corporation. Panasonic product support, 2011. (Cited on page 77.)

[36] R. Courant and D. Hilbert. *Methods of mathematical physics*, volume 1. Interscience Publishers, New York, USA, first english edition, 1937. Translated and revised from the German original. (Cited on pages 76, 102, 104, 106 and 108.)

[37] T. Cover and P. Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21 –27, January 1967. (Cited on page 76.)

[38] Nello Cristianini and John Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods.* Cambridge University Press, Cambridge CB2 2RU, UK, tenth edition, 2006. (Cited on pages 86, 87, 89, 107 and 108.)

[39] M. Cuturi and J.-P. Vert. A mutual information kernel for sequences. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, volume 3, pages 1905–1910, Washington, DC, USA, 2004. IEEE Computer Society. (Cited on pages 101 and 104.)

[40] Marco Cuturi, Jean-Philippe Vert, Qystein Birkenes, and Tomoko Matsui. A kernel for time series based on global alignments. In *Acoustics, Speech and Signal Processing*, volume 4, pages 413–416, Washington, DC, USA, 2007. IEEE Signal Processing Society. (Cited on page 106.)

[41] C. Dai, Y. Zheng, and X Li. Pedestrian detection and tracking in infrared imagery using shape and appearance. *Computer Vision and Image Understanding*, 106(2 and 3):288–299, 2007. (Cited on pages 59, 60 and 61.)

[42] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, December 2006. (Cited on pages 133 and 136.)

[43] Elena Deza and Michel Marie Deza. *Encyclopedia of distances*. Springer-Verlag Berlin Heidelberg, New York, USA, 2009. (Cited on pages 60 and 102.)

[44] Anup Doshi and Mohan M. Trivedi. Head and gaze dynamics in visual attention and context learning. In *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pages 77 –84, Washington, DC, USA, June 2009. IEEE Computer Society. (Cited on page 33.)

[45] Ayers Douglas and Mubarak Shah. Monitoring human behavior from video taken in an office environment. *International Journal of Computer Vision*, 19(12):833–846, 2001. (Cited on page 144.)

[46] Olive Jean Dunn. Multiple comparisons among means. *Journal of the American Statistical Association*, 56(293):pp. 52–64, 1961. (Cited on page 128.)

[47] H. Elzein, S. Lakshmanan, and P. Watta. A motion and shape-based pedestrian detection algorithm. In *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, pages 500 – 504, Washington, DC, USA, June 2003. IEEE Computer Society. (Cited on page 60.)

[48] M. Enzweiler and D.M. Gavrila. A multilevel mixture-of-experts framework for pedestrian classification. *Image Processing, IEEE Transactions on*, 20(10):2967–2979, Oct. 2011. (Cited on page 60.)

[49] A. Ess, B. Leibe, K. Schindler, and L. Van Gool. A mobile vision system for robust multi-person tracking. In *Computer Vision and Pattern*

Recognition, 2008. CVPR 2008. IEEE Conference on, pages 1–8, Washington, DC, USA, June 2008. IEEE Computer Society. (Cited on pages 57 and 61.)

[50] Tom Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27:861–874, 2006. (Cited on page 125.)

[51] Carlos Ferrer, Diana Torres, and María E. Hernández-Díaz. Using dynamic time warping of t0 contours in the evaluation of cycle-to-cycle pitch detection algorithms. *Pattern Recognition Letters*, 31(6):517–522, 2011. (Cited on page 104.)

[52] A. Fisher. *Statistical methods and scientific inference.* Hafner Publishing Co., New York, USA, 2nd edition, 1959. (Cited on page 133.)

[53] A. Frank and A. Asuncion. The uci machine learning repository, 2011. (Cited on pages 119 and 136.)

[54] Milton Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200):675–701, 1937. (Cited on page 128.)

[55] Kay Fuerstenberg and Volker Willhoeft. Object tracking and classification using laserscanners-pedestrian recognition in urban environment. In *IEEE Intelligent Transportation Systems Conference*, pages 451–453, Oakland(CA), USA, 2001. IEEE. (Cited on page 61.)

[56] K.Ch. Fuerstenberg, K.C.J. Dietmayer, and V. Willhoeft. Pedestrian recognition in urban traffic using a vehicle based multilayer laserscanner. In *Intelligent Vehicle Symposium, 2002. IEEE*, volume 1, pages 31–35, Washington, DC, USA, June 2002. IEEE Computer Society. (Cited on page 61.)

[57] T. Gandhi and M.M. Trivedi. Pedestrian collision avoidance systems: a survey of computer vision based recent studies. In *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE*, pages 976 –981,

Washington, DC, USA, Sept. 2006. IEEE Computer Society. (Cited on page 75.)

[58] Yin Gao and Ning Gu. Complexity, human agents, and architectural design: a computational framework. *Design Principles and Practices: An International Journal*, 3(6):115–125, 2009. (Cited on page 144.)

[59] D. M. Gavrila and S. Munder. Multi-cue pedestrian detection and tracking from the moving vehicle. *International Journal of Computer Vision*, 73(1):41–59, 2007. (Cited on pages 6, 57, 58, 70 and 76.)

[60] Dariu M. Gavrila, Uwe Franke, Christian Wohler, and Steffen Gorzig. Real-time vision for intelligent vehicles. *IEEE Instrumentation and Measurement Magazine*, 4(1):22–27, 2001. (Cited on page 61.)

[61] D. Geronimo, A.M. Lopez, A.D. Sappa, and T. Graf. Survey of pedestrian detection for advanced driver assistance systems. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(7):1239 –1258, 2010. (Cited on pages 58 and 75.)

[62] P.G. Gipps and B. Marksjo. A micro simulation model for pedestrian flows. *Mathematics and Computers in Simulation*, 27(2-3):95–105, 1985. (Cited on page 20.)

[63] Chiristian Gloor, Duncan Cavens, Eckart Lange, Kai Nagel, and Willy Schmid. A pedestrian simulation for very large scale applications. *Andreas KOCH and Peter MANDL(Eds.) Multi-Agenten-Systeme in der Geographie*, 23:1–16, 2003. (Cited on pages 19, 21 and 43.)

[64] Reginald G. Golledge and Robert John Stimson. *Spatial behavior, a geographic perspective*. The Guilford Press, New York, NY, USA, 1997. (Cited on page 6.)

[65] Rafael C. González and Richard Eugene Woods. *Digital image processing*. Pearson/Prentice Hall, New Jersey, USA, 3rd edition, 2008. (Cited on page 69.)

[66] T. Graepel, R. Herbrich, P. Bollmann-Sdorra, and K. Obermayer. Classification on pairwise proximity data. In Sara A. Solla, Todd K. Leen, and Klaus-Robert Müller, editors, *Advances in Neural Information Processing Systems*, volume 11, pages 438–444. The MIT Press, 1999. (Cited on page 128.)

[67] Thore Graepel, Ralf Herbrich, Peter Bollmann-Sdorra, and Klaus Obermayer. Classification on pairwise proximity data. In Michael S. Kearns, Sara A. Solla, and David A. Cohn, editors, *Proceedings of the 1998 Conference on Advances in Neural Information Processing Systems 11*, pages 438–444, Cambridge, MA, USA, 1999. MIT Press. (Cited on page 108.)

[68] Arnulf B. A. Graf, Alexander J. Smola, and Silvio Borer. Classification in a normalized feature space using support vector machines. *IEEE Transactions on Neural Networks*, 14(3):597–605, 2003. (Cited on page 123.)

[69] S. Gudmundsson, T.P. Runarsson, and S. Sigurdsson. Support vector machines and dynamic time warping for time series. In *Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 2772 –2776, Washington, DC, USA, June 2008. IEEE Computer Society. (Cited on pages 116 and 128.)

[70] S. Hagishima, K. Mitsuyoshi, and S. Kurose. Estimation of pedestrian shopping trips in neighborhood by using a spatial interaction model. *Environment and Planning A*, 19(9):1139–1152, 1987. (Cited on page 19.)

[71] Y. Hao, S. Chunfu, Z. Yi, and C. Xiaoming. Study on moving pedestrian tracking based on video sequences. *Journal of Transportation Systems Engineering and Information Technology*, 7(4):74–51, 2007. (Cited on pages 57, 58 and 61.)

[72] Ismail Haritaoglu, David Harwood, and Larry S. Davis. W4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830, 2000. (Cited on page 144.)

[73] Bastian Hartmann and Norbert Link. Gesture recognition with inertial sensors and optimized dtw prototypes. In *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, pages 2102–2109, Washington, DC, USA, 2010. IEEE Computer Society. (Cited on page 104.)

[74] M. Harville. Stereo person tracking with adaptive plan-view statistical templates. In *Workshop on Statistical Methods in Video Processing*, pages 67–72, 2002. (Cited on page 62.)

[75] Michael Harville. Stereo person tracking with adaptive plan-view templates of height and occupancy statistics. *International Journal of Computer Vision*, 22(1):127–142, 2004. (Cited on page 62.)

[76] M. Heikkila and M. Pietikainen. A texture-based method for modeling the background and detecting moving objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(4):657 –662, 2006. (Cited on pages 68 and 69.)

[77] Hermann Helmholtz. *Treatise on physiological optics 3rd edn.* Dover Publications, Mineola, N.Y., USA, 1962. (Cited on page 34.)

[78] C. Hilario, J.M. Collado, J.M. Armingol, and A. De la Escalera. Pedestrian detection for intelligent vehicles based on active contour models and stereo vision. In Gran Canaria, editor, *Proc. Int. Wksp. on Computer Aided Systems Theory*, Spain, 2005. (Cited on page 57.)

[79] Sepp Hochreiter and Klaus Obermayer. Support vector machines for dyadic data. *Neural Computation*, 18(6):1472–1510, June 2006. (Cited on pages 72, 81, 95, 96, 98, 106, 108, 117, 118, 120, 151 and 167.)

[80] James E. Hoffman. Visual attention and eye movements. In H Pashler, editor, *Attention*, page 119. Psychology Press, San Diego, 1998. (Cited on page 34.)

[81] Michael A Hogg. Interpersonal attraction, social identification and psychological group formation. *European Journal of Social Psychology*, 15(1):51–56, 1985. (Cited on page 36.)

[82] Weiming Hu, Tieniu Tan, Liang Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 34(3):334 –352, Aug. 2004. (Cited on page 6.)

[83] Chunsheng Hua, Haiyuan Wu, Qian Chen, and Toshikazu Wada. A general framework for tracking people. In *Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition*, FGR '06, pages 511–516, Washington, DC, USA, 2006. IEEE Computer Society. (Cited on pages 57 and 61.)

[84] Yun Wu Huang and Philip S. Yu. Adaptive query processing for time-series data. In Surajit Chaudhuri, David Madigan, and Usama Fayya, editors, *Proceedings of the 5th Int'l Conference on Knowledge Discovery and Data Mining*, pages 282–286, New York, NY, USA, 1999. ACM. (Cited on page 102.)

[85] Piotr Indyk, Nick Koudas, and S. Muthukrishnan. Identifying representative trends in massive time series data sets using sketches. In Amr El Abbadi, Michael L. Brodie, Sharma Chakravarthy, Umeshwar Dayal, Nabil Kamel, Gunter Schlageter, and Kyu-Young Whang, editors, *Proceedings of the 26th International Conference on Very Large Data Bases*, VLDB'00, pages 363–372, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. (Cited on page 102.)

[86] L. Iocchi and R.C. Bolles. Integrating plan-view tracking and color-based person models for multiple people tracking. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 3, pages 872–875, Washington, DC, USA, Sept. 2005. IEEE Computer Society. (Cited on page 62.)

[87] L Itti and C Koch. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research*, 40(1):1489–1506, 2000. (Cited on page 33.)

[88] Arash Jalalian and Stephan K. Chalup. Gdtw-p-svms: variable-length time series analysis using support vector machines. *Neurocomputing*, 2012. (Cited on pages 100, 145, 150, 167 and 176.)

[89] Arash Jalalian, Stephan K. Chalup, and J. Michael Ostwald. Learning attention-driven activities using pedestrian trajectory analysis in video data,. 2012. internal report, Interdisciplinary Machine Learning Research Group, School of Electrical Engineering and Computer Science, The University of Newcastle, (under review). (Cited on pages 142 and 177.)

[90] Arash Jalalian, Stephan K. Chalup, and Michael J. Ostwald. Intelligent evaluation of urban streetscape designs by analysing pedestrian body dynamics. In *The Third International Workshop on Advanced Computational Intelligence (IWACI2010)*, pages 442–447, Washington, DC, USA, Aug 25-27 2010. IEEE Computer Society. (Cited on pages 33, 101, 142, 143 and 176.)

[91] Arash Jalalian, Stephan K Chalup, and Michael J Ostwald. Simulating pedestrian flow dynamics for evaluating the design of urban and architectural space. In Chris Murphy, Susan J. Wake, David Turner, Graeme McConchie, and David Rhodes, editors, *44th Annual Conference of the Architectural Science Association, ANZAScA 2010*, Auckland, New Zealand, 2010. Unitec Institute of Technology. (Cited on pages 17, 33, 101, 143 and 176.)

[92] Arash Jalalian, Stephan K. Chalup, and Michael J. Ostwald. Agent-agent interaction as a component of agent-environment interaction in the modelling and analysis of pedestrian visual behaviour. In C. M. Herr, N. Gu, S. Roudavski, and M. A. Schnabel, editors, *The 16th International Conference of the Association for Computer-Aided Architectural Design Research in Asia*, pages 555–564, Newcastle, Australia, 2011. (Cited on

pages 8, 17, 33, 101, 142, 143, 145, 146, 147, 148, 149, 152, 161, 164 and 176.)

[93] Arash Jalalian, Stephan K Chalup, and Michael J Ostwald. Architectural evaluation of simulated pedestrian spatial behaviour. *Architectural Science Review*, 54(2):132–140, 2011. (Cited on pages 6, 17, 33, 34, 51, 101, 142, 143, 146, 149, 152 and 176.)

[94] Arash Jalalian, Stephan K. Chalup, and Michael J. Ostwald. Analysis of pedestrian spatial behaviour using gdtw-p-svm. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1574–1581, Washington, DC, USA, 2012. IEEE Computer Society. (Cited on pages 4, 142, 176 and 177.)

[95] Arash Jalalian and Mahmmod Fathy. Pedestrian detection from a moving camera with an advanced camera-motion estimator. In Kokou Yetongnon, Richard Chbeir, and Albert Dipanda, editors, *3rd International IEEE Conference on Signal-Image Technologies and Internet-Based System*, pages 965–971, Washington, DC, USA, Dec 16-18 2007. IEEE Computer Society. (Cited on pages 6, 57, 58, 61, 75, 77 and 101.)

[96] Young-Seon Jeong, Myong K. Jeong, and Olufemi A. Omitaomud. Weighted dynamic time warping for time series classification. *Pattern Recognition*, 44(9):2231–2240, 2011. (Cited on pages 102, 104 and 125.)

[97] B. Jiang. Simped: simulating pedestrian flows in a virtual urban environment. *Journal of Geographic Information and Decision Analysis*, 3(1):21–30, 1999. (Cited on pages 6 and 21.)

[98] Konstantinos Kalpakis, Dhiral Gada, and Puttagunta Vasundhara. Distance measures for effective clustering of arima time-series. In Nick Cercone, Tsau Young Lin, and Xindong Wu, editors, *Proceedings of the 2001 IEEE International Conference on Data Mining*, pages 273–280, Washington, DC, USA, 2001. IEEE Computer Society. (Cited on page 102.)

[99] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: theory and experiment. *In IEEE Transactions on Pat-*

tern Analysis and Machine Intelligence, 6(1):920–932, 1994. (Cited on page 61.)

[100] T. Kanda, D. Glas, M. Shiomi, H. Ishigurio, and N. Hagita. Who will be the customer?: A social robot that anticipates people behavior from their trajectories. In *10th International Conference on Ubiquitous Computing*, pages 380–389, New York, NY, USA, 2008. ACM. (Cited on page 61.)

[101] T. Kanda, M. Shiomi, L. Perrin, T. Nomura, H. Ishiguro, and N. Hagita. Analysis of people trajectories with ubiquitous sensors in a science museum. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 4846 –4853, Washington, DC, USA, April 2007. IEEE Computer Society. (Cited on page 61.)

[102] Mark A. Keck, Jr., James W. Davis, and Ambrish Tyagi. Tracking mean shift clustered point clouds for 3d surveillance. In *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*, VSSN '06, pages 187–194, New York, NY, USA, 2006. ACM. (Cited on page 62.)

[103] C.G. Keller, M. Enzweiler, M. Rohrbach, D.F. Llorca, C. Schnorr, and D.M. Gavrila. The benefits of dense stereo for pedestrian detection. *Intelligent Transportation Systems, IEEE Transactions on*, 12(4):1096–1106, 2011. (Cited on pages 57, 60 and 61.)

[104] Philip Kelly, Noel E. O'Connor, and Alan F. Smeaton. Pedestrian detection in uncontrolled environments using stereo and biometric information. In *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*, VSSN '06, pages 161–170, New York, NY, USA, 2006. ACM. (Cited on page 57.)

[105] Philip Kelly, Noel E. O'Connor, and Alan F. Smeaton. Robust pedestrian detection and tracking in crowded scenes. *Image Vision Comput.*, 27:1445–1458, September 2009. (Cited on page 57.)

[106] A. Kendon. Some function of gaze direction in social interaction. *Acta Psychol*, 26(1):22–63, 1967. (Cited on page 33.)

[107] E. Keogh, Q. Zhu, B. Hu, Y. Hao, X. Xi, L. Wei, and C. A. Ratanamahatana. The ucr time series classification/clustering data sets homepage, 2011. (Cited on pages 119, 124 and 126.)

[108] Eamonn Keogh and Shruti Kasetty. On the need for time series data mining benchmarks: a survey and empirical demonstration. In *The 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD'02, pages 102–111, New York, NY, USA, 2002. ACM. (Cited on page 102.)

[109] Eamonn J. Keogh and Michael J. Pazzani. Scaling up dynamic time warping for datamining applications. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD'00, pages 285–289, New York, NY, USA, 2000. ACM. (Cited on page 104.)

[110] Eamonn J Keogh and Michael J Pazzani. Derivative dynamic time warping. In Robert Grossman, Vipin Kumar, and Jiawei Han, editors, *First SIAM International Conference on Data Mining*, pages 1–11, Philadelphia, PA 19104-2688 USA, 2001. SIAM. (Cited on pages 102 and 104.)

[111] Ansgar Kirchner and Andreas Schadschneider. Simulation of evacuation processes using a bionics-inspired cellular automaton model for pedestrian dynamics. *Physica A: Statistical Mechanics and its Applications*, 312(1):260–276, 2002. (Cited on page 20.)

[112] Raymond M Klein and Amanda Pontefract. Does oculomotor readiness mediate cognitive control of visual attention? revisited! In Carlo Umilta, editor, *Attention and Performance XV*. MIT Press, 1994. (Cited on page 34.)

[113] H. W. Kuhn and A. W. Tucker. Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492, Berkeley, CA, USA, 1951. University of California Press. (Cited on page 107.)

[114] L. Lam, S.-W. Lee, and C.Y. Suen. Thinning methodologies-a comprehensive survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(9):869 –885, 1992. (Cited on page 75.)

[115] Pat Langley. Machine learning as an experimental science. *Machine Learning*, 3(1):5–8, 1988. (Cited on page 119.)

[116] H. Lau. Behavioural approach for multi-robot exploration. In *Australasian Conference on Robotics and Automation*, pages 1–7, 2003. (Cited on page 22.)

[117] Maria Cristina Dias Lay, Antonio Reis, Virginia Dreux, and Debora Becker. Urban security and spatial behavior: syntactic and perceptual analysis of the central area of porto alegre. In *6th International Space Syntax Symposium*, pages 128–01–128–10, 2007. (Cited on page 7.)

[118] J. Leal, S. Scheding, and G. Dissanayake. Propabilistic 2d mapping of unconstructed environments. In *Australasian Conference on Robotics and Automation*, pages 19–24, 2006. (Cited on page 22.)

[119] Hansheng Lei and Bingyu Sun. A study on the dynamic time warping in kernel machines. In Kokou Yetongnon, Richard Chbeir, and Albert Dipanda, editors, *Signal-image technologies and internet-based systems*, pages 839–845, Washington, DC, USA, 2007. IEEE Computer Society. (Cited on pages 106 and 113.)

[120] Zhizheng Liang and YouFu Li. Incremental support vector machine learning in the primal and applications. *Neurocomputing*, 72(10):2249 – 2258, 2009. (Cited on page 104.)

[121] Zhe Liao and Jun Wang. Forecasting model of global stock index by stochastic time effective neural network. *Expert Systems with Applications*, 37(1):834–841, 2010. (Cited on page 101.)

[122] Ji Hyoun Lim, O. Tsimhoni, and Yili Liu. Investigation of driver performance with night vision and pedestrian detection systems. *Intelligent*

Transportation Systems, IEEE Transactions on, 11(3):670 –677, 2010. (Cited on pages 59 and 61.)

[123] Ruijiang Lou and Yan Guo. Real-time stereo tracking of multiple moving heads. In *Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 2001. Proceedings. IEEE ICCV Workshop on*, pages 55 –60, Washington, DC, USA, 2001. IEEE Computer Society. (Cited on page 62.)

[124] D. R. Lowne, S. J. Roberts, and R. Garnett. Sequential non-stationary dynamic classification with sparse feedback. *Pattern Recognition*, 43(3):897–905, 2010. (Cited on page 101.)

[125] D. Lukas. *Least squares support vector machines classification applied to brain tumour recognition using magnetic resonance spectroscopy.* PhD thesis, Departement Elektrotechniek, Kasteelpark Arenberg 10, 3001 Leuven (Heverlee), 2003. (Cited on page 89.)

[126] Ronny Luss and Alexandre d'Aspremont. Support vector machine classification with indefinite kernels. *Computing Research Repository, CoRR*, abs/0804.0188:1–8, 2009. (Cited on pages 104 and 139.)

[127] Laurens van der Maaten. Learning discriminative fisher kernels. In *Proceedings of 28th International Conference on Machine Learning*, Bellevue, WA, USA, 2011. Only online version was available at the time of preparing this paper: www.icml-2011.org/papers/178_icmlpaper.pdf. (Cited on pages 137 and 138.)

[128] S. Maniccam. Effects of back step and update rule on congestion of mobile objects. *Physica A: Statistical Mechanics and its Applications*, 346(3):631–650, 2005. (Cited on page 20.)

[129] Marvin Marcus and Henryk Minc. *Introduction to linear algebra.* Dover Publications, New York, USA, 1988. (Cited on page 108.)

[130] Pierre-Francois Marteau. Time warp edit distance with stiffness adjustment for time series matching. *IEEE Transactions on Pattern Analysis*

and Machine Intelligence, 31(2):306–318, 2009. (Cited on pages 102, 109, 129, 132 and 150.)

[131] S. Matteoli, N. Acito, M. Diani, and G. Corsini. An automatic approach to adaptive local background estimation and suppression in hyperspectral target detection. *Geoscience and Remote Sensing, IEEE Transactions on*, 49(2):790–800, 2011. (Cited on pages 56 and 57.)

[132] J. Mercer. Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 209:415–446, 1909. (Cited on pages 89, 106, 108 and 113.)

[133] A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 23(4):349, 2001. (Cited on pages 56 and 60.)

[134] Masakuni Muramatsu, Tunemasa Irie, and Takashi Nagatani. Jamming transition in pedestrian counter flow. *Physica A: Statistical Mechanics and its Applications*, 267(3):487–498, 1999. (Cited on page 20.)

[135] Jacinto Nascimento, Mario A. T. Figueiredo, and Jorge S. Marques. Unsupervised learning of motion patterns using generative models. In *IEEE International Conference on Image Processing*, ICIP'08, pages 761–764, Washington, DC, USA, 2008. IEEE Computer Society. (Cited on page 144.)

[136] J.C. Nascimento, M. Figueiredo, and J.S. Marques. Trajectory classification using switched dynamical hidden markov models. *Image Processing, IEEE Transactions on*, 19(5):1338–1348, 2010. (Cited on page 144.)

[137] V Navalpakkam and L Itti. Modeling the influence of task on attention. *Vision Research*, 45(1):205–231, 2005. (Cited on page 33.)

[138] M. P. Ness, J. F. Morral, and B. G. Hutchinson. An analysis of central business district pedestrian circulation patterns. *Highway research record*, 283:11–18, 1969. (Cited on page 19.)

[139] Minh Hoai Nguyen. *Segment-based SVMs for time series analysis*. PhD thesis, The Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, 2012. (Cited on page 140.)

[140] Minh Hoai Nguyen and Fernando De la Torre. Maximum margin temporal clustering. *Journal of Machine Learning Research - Proceedings Track*, 22:520–528, 2012. (Cited on page 140.)

[141] Michael Oakes. Rescuing robinson crusoe. *American Journal of Epidemiology*, 168(1):9–12, 2008. (Cited on page 143.)

[142] S. J. Older. Movement of pedestrians on footways in shopping streets. *Traffic engineering and control*, 10(4):160–163, 1968. (Cited on page 19.)

[143] Michael J. O'Neill. Effects of signage and floor plan configuration on wayfinding accuracy. *Environment and Behavior*, 23(5):553–574, 1991. (Cited on page 6.)

[144] Cheng Soon Ong, Xavier Mary, Stéphane Canu, and Alexander J. Smola. Learning with non-positive kernels. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, pages 81–89, New York, NY, USA, 2004. ACM. (Cited on pages 104 and 139.)

[145] C. Orsenigo and C. Vercellis. Combining discrete svm and fixed cardinality warping distances for multivariate time series classification. *Pattern Recognition*, 43(11):3787–3794, 2010. (Cited on page 102.)

[146] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979. (Cited on pages 32, 66, 68 and 71.)

[147] Chia-Jung Pai, Hsiao-Rong Tyan, Yu-Ming Liang, Hong-Yuan Mark Liao, and Sei-Wang Chen. Pedestrian detection and tracking at crossroads. *Pattern Recognition*, 37(5):1025 – 1034, 2004. (Cited on page 57.)

[148] C. Papageorgiou and T. Poggio. A trainable system for object detection. *International Journal of Computer Vision.*, 38(1):15–33, 2000. (Cited on page 60.)

[149] Sanghyun Park, Sang-wook Kim, and Wesley W. Chu. Segment-based approach for subsequence searches in sequence database. In *Proceedings of the 2001 ACM Symposium on Applied Computing*, SAC'01, pages 248–252, New York, NY, USA, 2001. ACM. (Cited on page 102.)

[150] D. C. Parker, S. M. Manson, M. A. Janssen, M. J. Hoffmann, and P. Deadman. Multi-agent systems for simulation of land-use and land-cover change: a review. *Annals of the Association of American Geographers*, 93(2):316–340, 2003. (Cited on pages 6 and 21.)

[151] N. G. Pavlidis, D. K. Tasoulis, N. M. Adams, and D. J. Hand. Lambda-perceptron: An adaptive classifier for data streams. *Pattern Recognition*, 44(1):78–96, 2011. (Cited on page 101.)

[152] Alex Pentland and Andrew Liu. Modeling and prediction of human behvaior. *Neural Computation*, 11(1):229–242, 1999. (Cited on page 33.)

[153] Chotirat Ann Ratanamahatana and Eamonn Keogh. Making time-series classification more accurate using learned constraints. In Michael W. Berry, Umeshwar Dayal, Chandrika Kamath, and David Skillicorn, editors, *Proceedings of the Fourth SIAM International Conference on Data Mining*, pages 11–22, Philadelphia, PA 19104-2688 USA, 2004. SIAM. (Cited on page 125.)

[154] P. Remagnino and G.L. Foresti. Ambient intelligence: A new multidisciplinary paradigm. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 35(1):1 – 6, Jan. 2005. (Cited on page 6.)

[155] Ying Ren, Chin-Seng Chua, and Yeong-Khing Ho. Motion detection with non-stationary background. In *Image Analysis and Processing, 2001. Proceedings. 11th International Conference on*, pages 78 –83, Washington, DC, USA, 2001. IEEE Computer Society. (Cited on page 56.)

[156] Frigyes Riesz and Bėla Szökefalvi Nagy. *Functional analysis.* Dover Publications, INC., Mineola, N.Y., USA, 1990. (Cited on page 108.)

[157] Mark Rosenblum, Yaser Yacoob, and Larry S. Davis. Human expression recognition from motion using a radial basis function network architecture. *IEEE Transactions on Pattern Anamysis and Machine Intelligence*, 7(5):1121–1138, 1996. (Cited on page 144.)

[158] C. A. Rothkopf, D. H. Ballard, and M. M. Hayhoe. Task and context determine where you look. *Journal of Vision*, 7(14):1–20, 2007. (Cited on page 33.)

[159] Dick Saarloos, Jae-Eun Kim, and Harry Timmermans. The built environment and health: introducing individual space-time behavior. *International Journal of Environmental Research and Public Health*, 6(6):1724–1743, 2009. (Cited on page 144.)

[160] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 26(1):43–49, 1978. (Cited on pages 109 and 150.)

[161] Stan Salvador and Philip Chan. Toward accurate dynamic time warping in linear time and space. *Intell. Data Anal.*, 11(5):561–580, 2007. (Cited on page 121.)

[162] Steven Salzberg. On comparing classifiers: pitfalls to avoid and a recommended approach. *Data Mining and Knowledge Discovery*, 1(3):317–327, 1997. (Cited on page 119.)

[163] N. Sapankevych and R. Sankar. Time series prediction using support vector machines: A survey. *Computational Intelligence Magazine, IEEE*, 4(2):24 –38, 2009. (Cited on page 86.)

[164] I. J. Schoenberg. Metric spaces and positive definite functions. *Transactions of the American Mathematical Society*, 44(3):522–536, 1938. (Cited on page 108.)

[165] B. Scholkopf, Kah-Kay Sung, C.J.C. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik. Comparing support vector machines with gaussian kernels to radial basis function classifiers. *Signal Processing, IEEE Transactions on*, 45(11):2758 –2765, 1997. (Cited on page 94.)

[166] Bernhard Scholkopf, Christopher J. C. Burges, and Alexander J. Smola. *Advances in kernel methods*. Support vector learning. MIT Press, 1998. (Cited on page 95.)

[167] Bernhard Schölkopf, John C. Platt, John C. Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, July 2001. (Cited on page 104.)

[168] Bernhard Schölkopf and Alexander J. Smola. *Learning with kernels, Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, Cambridge, MA, USA, 2001. (Cited on pages 106 and 116.)

[169] Matthias Seeger. Covariance kernels from bayesian generative models. In S. Thrun S. Becker and K. Obermayer, editors, *Neural Information Processing Systems*, volume 14, pages 905–912, Cambridge MA. USA, 2000. The MIT Press. (Cited on page 104.)

[170] A. Shashua, Y. Gdalyahu, and G. Hayun. Pedestrian detection for driving assistance systems: single-frame classification and system level performance. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 1 – 6, Washington, DC, USA, June 2004. IEEE Computer Society. (Cited on page 60.)

[171] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA, 2004. (Cited on pages 136 and 137.)

[172] John Shawe-Taylor and Shiliang Sun. A review of optimization methodologies in support vector machines. *Neurocomputing*, 74(17):3609 – 3618, 2011. (Cited on page 104.)

[173] Ameya Shendarkar, Karthik Vasudevan, Seungho Lee, and Young-Jun Son. Crowd simulation for emergency response using bdi agent based on virtual reality. In *Simulation Conference, 2006. WSC 06. Proceedings of the Winter*, pages 545 –553, Dec. 2006. (Cited on page 7.)

[174] H. Shimizu and T. Poggio. Direction estimation of pedestrian from multiple still images. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 596 – 600, Washington, DC, USA, 2004. IEEE Computer Society. (Cited on page 60.)

[175] Hiroshi Shimodaira, Ken-ichi Noma, Mitsuru Nakai, and Sagayama Shigeki. Dynamic time-alignment kernel in support vector machine. In Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 921–928, Cambridge, MA, USA, 2002. MIT Press. (Cited on pages 72, 104 and 109.)

[176] N. Suzuki, K. Hirasawa, K. Tanaka, and Y. Kobayashi. Learning motion patterns and anomaly detection by human trajectory analysis. In *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, pages 498–503, Washington, DC, USA, 2007. IEEE Computer Society. (Cited on pages 7, 21 and 144.)

[177] Wang Tao and Chen Jun. An improved cellular automaton model for urban walkway bi-directional pedestrian flow. In *Measuring Technology and Mechatronics Automation, 2009. ICMTMA '09. International Conference on*, volume 3, pages 458–461, Washington, DC, USA, 2009. IEEE Computer Society. (Cited on pages 21 and 43.)

[178] Luke Thiele. Virtual people and architecture representation. In Mark Luther, Tony Dawson, Jeremy Ham, Monica Moore, John Rollo, and Graham Treloar, editors, *36th annual conference of the Australian and New Zealand Architectural Science Association (ANZAScA)*, volume 2, pages 521–524, Australia, 2002. Deakin University, School of Architecture and Building. (Cited on pages 5, 6, 21 and 33.)

[179] A. Turner and A. Penn. Encoding natural movement as an agent-based system: an investigation into human pedestrian behavior in the built environment. *Environment and Planning B: Planning and Design*, 29(1):473–490, 2002. (Cited on page 61.)

[180] K. Uchida, J. Miura, and Y. Shirai. Tracking multiple pedestrians in crowd. *Transactions of the Institute of Systems, Control and Information Engineers*, 14(4):180–185, 2000. (Cited on pages 56, 58 and 60.)

[181] Vladimir Naumovich Vapnik. *Statistical learning theory*. Adaptive and Learning Systems for Signal Processing, Communications, and Control. Wiley, New York, NY, USA, 1998. (Cited on pages 60, 82, 86, 91, 92, 94, 104 and 162.)

[182] V. M. Velichko and N. G. Zagoruyko. Automatic recognition of 200 words. *International Journal of Man-Machine Studies*, 2(3):223–234, 1970. (Cited on pages 102, 103, 105, 109, 125, 147 and 150.)

[183] T Vincent, L Risser, and P Ciuciu. Spatially adaptive mixture modeling for analysis of fmri time series. *IEEE Transactions on Medical Imaging*, 29(4):1059–1074, 2010. (Cited on page 101.)

[184] Baohua Wang, Hejiao Huang, and Xiaolong Wang. A novel text mining approach to financial time series forecasting. *Neurocomputing*, 83(1):136 – 145, 2012. (Cited on page 101.)

[185] Changzhou Wang and X. Sean Wang. Supporting content-based searches on time series via approximation. In O. Gunther and H. J. Lenz, editors, *The 12th International Conference on Scientific and Statistical Database Management*, pages 69–81, Washington, DC, USA, 2000. IEEE Computer Society. (Cited on page 102.)

[186] Guosheng Wang. A survey on training algorithms for support vector machine classifiers. In *Networked Computing and Advanced Information Management, 2008. NCM '08. Fourth International Conference on*, volume 1, pages 123 –128, 2008. (Cited on page 86.)

[187] Fang Weifeng, Yang Lizhong, and Fan Weicheng. Simulation of bi-direction pedestrian movement using a cellular automata model. *Physica A: Statistical Mechanics and its Applications*, 321(3):633–640, 2003. (Cited on page 20.)

[188] Jason Weston, Bernhard Schölkopf, Eleazar Eskin, Christina Leslie, and William Noble. Dealing with large diagonals in kernel matrices. *Annals of the Institute of Statistical Mathematics*, 55:391–408, 2003. (Cited on page 116.)

[189] William H. Whyte. *The social life of small urban spaces*. Project of Public Spaces, New York, NY, USA, 1980. (Cited on pages 6, 143 and 144.)

[190] U. Wilensky. Netlogo, 1999. Center for Connected Learning and Computer-Based Modeling, Northwestern University. (Cited on page 20.)

[191] Christian Wohler, Joachim K. Anlauf, Till Portner, and Uwe Franke. A time delay deural network aalgorithm for real-time pedestrian recognition. In *International Conference on Intelligent Vehicle*, pages 247–251, 1998. (Cited on page 6.)

[192] Martin Wöllmer, Marc Al-Hames, Florian Eyben, Björn Schuller, and Gerhard Rigoll. A multidimensional dynamic time warping algorithm for efficient multimodal fusion of asynchronous data streams. *Neurocomputing*, 73(1):366–380, 2009. (Cited on page 104.)

[193] Aaron S. W. Wong, Stephan K. Chalup, Shashank Bhatia, Arash Jalalian, Jason Kulk, and Michael J. Ostwald. Humanoid robots for modelling and analysing visual gaze dynamics of pedestrians moving in urban space. In Chris Murphy, Susan J. Wake, David Turner, Graeme McConchie, and David Rhodes, editors, *45th Annual Conference of the Architectural Science Association*, ANZASCA2011, Sydney, Australia, 2011. (Cited on page 144.)

[194] Huanmei Wu, Betty Salzberg, and Donghui Zhang. Online event-driven subsequence matching over financial data streams. In Gerhard Weikum,

Arnd Christian Konig, and Stefan Debloch, editors, *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, SIGMOD'04, pages 23–34, New York, NY, USA, June 13-18 2004. ACM. (Cited on page 101.)

[195] Tao Xiang and Shaogang Gong. Beyond tracking: modelling activity and understanding behaviour. *International Journal of Computer Vision*, 1(67):21–51, 2006. (Cited on page 144.)

[196] Zhaojun Xue, Dong Ming, Wei Song, Baikun Wan, and Shijiu Jin. Infrared gait recognition based on wavelet transform and support vector machine. *Pattern Recognition*, 43(8):2904–2910, 2010. (Cited on page 101.)

[197] Kazuhiro Yamamoto, Satoshi Kokubo, and Katsuhiro Nishinari. Simulation for pedestrian dynamics by real-coded cellular automata(rca). *Physica A: Statistical Mechanics and its Applications*, 379(2):654–660, 2007. (Cited on page 21.)

[198] Wei Yan and Yehuda E Kalay. Geometric cognitive and behavioral modeling of environmental users. In *Design Computing and Cognition*, pages 61–79. Springer Netherlands, 2006. (Cited on page 20.)

[199] George Yannis and John Golias. A critical assessment of pedestrian behaviour models. *Transportation Research Part F: Traffic Psychology and Behaviour*, 12(3):242–255, 2009. (Cited on pages 19 and 20.)

[200] S Yantis. Control of visual attention. In H Pashler, editor, *Attention*. Psychology Press, Baltimore, 1998. (Cited on pages 33, 34 and 35.)

[201] A L Yarbus. *Eye movements and vision*. Plenum Press, New York, 1967. (Cited on page 34.)

[202] Yasser Yasami, Siavash Khosravi, Saadat Pour Mozaffari, and Arash Jalalian. An unsupervised network anomaly detection approach by k-means clustering and id3 algorithms. In *Symposium on Computers and*

Communications, pages 398–403, Washington, DC, USA, 2008. IEEE Computer Society. (Cited on page 101.)

[203] Hao Yue, Hongzhi Guan, Juan Zhang, and Chunfu Shao. Study on bi-direction pedestrian flow using cellular automata simulation. *Physica A: Statistical Mechanics and its Applications*, 389(3):527–539, 2010. (Cited on page 19.)

[204] Long Zeng, Lip Man-Lip Lai, Di Qi, Yuen-Hoo Lai, and Matthew Ming-Fai Yuen. Efficient slicing procedure based on adaptive layer depth normal image. *Computer-Aided Design*, 43(12):1577–1586, 2011. (Cited on page 74.)

[205] Tianzhu Zhang, Hanqing Lu, and S.Z. Li. Learning semantic scene models by object classification and trajectory clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR'09, pages 1940–1947, Washington, DC, USA, 2009. IEEE Computer Society. (Cited on page 144.)

[206] L. Zhao and C. Thorpe. Stereo- and neural network-based pedestrian detection. *IEEE Transactions on Intelligent Transportation Systems*, 1(3), 2000. (Cited on page 57.)

[207] Yu Zheng, Like Liu, Longhao Wang, and Xing Xie. Learning transportation mode from raw gps data for geographic applications on the web. In *Proceedings of the 17th International Conference on World Wide Web*, WWW'08, pages 247–256, New York, NY, USA, 2008. ACM. (Cited on pages 138, 144 and 145.)

[208] Yu Zheng, Xing Xie, and Wei-Ying Ma. Geolife: A collaborative social networking service among user, location and trajectory. *IEEE Database Engineering Bulletin*, 33(2):32–39, 2010. (Cited on pages 136, 138, 139, 144, 145 and 147.)

[209] Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. Mining interesting locations and travel sequences from gps trajectories. In *Proceedings of*

18th International Conference on World Wide Web, WWW'09, pages 791–800, New York, NY, USA, 2009. ACM. (Cited on page 119.)