Learning Object Behaviour Models

by

Neil Johnson

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

The University of Leeds
School of Computer Studies
September 1998

The candidate confirms that the work submitted is his own and that appropriate credit has been given where reference has been made to the work of others.

Abstract

The human visual system is capable of interpreting a remarkable variety of often subtle, learnt, characteristic behaviours. For instance we can determine the gender of a distant walking figure from their gait, interpret a facial expression as that of surprise, or identify suspicious behaviour in the movements of an individual within a car-park. Machine vision systems wishing to exploit such behavioural knowledge have been limited by the inaccuracies inherent in hand-crafted models and the absence of a unified framework for the perception of powerful behaviour models.

The research described in this thesis attempts to address these limitations, using a statistical modelling approach to provide a framework in which detailed behavioural knowledge is acquired from the observation of long image sequences. The core of the behaviour modelling framework is an optimised sample-set representation of the probability density in a behaviour space defined by a novel temporal pattern formation strategy.

This representation of behaviour is both concise and accurate and facilitates the recognition of actions or events and the assessment of behaviour typicality. The inclusion of generative capabilities is achieved via the addition of a learnt stochastic process model, thus facilitating the generation of predictions and realistic sample behaviours. Experimental results demonstrate the acquisition of behaviour models and suggest a variety of possible applications, including automated visual surveillance, object tracking, gesture recognition, and the generation of realistic object behaviours within animations, virtual worlds, and computer generated film sequences.

The utility of the behaviour modelling framework is further extended through the modelling of object interaction. Two separate approaches are presented, and a technique is developed which, using learnt models of joint behaviour together with a stochastic tracking algorithm, can be used to equip a virtual object with the ability to interact in a natural way. Experimental results demonstrate the simulation of a plausible virtual partner during interaction between a user and the machine.

Acknowledgements

I would like to thank my colleagues in the Vision Group for many stimulating and profitable discussions, Aphrodite Galata for the provision of experimental shape and interaction data, and in particular, David Hogg for encouragement and guidance throughout the period of this research.

Declarations

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

N. Johnson and D. Hogg. “Learning the Distribution of Object Trajectories for Event Recognition”. In *Proceedings British Machine Vision Conference*, volume 2, pages 583–592, September 1995.

N. Johnson and D. Hogg. “Learning the distribution of object trajectories for event recognition”. *Image and Vision Computing*, 14(8):609–615, August 1996.

N. Johnson, A. Galata and D. Hogg. “The Acquisition and Use of Interaction Behaviour Models”. In *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 866–871, June 1998.

Contents

List of Figures

Chapter 1

Introduction

The human visual system is capable of interpreting a remarkable variety of often subtle, learnt, characteristic behaviours. For instance we can determine the gender of a distant walking figure from their gait, interpret a facial expression as that of surprise, or identify suspicious behaviour in the movements of an individual within a car-park. Machine vision systems wishing to exploit such behavioural knowledge have been limited by the inaccuracies inherent in hand-crafted models and the absence of a unified framework for the perception of powerful behaviour models. The research described in this thesis was motivated by a desire to address these limitations and provide a framework allowing the perception of effective models of characteristic object behaviours from the continuous observation of long image sequences.

The perception of behaviour implies that behavioural knowledge is derived empirically, thus favouring a low-level statistical modelling approach, where detailed behavioural knowledge evolves as learning proceeds. A natural learning process should enable model acquisition with a minimum of human intervention and should allow gradual adaptation, enabling model evolution with occasional changes in characteristic behaviour. Such a system would thus enable the acquisition of detailed behavioural knowledge from observation alone and, provided the resulting models were both analytic and generative, would have a wide range of applications.

The analysis of behaviour is fundamental to tasks such as automated visual surveillance and gesture recognition which are concerned with the interpretation of observed behaviours. Statistically based behaviour models allow these tasks to be approached without the need for *a priori* knowledge. Attentional control mechanisms which identify interesting incidents or actions can be implemented from the assessment of behaviour typicality, and the recognition of events or gestures achieved through the identification and semantic labelling of related classes of behaviour. Figure 1.1 illustrates an example of behaviour analysis. Using a learnt behaviour model, typicality assessment is performed based on the entire history of each tracked pedestrian's location within the scene (illustrated by the white trajectories). Whilst the behaviour of two of the observed pedestrians is judged to be typical (indicated by blue circles), the behaviour of the third - the individual loitering by a chained bicycle - is judged to be an atypical incident (indicated by a red triangle).

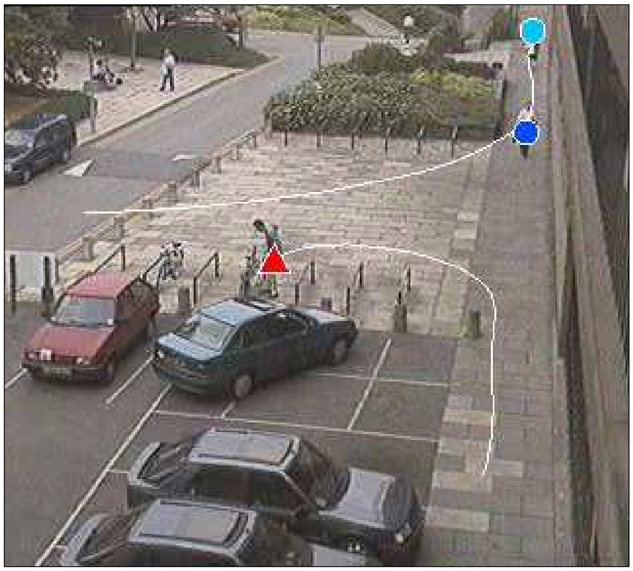


Figure 1.1: Sample behaviour analysis - incident detection through typicality assessment.

The generation of behaviour from statistically based models allows the prediction or extrapolation of future behaviours as well as the generation of realistic sample behaviours. Maximum likelihood behaviour predictions and extrapolations can be used to increase the robustness of object tracking and to aid in occlusion reasoning, whilst stochastically generated predictions can be exploited within stochastic tracking algorithms. Both maximum likelihood and stochastically generated sample behaviours can be applied to virtual objects within animations or virtual worlds, thus increasing realism. Figure 1.2 illustrates an example of behaviour generation. Using a learnt behaviour model, a short sequence of extrapolated behaviour (illustrated by the unfilled contours) is generated during the tracking of an exercise routine. This maximum likelihood extrapolation is based on a history of recently observed object shape (illustrated by the filled contours), and could clearly be a valuable aid to tracking.

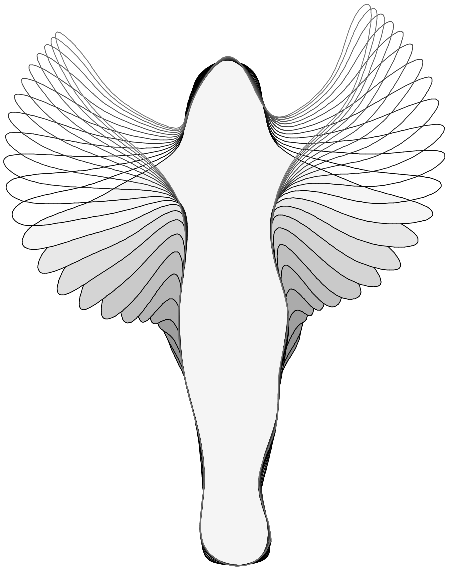


Figure 1.2: Sample behaviour generation - extrapolation of an exercise routine.

1.1 Object behaviour modelling philosophy

Before discussing the approach to object behaviour modelling adopted within this research, it is useful to consider the types of objects and behaviours which may be of interest, and to produce a suitable definition of behaviour. Intuitively, objects (spatial entities) with measurable dynamic characteristics which conform to some structured pattern may be seen as objects with interesting behaviours. For instance, pedestrians have a number of such characteristics, in particular their location within a scene, the shape of their silhouette, and perhaps their texture or interaction. Many other objects such as moving vehicles and flocks of birds share these interesting characteristics.

The evolution of these characteristics can be considered from a short-term point of view, relating to instantaneous changes, or a longer-term point of view, relating to (possibly entire) temporal histories. The following definitions, stated in order of increasing temporal extent, are presented in order to clarify this interpretation of object behaviour and to provide a basis for the adopted modelling approach:

- *The **state** of a particular measurable object characteristic is the current measurement of the characteristic together with its first derivative, and can be represented as a situated vector within the characteristic's measurement space.*

- *The **behaviour** of a particular measurable object characteristic is a (possibly entire) temporal history of the characteristic's measurements, and can be represented as a continuous trajectory within the characteristic's measurement space.*

1.1.1 Adopted modelling approach

The first stage of the perceptual process involves identifying objects within the scene and generating feature vectors representing those characteristics which are of interest. Within this research, this is achieved by employing existing tracking systems developed by Baumberg and Hogg [5, 4, 7] to track moving objects within real world scenes, resulting in frame by frame updates to the relevant characteristic for each uniquely labelled object.

The extended observation of object characteristics exhibiting interesting behaviours will define probability distributions over the *state spaces* and *behaviour spaces* of the characteristics. It is these distributions, defining the space of observed behaviours and their probabilities, which will be acquired during model learning. Such distributions are likely to be complex in structure and unsuitable for modelling using conventional parametric distributions. Instead, probability density over state and behaviour spaces is modelled by the distribution of prototype vectors placed in an unsupervised manner using a robust Vector Quantization algorithm. This technique provides the desired natural learning process, resulting in a representation of probability density which is both concise and accurate and which facilitates typicality assessment and attentional control.

The perception of behaviour is thus achieved by transforming sequences of observed characteristics into their state and behaviour spaces where they are used as training data for the unsupervised learning process. Although the state space associated with a particular characteristic is simply a vector space describing measurements and their first derivatives, developing a representation describing a behaviour space is non-trivial since the representation must encode spatio-temporal trajectories of different lengths. A novel spatio-temporal trajectory representation is developed which utilises the corresponding state density model and uses a temporal pattern formation strategy to encode different length behaviour sequences.

The discrete nature of this probability density representation is beneficial to semantic labelling and allows generative tasks to be performed using a transition-based prediction scheme. The parameters of this probabilistic prediction scheme are derived during a further unsupervised learning phase, resulting in a model where the production of both maximum likelihood and stochastically generated predictions, extrapolations, and sample behaviours is possible.

1.2 Overview of the thesis

This introductory discussion has identified the need within computer vision for a framework allowing the perception of powerful object behaviour models from the observation of image sequences, has provided a concise definition of object behaviour, and has given a broad outline of the approach adopted within this research to providing such a framework. Related research, describing techniques for the characterisation of motions and behaviours which may be broadly considered as behaviour modelling, is reviewed in Chapter 2. The remainder of the thesis gives a detailed description of the original research undertaken, including relevant experimental results, with descriptions of related techniques included where relevant.

After an overview of the acquisition, pre-processing, and properties of the experimental data used within this thesis, Chapter 3 describes a robust technique for the unsupervised learning of probability density over state and behaviour spaces. Using this technique, models of characteristic object states and behaviours are developed, where the modelling of object behaviours is achieved using a novel spatio-temporal trajectory representation. Finally, typicality assessment and incident detection using these learnt state and behaviour models is demonstrated.

Since the models developed in Chapter 3 are non-generative, Chapter 4 describes the enhancement of state and behaviour models to include generative capabilities via the superimposition of a learnt probabilistic prediction scheme. Using this technique, both maximum likelihood behaviour extrapolation and the stochastic generation of realistic sample behaviours are demonstrated. To further demonstrate the utility of predictive models, the performance of both state-based and behaviour-based predictors is compared with a linear prediction scheme. Finally, the similarities between the enhanced models and Hidden Markov Models, commonly used for the recognition of gesture and speech, are discussed.

Throughout the development of the behaviour modelling framework, it is the behaviours of single objects which are considered. To extend the utility of the framework, the modelling of object interaction is investigated in Chapter 5. Object interaction is a particularly interesting form of behaviour since it allows reasoning to be extended from individuals to groups of objects, whilst providing a machine with the ability to learn and use models of natural interaction may prove beneficial to the provision of natural user-machine interaction. Two approaches to binary interaction modelling are investigated. The first approach considers the statistical co-occurrence of events within models of the state or behaviour of individual objects, whilst the second approach attempts to explicitly model interaction as joint behaviour. This latter approach is used within a stochastic tracking algorithm to demonstrate how a learnt joint behaviour model can be used to equip a virtual object with the ability to interact in a natural way.

Finally, in Chapter 6, the thesis is summarised, some general conclusions are drawn, and possibilities for future research are discussed.

Chapter 2

Modelling object motions and behaviours: A review

In recent years, many researchers have become interested in techniques allowing the characterisation of complex motions and behaviours. This has coincided with a shift in attention from the interpretation of static images to the interpretation of image sequences. The analysis of object motions and behaviours offers to impart a richer understanding of a dynamic world than that available from the analysis of static scenes. The focus of much of this research has been the analysis of human behaviours, motivated by applications such as perceptual user-machine interfaces, automated visual surveillance systems, and realistic virtual environments. In this chapter, several current approaches which may broadly be considered as modelling object motions and behaviours are reviewed, focusing on approaches from object tracking, automated visual surveillance, gesture recognition and computer graphics. Reviews of techniques for motion-based recognition can also be found in the survey paper of Cédras and Shah [19] or the introductory paper [82] to a collection of relevant papers edited by Shah and Jain [81]. The remainder of this introduction discusses some of the most significant attributes by which the various approaches may be compared.

Model-based object recognition from static images has been based largely on shape information, using a wide range of 2-D features such as edges, curves, and regions, and 3-D features, such as surface patches and cylinders. As attention has moved to the interpretation of image sequences, the relative importance of motion over shape when performing different tasks has been reflected by the extent to which it has been included in the relevant models. At one end of this spectrum are models which are essentially shape based but include some simple motion information. For example, the deformable 2-D contour models developed by Baumberg and Hogg [4] for tracking articulated and non-rigid motion use the conventional Point Distribution Model (PDM) of Cootes *et al.* [22, 20] to represent shape variation, whilst assuming objects undergo uniform 2-D motion with random acceleration and shape change. At the other extreme are models which include detailed motion information. For example, Bobick and Davis [12] recognise actions using temporal templates constructed from *motion-energy* and *motion-history* images which identify the location and recency of motion.

Recognition based entirely on motion has been motivated by the ability of humans to recognise many different kinds of motion. Our ability to perform recognition based purely on motion information was first demonstrated by Johansson's pioneering work on Moving Light Displays (MLDs) [49], where the trajectories of small reflective patches attached to the actor's joints were shown to be sufficient information for the recognition of walking. Such results also suggest that the recognition of behaviour is feasible using only sparse view-based information. This is reflected within the literature by the diverse set of features used within models, ranging from 2-D view-based features such as optical flow and object silhouette, to explicit 3-D object representations. For example, Black *et al.* [9] use parameterised models of optical flow to recognise facial expressions and articulated motions, whilst Hogg [41, 42], and later Rohr [76], use a set of cylinders to model the human body with joint curves to model walking.

Hand-crafted models, such as those used by both Hogg and Rohr, embody both knowledge and constraints which are *a priori* in nature in the sense that they are not derived empirically by the system. The extent to which a model is based on *a priori* knowledge and constraints often limits its realism and utility due to implicit inaccuracies. All modelling frameworks introduce some *a priori* constraints by virtue of the assumptions and simplifications they make. For example, shape descriptions based on B-splines, such as those used by Blake *et al.* [10], impose a constraint on the shape of each curve segment. Hand-crafted knowledge is often inaccurate and fails to describe the variations which are evident in real behaviour. This type of *a priori* knowledge is increasingly being avoided by using statistical estimation techniques. For example, approaches based on the PDM incorporate knowledge of variability which is derived from training data using a Principle Components Analysis (PCA). Ideally, models should be acquired via a learning scheme which allows the continuous evolution of knowledge with empirical evidence.

Whilst statistical estimation techniques improve the accuracy of knowledge, it is also desirable to maximise a model's specificity and compactness. A highly specific model will include only valid information, and a highly compact model will require a minimum of parameters to describe that information. For example, one of the principal limitations of the PDM is its non-specificity when modelling shapes which undergo complex non-rigid and articulated deformations. This limitation is addressed by techniques such as the Hierarchical PDM proposed by Heap and Hogg [38] or the use of Gaussian Mixture Models as proposed by Cootes and Taylor [21]. The similarities which exist between these recently published methods and the research described in this thesis is an indication of the importance of highly specific models as attempts are made to model more complex objects and behaviours.

Since object motions and behaviours are spatio-temporal entities, the way temporal information is represented within a model is of fundamental importance. One of the principal limitations of many of the methods discussed in this chapter is the low order of the dynamics modelled. Whilst first or second order dynamics may be sufficient to model relatively short-term effects, they will inevitably fail to represent the dynamics of many real behaviours which are likely to involve a higher temporal dependence - one of the novel aspects of the research described in this thesis is to model the entire temporal extent of variable length behaviours. Another important aspect of temporal information representation is time-scale invariance. In many gesture recognition problems, time-scale invariance is considered advantageous, since only the temporal ordering of the gesture is important, whilst in automated visual surveillance tasks the rate at which an action is performed is also important.

Since existing models of object motions and behaviours have been designed for specific tasks, they generally lack the range of analytic and generative capabilities required for a unified framework. This is largely due to limitations in the modelling techniques employed. A wide variety of techniques have been investigated, many of which have been adopted from other disciplines. For example, many of the techniques originate from areas such as signal processing, pattern recognition, and statistical modelling. In particular, there is a high degree of correspondence between the problem of gesture recognition and that of continuous speech recognition. This similarity has recently resulted in a number of attempts to apply techniques used in continuous speech recognition, such as Dynamic Time Warping (DTW) (see, for example, Huang *et al.* [45]) or its successor, Hidden Markov Modelling (HMM) (see Rabiner and Juang [68] or, for example, Huang *et al.* [45]), to gesture recognition problems.

2.1 Object tracking

Central to the automated interpretation of image sequences is the task of locating and tracking specific classes of object. For example, automated visual surveillance systems require real-time information about the location of the different classes of objects under observation in order to reason about their behaviours and interactions, whilst most gesture recognition systems require detailed, real-time, information about object pose in order to interpret behaviours. Real image sequences are typically subject to the presence of noise and background clutter, thus demanding models of an object's spatial characteristics, and increasingly its dynamics, to help locate possible model instances within the scene. Models of motion and dynamics allow an object tracker to better predict the expected location and pose of the object in future time instants, thus improving performance. The techniques used to model such information in a number of the key approaches are discussed in this section.

The work by Hogg on tracking humans [41, 42] is typical of hand-crafted modelling approaches. Hogg's WALKER model uses a set of cylinders to represent rigid body parts, with posture represented by parameterised joint angles. Walking is represented by a canonical walk cycle, modelled by periodic functions of a single pose parameter which were precomputed by analysis of a single walk sequence. The space of possible walks is represented by a set of hand-crafted constraints on factors such as the rate at which the pose parameter may vary and the overall speed of motion of the body. In particular, individual walking styles are permitted by allowing each joint angle to be advanced or retarded relative to the pose parameter. Rohr [76] uses a similar model, based on the work of Hogg, where the joint curves were derived from medical motion studies of sixty males.

Many recent frameworks for tracking non-rigid and articulated objects are based on the PDM of Cootes *et al.* [22, 20] where a model of shape variation is derived from a statistical analysis of training data. The work of Baumberg and Hogg [5, 4] is typical of such approaches, using a deformable contour model to represent the shape variation of a walking human, whilst also demonstrating the automatic acquisition of training data. A simple stochastic process models shape change, with an additive, isotropic noise process associated with each shape parameter. Objects are assumed to undergo uniform 2D motion with random acceleration, again modelled by an additive, isotropic noise process. This approach is extended in recent work by Heap and Hogg, where specificity is improved by representing shape using a number of smaller PDMs [38], and discontinuous shape changes are handled by modelling transitions between these separate models [39]. These transitions also serve to provide a crude statistical model of the dynamics of the object.

Blake *et al.* [10] use learnt second order stochastic difference equations to model more complex dynamics for contour tracking. In this approach, the matrix coefficients governing both the deterministic dynamics, and the coupling into the system of the stochastic noise process, are learnt from observation. This learning process involves maximum likelihood estimation of the parameters via least-squares minimisation. This approach is extended by Reynard *et al.* [71] to decouple *class* and *dynamical* variability, and by North and Blake [61] to improve the robustness of dynamics learnt from noisy training data by using the Expectation-Maximisation (EM) algorithm (see Dempster *et al.* [25] or, for example, Ripley [74] or Huang *et al.* [45]) to perform maximum likelihood estimation.

Recent work by Isard and Blake [47] attempts to extend the range of motions which can be modelled using stochastic difference equations by allowing dynamics to be represented by multiple models. Multiple models are supported naturally within the framework of their CONDENSATION tracking algorithm [46], where the addition of probabilities governing the transition between models allows automatic model switching to occur when appropriate. As well as allowing more complex dynamics to be modelled, the recognition of different classes of motion is facilitated by virtue of the model switching.

Baumberg and Hogg also model more complex dynamics in their spatio-temporal model [6]. This approach extends the approach of Blake and Isard to automatically learn dynamics which are constrained to be physically plausible, resulting in trained ‘vibration modes’ which are orthogonal and can thus be tracked independently. An object is considered to be an elastically deformable physical system with certain material properties, in the context of the Finite Element Method from engineering, and the set of vibration modes describing object dynamics (and implicitly defining physical properties of the system such as stiffness) are learnt from the analysis of training examples using the method introduced by Blake and Isard.

A higher order model for predicting the shape and position of deformable contours is investigated by Xu and Hogg [90]. This method is based on the PDM parameterisation of Baumberg and Hogg but uses a set of simple recurrent neural networks (see Elman [27] or, for example, Haykin [37]) to predict the value of each parameter in the next time-step - a multivariate time series prediction approach. Each network models typical nonlinear dynamics via the cooperation between a number of sets of time delayed inputs and a set of exponential memory units, and is taught separately from a set of training examples. Each set of inputs represents the state of the contour at a particular time instant by a set of shape parameters and a position within the image plane.

If an object exhibits continuous motion, it is possible to perform very crude tracking using techniques based solely on motion, such as change detection, background subtraction, and optical flow. For example, Baumberg and Hogg [5] use background subtraction to automatically acquire the training data for their deformable contour model, whilst Niyogi and Adelson [60] build a spatio-temporal model around the volume created in XYT space when change detection is applied to image sequences of a person walking. In this latter approach, the canonical walk is modelled by a smooth spatio-temporal surface generated by combining data from several image sequences. The surface is parameterised by spatial position and scale, and temporal period and phase, thus exploiting the periodicity of the gait. An individual walk is expressed as a combination of the canonical walk and a deviation surface that is specific to the individual.

Yacoob and Davis [92] attempt to learn models of articulated motion based on optical flow. This approach uses a parametric model of body part dynamics, the ‘cardboard body’ model introduced by Black *et al.* [9], where activity is described by the relative motion of a number of planar patches which are constrained to exhibit similar motion at given articulation points. This model, which is used to acquire training data, assumes that image flow is either constant, or satisfies constant acceleration, over the small temporal windows over which model parameters are estimated. In order to learn periodic, articulated activities such as walking, sequences of model parameters covering one entire period of the activity are generated. These training sequences are then analysed using a PCA to learn a low-dimensional model of the complex dynamics underlying the activity.

2.2 Automated visual surveillance

There is an increasing interest in visual surveillance in many aspects of modern life. For example, the monitoring of business and residential properties, city centres, and car parks offers to address the perceived increase in levels of violence and crime, whilst the monitoring of livestock offers to improve animal welfare by analysing behaviour under different living conditions. In the future, it will not be feasible for human operators to process the huge volume of information generated, and thus the automation of visual surveillance tasks is essential. Central to the automation of these tasks is the understanding of complex object behaviours, and thus models of these behaviours are fundamental. The techniques used to model such information in a number of the key approaches are discussed in this section.

One of the simplest visual surveillance tasks is the filtering of alarm events from a perimeter intrusion detection system. Such a system consists of a variety of alarm devices which, when triggered, activate cameras viewing the scene to capture an image sequence spanning the alarm event. Rosin and Ellis [77] describe a vision system for the analysis of these image sequences, discriminating between alarms triggered by human intruders and false alarms caused by animals or other causes. The classification of alarm events is based on hand-crafted knowledge about the scene and the appearance and dynamic behaviour of target objects which are tracked using background subtraction. This knowledge is modelled using a classical frame-based structure where dynamic behaviour is modelled by the range of maximum speed and acceleration values expected over a sequence, whilst other basic behavioural knowledge such as the expected location or time of day of an appearance is included where relevant.

A more effective visual surveillance system must be capable of a wider range of tasks than simply alarm event filtering. Such tasks may include incident detection and classification, the generation of conceptual event descriptions, and the generation of warnings relating to predicted future incidents. The VIEWS (Visual Inspection and Evaluation of Wide-area Scenes) project described by Corrall and Hill [23] is an example of a visual surveillance system designed primarily for incident detection and classification. The system relies on detailed hand-crafted knowledge of the scene, the objects to be identified, and the specific events and behaviours to be recognised. Knowledge of the scene layout is modelled using the spatial representation described by Howarth and Buxton

[44], where the scene is partitioned into semantically relevant regions. Simple events are generated when an object (identified by a model-based vehicle tracker) undergoes a state change based on properties such as speed, region occupancy, or proximity to another object. Event and interaction histories are maintained and matched against hand-crafted behaviour clauses to facilitate recognition. An alternative approach described by Howarth and Buxton [43] uses a combination of static and dynamic Bayesian belief networks to model and evaluate behaviours under attentional control. This approach is extended by Buxton and Gong [17] to improve tracker robustness by modelling constraints relating to object motion and size.

The delivery of natural language (or conceptual) descriptions within automated visual surveillance has been discussed by Nagel [59], where relevant approaches from the road traffic domain are reviewed. The provision of a running commentary of sporting events is an application of such systems which has received much interest. For example, the VITRA (VIsual TRAnslator) project described by Herzog and Wazinski [40] has been demonstrated on short sequences obtained from a static camera viewing a football match. In this domain, incremental event recognition is required since a retrospective description of behaviours is inadequate. The system uses a hand-crafted scene model and events are represented by *course diagrams* - directed graphs labelled with conditions, as described by André *et al.* [1]. Incremental event recognition is achieved as graph edges are traversed, triggered by updates to the configuration of objects within the scene. An extension to this system, enabling the recognition of intentions using a hand-crafted plan hierarchy detailing stereotypical tactics, is described by Retz-Schmidt [70].

A potentially more powerful approach than using hand-crafted knowledge is to introduce model learning. The use of statistically-based models and learning techniques allows knowledge to be acquired from observation in an unsupervised manner. A method for learning semantic scene partitioning, based on the spatial representation of Howarth and Buxton, is proposed by Fernyhough *et al.* [29]. This method uses an object tracker to gather instances of regions representing the accumulation of image pixels occupied by an object as it moves along its path. Regions are maintained within a database, and those with a high degree of overlap are merged to generate a frequency distribution identifying the most commonly used path. After learning, extraction of the most commonly used paths, and removal of low frequency paths, results in the desired scene partitioning. Fernyhough *et al.* [28] also demonstrate how event models can be generated from a statistical analysis of training data. The learnt spatial representation is augmented with temporal indexing information, which, together with a qualitative notion of proximity, is used to generate descriptions of commonly occurring binary object interactions.

Using statistics acquired from observation, some visual surveillance tasks may be achieved without the need for scene knowledge. Morris and Hogg [57] present a method for assessing the likelihood of object trajectories, based on the interactions between a moving object and other static objects within the scene. Trajectories are characterised by sets of landmark points which identify interactions (points of closest proximity) between the moving object and the closest static object. The cumulative probability distribution of a set of descriptive measurements, made at each landmark point, is calculated from training data, and used to assign a probability to each interaction in a sequence. Finally, trajectories are classified as typical or atypical by thresholding the weighted sum of the lowest few probabilities, using weights obtained during a supervised training phase.

In order to produce predictions of future behaviours, more detailed statistical models are required. Gong and Buxton [34] have investigated modelling simple object motion characteristics from which visual expectations can be generated to guide the perception process, using techniques commonly used for speech and gesture recognition (see Section 2.3). Initially, HMMs of fixed topology, modelling discretised vehicle orientations and displacements, are learnt from the observation of a small number of training sequences. Due to the uniqueness of the maximum likelihood expectations generated from these models, and the instability of stochastically generated expectations, Gong and Buxton instead propose the use of Augmented Hidden Markov Models (AHMMs), as introduced by Rimey and Brown [73]. AHMMs allow model parameters to be modified during the observation process to reflect current visual evidence. A modification gain determines the influence of the new visual evidence, whilst a decay gain ensures that changes dissipate as the visual evidence for them weakens.

An extension to the HMM framework for modelling interacting processes, the Coupled Hidden Markov Model (CHMM), has recently been applied to surveillance tasks by Oliver *et al.* [62]. The CHMM framework, introduced by Brand *et al.* [14] for action recognition (see Section 2.3), allows the hidden states of individual chains to be coupled via matrices of conditional probabilities modelling causal influences between the processes. Oliver *et al.* use small CHMMs with unconstrained structure to model a number of simple interactive behaviours such as following and meeting. Models are learnt from training sequences generated by synthetic agents designed to mimic simple human behaviours. Each training sequence corresponds to a pair of nearby pedestrians, and encodes simple relative motion parameters which are invariant to both the absolute position and direction of the agents and to the scene.

Another statistical approach, PCA, which is commonly used within object trackers in the form of the PDM (see Section 2.1), has also been applied to behaviour modelling tasks. Sumpter *et al.* [85] describe an approach to modelling interactive animal behaviour where the PDM is extending to include non-shape parameters governing the interaction, such as relative separation and velocity. The inclusion of these parameters requires that their influence in the model is correctly scaled, and here an information-theoretic solution, the maximisation of *eigen-entropy*, is proposed.

2.3 Gesture recognition

Research into the recognition of human actions, gestures, and facial expressions has provided perhaps the richest set of spatio-temporal behaviour models to date. This is due to the diverse set of behaviours considered, their relative complexity, and the wide range of features available for recognition. Much of this research has been motivated by an interest in developing techniques to allow a more natural form of interface between the user and the machine, utilising interactive spaces (such as the Interactive Virtual Environments described by Pentland [63]) equipped with cameras and microphones where such techniques can be developed and tested. A number of the key behaviour modelling techniques, many of which are similar in spirit to the techniques described in this thesis, are discussed in this section.

The recognition of actions and gestures is often achieved by considering some abstraction of the trajectories traced in measurement space as a particular gesture is performed. For example, Nagaya *et al.* [58] propose the use of polygonal approximations to pattern space trajectories for gesture modelling. In this approach, a pattern space is defined in which each point represents a unique image. The continuous trajectories traced within this space by specific gestures are segmented at points of maximum and minimum curvature, and the polygonal approximation defined in terms of the relative distance and angle formed between these landmark points. Assuming that the object of interest does not extend beyond the image boundary, and that the background is static, this representation of a gesture is shown to be invariant to affine transformations of the object. Recognition is achieved using dynamic programming to select the gesture model which best matches the input sequence.

Bobick and Davis [12] also model gesture without direct recognition of the object performing the action. In their approach, a number of view-specific *temporal templates* are used for gesture modelling. Each template consists of two components, a binary *motion-energy* image indicating where motion has occurred within the image, and an integer-valued *motion-history* image indicating the recency of motion and thus encoding a history of the motion defining the gesture. The actions described by these motion-history images are immediately visually apparent due to the images' motion blurred appearance. Temporal templates for each action are collected from a variety of viewing angles and characterised by a set of statistical moment-based features, allowing recognition to be achieved by matching based on the similarity between feature sets.

Since spatio-temporal trajectories are continuous multivariate time series, neural networks provide a natural modelling framework, allowing both recognition and prediction. Psarrou *et al.* [67] develop a framework for the recognition of face sequences, based on a partially recurrent neural network with exponential memory (see Elman [27] or, for example, Haykin [37]) and the *eigenface* representation of Turk and Pentland [87]. For each face class (individual), a set of eigenface models are acquired, from fixed length image sequences of face movements, to represent the temporal face sub-space of that class. Trajectories formed by projecting successive image frames of a face sequence into this temporal face sub-space are learnt by the neural network, and the temporal changes over these trajectories used as a temporal face signature for recognition.

The recognition and prediction of human motion using neural networks has been investigated by Bulpitt and Allinson [16, 15]. Using data acquired from the analysis of MLDs of actors performing different activities, a network incorporating two interacting Adaptive Resonance Theory (ART) networks (see Carpenter and Grossberg [18] or, for example, Ripley [74]) is used to learn parameterised motion trajectories. The first ART network is used to distinguish between the different instantaneous patterns in each sample of a motion sequence, whilst the second ART network is used to learn the temporal relationship between these events, utilising a temporal decay operator to provide the network with memory. Recognition of a particular sequence or sub-sequence is based on classification of the pattern of activation on the output layer.

Just as the first network in Bulpitt's architecture distinguishes the different instantaneous patterns that form a trajectory, so many other approaches use sequences of discrete states to represent a trajectory. For example, Bobick and Wilson [13] generate descriptions of a gesture and its variability based on sequences of configuration states. Initially, a prototype gesture trajectory is generated by fitting a principal curve to noisy training trajectories in a configuration space, using a time-collapsing technique to maintain temporal ordering. Fuzzy states are then generated by clustering the vectors defining the prototype trajectory and fitting a single oriented Gaussian at each state to define the local variability. Finally, recognition is achieved using a matching technique based on dynamic programming.

Hidden Markov Models (HMMs) are a popular, state-based, probabilistic mechanism for describing the temporal structure of time-varying processes, although they are generally limited by factors such as the first-order process assumption and the local optima encountered when learning models with many free parameters. HMMs have been extensively used for speech recognition tasks (see, for example, Huang *et al.* [45]), and have recently become popular for describing the temporal structure of actions and gestures. For example, Yamato *et al.* [93] use HMMs of unconstrained topology to model different tennis swings. Simple region-based features are derived for each training sequence image and a set of discrete observation symbols generated using Vector Quantization (VQ) (see, for example, Linde *et al.* [54] Gray [35], or Gersho and Gray [33]). In learning the resulting symbol sequences, Yamato *et al.* report that the globally optimal model is not always found.

HMMs with continuous observation distributions have also been applied to gesture recognition. Starner and Pentland [84] use HMMs of fixed topology to model American Sign Language from relatively low resolution hand tracking. A single model is associated with each sign and contains just four states with forward and skip transitions. The probability of observing a particular hand configuration (position, orientation, and eccentricity of each hand's bounding ellipse) at each state is modelled by a single Gaussian. Since the model contains no contextual information, a word level grammar is used to increase recognition accuracy.

Since it may be advantageous to consider a number of different sets of features concurrently when recognising gestures, Wilson and Bobick [89] propose an extension to the HMM framework in which multiple models are maintained at each state. A normal joint distribution is used to model the probability of observing a particular set of features from the multiple view-based representations at each state, based on the distance from each observed feature to the relevant model sub-space. Small HMMs with unconstrained topology are used and the parameters of the multiple representations at each state are estimated concurrently with the learning of HMM parameters.

One of the most attractive features of HMMs is their time-scale invariance, allowing the recognition of a single action performed at varying speeds. Such time-scale invariance in recognition can also be achieved using the Dynamic Time Warping (DTW) algorithm which, like HMM, has been extensively used by the speech recognition community (see, for example, Huang *et al.* [45]). For example, Darell and Pentland [24] represent gestures using sets of key-frames and use sequences of normalised correlation scores and DTW to match gestures, whilst Gavrila and Davis [32] use simple joint-angle parameterised 3-D pose templates and DTW to match gestures in MLDs.

HMMs have also been applied to problems involving multiple interacting processes. Since the conventional HMM framework assumes a single process, Brand *et al.* [14] have recently introduced an extended framework which they call Coupled Hidden Markov Models (CHMMs), in which multiple HMMs are coupled via matrices of conditional probabilities which reflect the causal influences between processes which are neither independent nor wholly mutually determined. In experiments based on the recognition of T'ai Chi gestures, each gesture is represented by sequences of 3-D hand positions. Each hand is considered to be a separate process, and the gestural behaviour (an interaction between the hands) is modelled by small CHMMs with unconstrained structure. An alternative approach, which obtains a much poorer classification accuracy, is to model the interaction as a single process using a single HMM. Such modelling of interacting processes as *joint* behaviour is also used by Kakusho *et al.* for the recognition of social dancing [50], where a particular dance is modelled by a sequence of constituent figures, each characterised by coarse overall motions of the pair.

Another statistical approach, PCA, which is commonly used within object trackers in the form of the PDM (see Section 2.1), has also been applied to gesture recognition. Yacoob and Davis [91] learn models of activity and the variability in activity caused by natural variation and *admissible transformations* (such as time scaling, differences in viewing angle, and partial data). Activities are tracked using the 'cardboard body' model introduced by Black *et al.* [9], where activity is described by measurements on a number of planar patches, and the model is learnt from a PCA of a number of exemplar actions. Recognition is based on matching an observed sequence to model instances, allowing admissible transformations on the observation to minimise a matching error.

The periodic nature of many natural motions, particularly human activities such as walking, provides a strong cue for recognition. This is reflected by a number of approaches to action and gesture recognition which analyse cyclic or near-periodic motions. For example, Polana and Nelson [66] identify three distinct categories of motion based on the extent of any spatial or temporal repetition. The first of these classes, *temporal textures*, comprises repetitive motions with indeterminate spatial and temporal extent, such as the motion of leaves in the wind, which can be recognised from the statistical properties of an optical flow field. The second class, *activities*, comprises periodic, spatially compact, motions such as walking which can be recognised from analysis of the periodicity of low-level image motion (see Polana and Nelson [65]). Tracking is initially performed by identifying areas of motion, resulting in sequences consisting of activity at a constant position and scale. By generating low resolution flow magnitude templates, and deriving a periodicity measure from a Fourier analysis of the motion magnitude sequence for each cell, the period of an action can be established. Mean flow magnitude templates covering a single period are generated, and recognition achieved by locating the nearest (bounded) reference template.

Liu and Picard [55] present a method for locating regions of periodicity which is more robust than the method of Polana and Nelson and does not require optical flow computation. After frame alignment (keeping the object of interest stationary), a robust periodicity measure is derived from a Fourier analysis of the intensities of each image pixel over the sequence. This process can be considered as a low-level periodicity filter, resulting in a periodicity template which identifies image regions in which periodic events occur, also giving a measure of the amount of periodic energy at each pixel and the fundamental frequency of the behaviour. Using this template, different periodic actions can be located and classified.

One of the problems with the above approaches is that they only identify strictly periodic behaviours. Seitz and Dyer [80] describe an approach to the view-independent analysis of behaviours which repeat but are irregular - cyclic behaviours. Using an image matching procedure which is invariant to affine transformations, a set of functions describing the combined length of the instantaneous period, over different numbers of cycles in the past and future, is estimated. These functions, known as the *period trace* of the sequence, allow cyclic behaviours to be analysed, resulting in the identification of irregular intervals and various characteristic features of the cyclic behaviour.

2.4 Computer graphics

In recent years, increases in processor speed and advances in both hardware and software graphics technology has made the rendering of fairly convincing animations and virtual environments a reality on moderately priced workstations. In order to further improve realism, it has become necessary to equip the dynamic objects appearing in such graphics with more realistic motions and behaviours. For example, animated humans need to be able to walk, run, interact etc. in a convincing manner, whilst autonomous characters in games and virtual environments can be made to appear more intelligent using behaviour-based control systems. The techniques used to model and simulate such realistic motions and behaviours in a number of the key approaches are discussed in this section.

Virtual humans, such as the Jack system of Badler *et al.* [3, 2], offer to provide both a substitute for real humans in the domain of computer-based design, and realistic representations of ourselves within virtual environments (and, in the future, realistic autonomous characters and virtual actors). Such systems typically employ motion and behaviour models which are either based on predefined motions or which utilise physically-based control strategies. Predefined motions are typically derived from biomechanical or motion capture data and are similar to the motion model used by Hogg [42] in his pedestrian tracker (see Section 2.1). To increase movement variability and add 'personality' to such motions, Perlin [64] adds periodic noise to the joint transformations. Physically-based animation can lead to more general locomotion solutions, but requires powerful control strategies such as the limit cycle control proposed by Laszlo *et al.* [52] to maintain stability during inherently unstable motions such as walking and running.

Autonomous creatures with realistic appearances, motions, and behaviours, have also been produced using hand-crafted physically-based models. For example, Tu and Terzopoulos [86] describe autonomous physically-based fish models which employ perceptual, behavioural, and motor control systems. Behavioural modelling is achieved using an intention generator which issues intentions based on the individual's habits, current mental state, and incoming sensory information. Behaviour routines are executed to attend to the current intention, resulting in the execution of the appropriate motor control routines. When generating such characters for interactive virtual environments, Blumberg and Galyean [11] propose that pure autonomy should not be the ultimate goal -

the ability to direct the behaviour at multiple levels may also be important. Blumberg and Galyean's virtual dog *Silas* is an autonomous creature which is also capable of responding to external control. A layered behavioural system allows external directives to be applied at three levels - *motivational*, *task*, and *direct*.

A more powerful approach than hand-crafting autonomous creatures is to use machine learning and evolutionary techniques. For example, Grzeszczuk and Terzopoulos [36] describe a technique which allows creatures with highly deformable bodies to learn locomotion automatically. By repeatedly attempting to improve locomotion using different actions, and remembering energetically efficient solutions, life-like locomotion is eventually achieved. Finally, the learnt, low-level, muscle control functions are abstracted to produce compact, efficient, motor controllers and higher-level motor tasks are learnt. Sims [83] describes a more general solution - the evolution of entire creatures. In this approach, both the morphology and control systems of simple creatures are evolved towards specific behaviours using genetic algorithms. The genetic representation of a particular creature encodes a directed graph structure describing both a hierarchy of body parts and the creature's nervous system, whilst the fitness of a particular creature is assessed within a simulated physical environment.

As well as the behaviours of single objects, the self-organising behaviour of large groups of objects is also of interest to the computer graphics community. The pioneering work of Reynolds [72] on the simulation of flocking behaviours has enabled the animation of complex flocking sequences without having to script the motion of each individual creature. Reynolds' *boids* (bird-oids) employ a distributed behavioural model in which each boid is an independent, identical actor. The behaviour of each individual is based on its perception of local flock-mates and the opposing forces of collision avoidance and an urge to join the flock. Although the behaviour of each individual is hand-crafted and relatively simple, the behaviour of the entire flock appears natural, complex, and unpredictable.

Chapter 3

Learning statistical behaviour models

After an overview of the acquisition, pre-processing, and properties of the experimental data used within this thesis, this chapter describes a robust technique for the unsupervised learning of probability density over state and behaviour spaces. Using this technique, models of characteristic object states and behaviours are developed, where the modelling of object behaviours is achieved using a novel spatio-temporal trajectory representation. Finally, typicality assessment and incident detection using these learnt state and behaviour models is demonstrated.

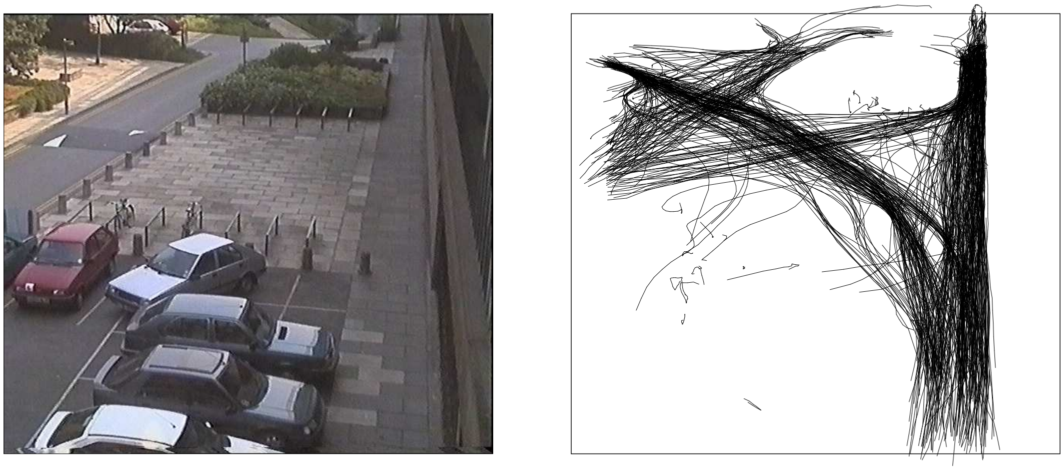
3.1 Experimental data acquisition, pre-processing, and properties

The research described in this thesis assumes the availability of experimental data representing the temporal evolution of particular behavioural characteristics. This raw data is the result of the initial stage of behaviour perception where objects of interest are identified and tracked within image sequences. All experimental data used in this research has been generated by employing existing tracking systems to track moving objects within real world scenes viewed with static cameras. A view-based approach to behavioural reasoning is adopted with two distinct object characteristics being considered - object location within the image plane and object silhouette shape. The use of view-based data avoids both the need for three-dimensional trackers and the introduction of errors associated with the transformation of coordinates from the image plane to a world coordinate system. The remainder of this section briefly identifies the techniques employed within the object trackers, some properties of the raw data they generate, and the pre-processing applied to the raw data.

3.1.1 Object location

Experimental location data is generated using an object tracker developed by Baumberg and Hogg [4, 7] which is based on the Active Shape Models of Cootes *et al.* [22, 20] and acquired automatically from observing long image sequences [5, 7]. This system provides efficient real time tracking of multiple articulated non-rigid objects in motion, and copes with moderate levels of occlusion. In our experiments, pedestrians are tracked in outdoor scenes using a previously acquired pedestrian shape model.

There is a one way flow of location data from the tracker consisting of frame by frame updates to the position in the image plane of the centroid (x', y') of uniquely labelled objects. Since each new object being tracked is allocated a unique identifier, it is possible to maintain a history of the path taken by each object from frame to frame. For example, Figure 3.1 shows a typical outdoor pedestrian scene, (a), and a set of smoothed, sub-sampled image plane trajectories representing the motion of pedestrians within the scene, (b).



(a) (b)

Figure 3.1: Sample location data: (a) pedestrian dominated scene, and (b) pedestrian trajectories.

In experiments considering the location of tracked pedestrians, image plane motion is typically slow and locally linear with respect to video frame rates whilst shape change is typically rapid and non-linear. Thus, although the tracker must operate at high frame rates to operate effectively, trajectory data can be sub-sampled with little or no loss of information, greatly reducing the volume of data to be processed. When processing live video streams, the tracker's frame rate varies depending on the number of objects being tracked at any particular instant. Thus the capture time of each new image frame must be recorded and a regularly sampled sequence obtained from piecewise linear interpolation of the raw data.

Further pre-processing of raw data aims to reduce the presence of noise associated with the tracking of spurious objects such as shadows and reflections, and to constrain characteristic vectors to lie approximately within a unit hypercube, thus simplifying subsequent stages of the perception process. Noise due to the tracking of spurious objects can be minimised by rejecting objects which exist for less than l frames (typically, $l \approx 50$), since image evidence supporting the existence of such objects is typically short-lived. The constraining of characteristic vectors is simply a matter of transforming image coordinates using a constant scaling factor such that each component of transformed centroids $(x = \varepsilon x', \; y = \; \varepsilon y')$ lies approximately in the interval $[0, \; 1]$. The image plane trajectory of each tracked object is thus represented by an ordered set of *characteristic vectors* $\mathbf{C}_t \in [0, \; 1]^2$:

$$\mathcal{C} = \{\mathbf{C}_0, \; \mathbf{C}_1, \; \ldots, \; \mathbf{C}_m\}, \qquad (3.1)$$

where $(m+1) \geq l$,

$$\mathbf{C}_t = (x(t), \; y(t)), \qquad (3.2)$$

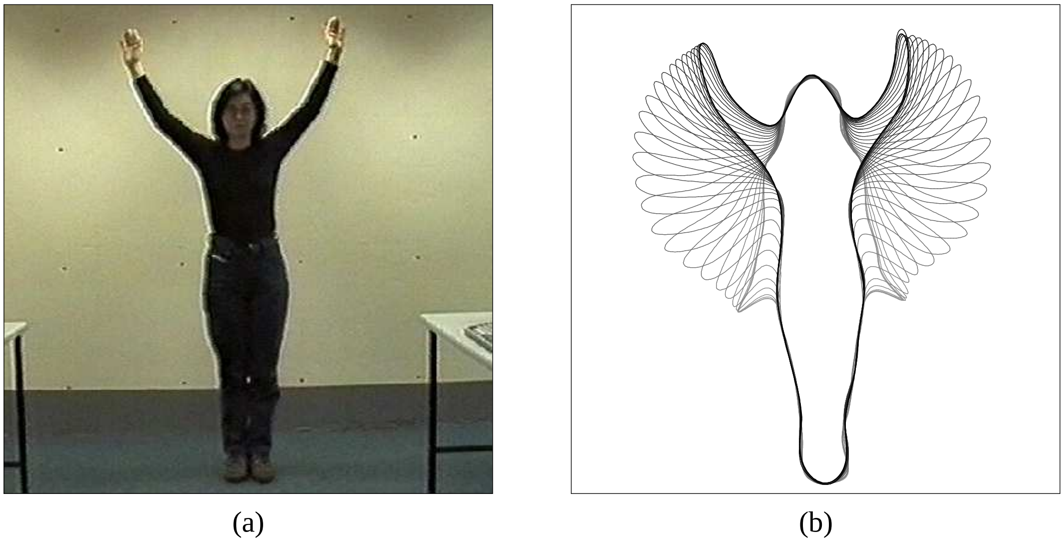
and each characteristic vector lies approximately within a unit square.

In experiments considering the location of tracked pedestrians, sequences are generally *simple* in nature with no recurring subsequences. Due to the variety of observed behaviours within a complex scene, the time needed to observe a representative sample of the behaviour population is likely to be large, perhaps as much as a number of days.

3.1.2 Object shape

For the generation of experimental shape data, the tracker described above was found to be unsuitable since it produces excessive smoothing of object silhouettes, resulting in the loss of required shape detail. This problem occurs unless the entire set of principal components is utilised, and is due to the shape model sub-space excluding these finer details. Instead, shape data is generating using the silhouette extraction method used by Baumberg and Hogg for the generation of training data [5, 7]. This system uses image differencing to locate moving objects and does not form a robust tracker, being non object-specific, sensitive to background texture and lighting fluctuations, and unsuitable for tracking occluded objects. In these experiments, individuals wearing dark clothing are tracked in uncluttered indoor scenes, resulting in the generation of data of sufficient quality.

There is a one way flow of shape data from the tracker consisting of frame by frame updates to the position within the image plane of the n control points (x'_i, y'_i), $1 \leq i \leq n$, of a closed uniform B-spline approximation to the silhouette boundary of uniquely labelled objects. For example, Figure 3.2 shows an individual performing an exercise routine, (a), and a number of smoothed shapes from a sequence representing the evolving silhouette boundary of the tracked individual, (b).


(a)

(b)

Figure 3.2: Sample shape data: (a) exercise scene, and (b) some shapes from the exercise sequence.

Spline control points are evenly spaced around the silhouette and are ordered relative to a consistent point of reference which also defines the object's position $(X = x'_1, Y = y'_1)$. The method for the location of this reference point has been enhanced from [5, 7] to allow the top of an individual's head to be more accurately located. This enhancement involves local adjustment of the reference point such that it coincides with the locally highest part of the silhouette boundary. Figure 3.3 illustrates shape representation, showing a number of sample silhouette boundaries with circles indicating the corresponding spline control points and the reference points filled.

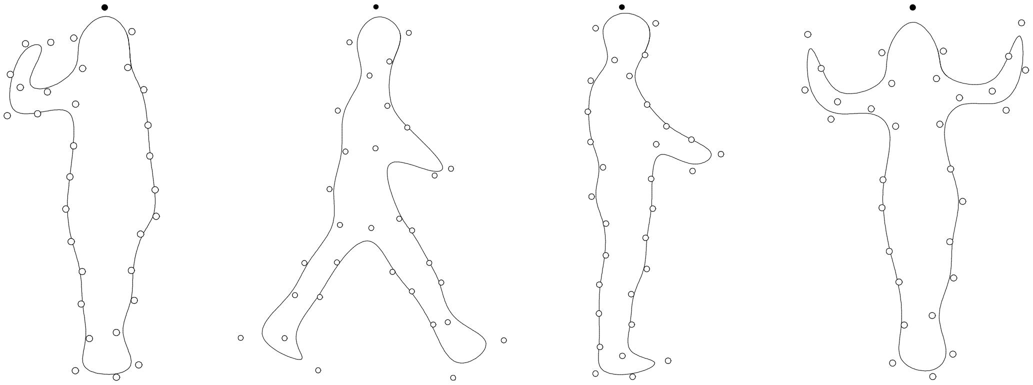


Figure 3.3: Shape representation - silhouette boundaries and corresponding spline control points.

In experiments considering the shape of tracked individuals performing exercise routines and interactions, shape space motion is typically rapid and non-linear with respect to video frame rates. For instance, when an individual being viewed head-on raises an arm from the side of the body, there is a rapid change in silhouette perimeter and shape as the arm ‘separates’ from the body, resulting in rapid non-linear motion of the spline control points. Thus the tracker needs to generate data at as high a frame rate as possible to accurately define behaviour sequences, and temporal re-sampling need only be performed to compensate for variation in the tracker’s frame rate whilst processing live video streams.

In contrast to raw location data, raw shape data sequences do not contain noise due to the tracking of spurious objects. This is due to both the careful choice of scene, and to the tracker’s rejection of small motion regions. Further pre-processing of raw data is still needed to constrain characteristic vectors to lie approximately within a unit hypercube to simplify subsequent stages of the perception process. Again this is simply a matter of transforming image coordinates using a constant scaling factor such that each component of transformed control points $(x_i = \varepsilon x'_i, \ y_i = \ \varepsilon y'_i)$ lies approximately in the interval $[0, \ 1]$. The evolving silhouette boundary of each tracked object is thus represented by an ordered set of characteristic vectors $\mathbf{C}_t \in [0, \ 1]^{2n}$:

$$\mathcal{C} = \{\mathbf{C}_0, \ \mathbf{C}_1, \ \ldots, \ \mathbf{C}_m\}, \tag{3.3}$$

where

$$\mathbf{C}_t = (x_1(t), y_1(t), x_2(t), y_2(t), \ \ldots, \ x_n(t), y_n(t)), \tag{3.4}$$

and each characteristic vector lies approximately within a unit hypercube.

In common with many natural behaviours, the shape sequences studied within this thesis are generally *complex* in nature with many recurring subsequences. Due to the wide variety of human shapes and subtle differences in these behaviours, a truly representative sample of the behaviour population can only be generated by observing many different individuals performing many examples of the behaviour. It is interesting to note that 'joint angle'-based pose representations, such as that used in Hogg's WALKER model [41, 42], are largely invariant to shape variation between individuals, and thus only variation in behaviour would be represented in the corresponding data sets.

3.1.3 State sequence approximation

The final stage of experimental data acquisition is the generation of state sequences - ordered sets of *state vectors* representing the evolution of both a behavioural characteristic and its instantaneous change. State vectors are used as the discrete unit from which behaviours are defined for a number of reasons:

- State space contains less ambiguity than characteristic space and thus state vector sequences are less complex in nature, with less recurring subsequences, than the corresponding characteristic vector sequences.
- Since state space represents instantaneous temporal changes in the measured characteristics, it is a logical starting point for the study of the longer-term temporal evolution of these characteristics.
- The instantaneous temporal information held in state vectors allows both the approximation of the time interval between two state vectors (assuming linearity and constant acceleration), and Hermite (cubic) interpolation. These capabilities are shown to be invaluable in achieving behaviour generation using a transition-based prediction scheme (see Chapter 4).

Before ordered sets of state vectors are generated, some further pre-processing of data is required. Due to inaccuracies in the tracking processes, characteristic vector sequences will be subject to high frequency noise. This noise is assumed to originate from an additive, isotropic noise process with zero mean and constant variance, and is minimised by smoothing sequences with averaging over a moving temporal window of width w. To avoid data loss from the start and end of sequences, smoothing starts after $(w-1)/2$ frames and uses the maximum width window (up to w) centred on each element of the sequence to generate a corresponding output, starting with the first element and a window of unit width.

State vectors $\mathbf{F}_t \in [0,\ 1]^{2d}$, where d is the dimensionality of characteristic vectors (i.e. $d=2$ for location data and $d=2n$ for shape data), consist of a characteristic vector $\mathbf{C}_t$ and its transformed first derivative $\dot{\mathbf{C}}_t$, approximated by the difference in characteristic vectors between successive frames:

$$\mathbf{F}_t = \left(\mathbf{C}_t,\ \lambda\dot{\mathbf{C}}_t + \mathbf{H}\right), \tag{3.5}$$

where

$$\dot{\mathbf{C}}_t = \mathbf{C}_t - \mathbf{C}_{t-1}, \tag{3.6}$$

λ is a scaling factor, and $\mathbf{H} \in \Re^d$ is a vector with all components equal to $\frac{1}{2}$.

The translation of $\dot{\mathbf{C}}_t$ by $\mathbf{H}$ ensures that each component will lie approximately in the interval $[0,\ 1]$, whilst scaling ensures that the contribution of characteristic vector components and their first derivatives are balanced when using the Euclidean distance as a measure of state vector dissimilarity. Thus λ is chosen to equalise the observed range of characteristic vector components and their first derivatives over a sample data set. This scaling of differential components can be viewed as a simplification of the use of the Mahalanobis distance (see, for example, Huang *et al.* [45] or Ripley [74]) as a dissimilarity measure. The Mahalanobis distance (or generalised distance) $D(\mathbf{x},\ \mathbf{y})$ between vectors $\mathbf{x}$ and $\mathbf{y}$ with m variables is defined as

$$D(\mathbf{x},\ \mathbf{y}) = \sqrt{(\mathbf{x}-\mathbf{y})\Sigma^{-1}(\mathbf{x}-\mathbf{y})^{\mathrm{T}}}, \tag{3.7}$$

where Σ is the $m \times m$ covariance matrix of the sample data, and thus the distance takes into consideration the variance and correlation of the variables (i.e. differences in directions with less variation are given greater weighting).

The evolving behaviour of object characteristics is thus, after pre-processing, represented by ordered data sets $\mathcal{F}_j$ of the form

$$\mathcal{F} = \{\mathbf{F}_1,\ \mathbf{F}_2,\ \ldots,\ \mathbf{F}_m\}, \tag{3.8}$$

where each state vector $\mathbf{F}_t$ lies approximately within a unit hypercube.

3.2 Learning models of probability density

The extended observation of object characteristics exhibiting interesting behaviours will define probability distributions over the state and behaviour spaces[1] of the characteristics. These distributions are likely to be complex in structure and are thus unsuitable for modelling using conventional parametric distributions. In modelling probability density functions over a feature space, the choice of modelling framework is influenced by the following aims:

- To enable model acquisition and gradual adaptation via an iterative, unsupervised learning process.
- To enable the estimation of local relative probability density, facilitating typicality assessment and attentional control.
- To enable the association of semantics with related classes of behaviour, facilitating event recognition.
- To enable the prediction or extrapolation of future behaviours and the generation of realistic sample behaviours.
- To form as concise and accurate a representation as possible.

Various methods exist for such density representation. For instance, maintaining frequency counts over a discretisation of the feature space (as used by Fernyhough *et al.* to represent path usage [29]), together with a transition-based prediction scheme, would fulfil the first four aims but would be extremely inefficient. Density representation using a mixture of situated Gaussians with parameters estimated using the EM algorithm (see Dempster *et al.* [25] or, for example, Ripley [74] or Huang *et al.* [45]) would provide an efficient solution, although the distribution of Gaussian centres is likely to be sub-optimal for semantic labelling and transition-based prediction. Instead, Vector Quantization is used to place a set of prototype vectors whose point density approximates the probability density of sample data, providing a representation in which the level of detail is proportional to probability density.

[1] Referred to generically as *feature spaces* in this section.

3.2.1 Vector Quantization

Vector Quantization (VQ) is a classical technique from signal processing, originally used for data compression, which provides a method for modelling probability density functions by the distribution of prototype vectors (see, for example, Linde *et al.* [54], Gray [35], or Gersho and Gray [33]). Most VQ algorithms, such as the *k-means* algorithm commonly used in cluster analysis (see, for example, Schalkoff [79], Haykin [37], or Ripley [74]), are unsuitable as they operate in a *batch* training mode, only updating prototype positions after each observation of the entire training set. Instead, an iterative algorithm based on the Competitive Learning paradigm (see, for example, Rumelhart and Zipser [78] and Kohonen [51]) is used. The algorithm places a set of k prototypes $\mathbf{c}_i \in [0,\ 1]^d$, referred to as the *codebook,* over N iterations:

1. Randomly place the k prototypes within the unit hypercube $[0,\ 1]^d$.

2. Select $\mathbf{z}(t)$, the current training vector, randomly from the distribution to be modelled[2].

3. Find the prototype $\mathbf{c}_j(t)$ which is nearest to the current training vector $\mathbf{z}(t)$ by the Euclidean metric:

$$|\mathbf{z}(t) - \mathbf{c}_j(t)| = \min_i \{|\mathbf{z}(t) - \mathbf{c}_i(t)|\}. \tag{3.9}$$

4. Update prototypes as follows:

$$\mathbf{c}_i(t+1) = \begin{cases} \mathbf{c}_i(t) + \alpha(t)\,[\mathbf{z}(t) - \mathbf{c}_i(t)] & \text{if} \quad i = j \\ \mathbf{c}_i(t) & \text{otherwise,} \end{cases} \tag{3.10}$$

where $\alpha(t)$ is a monotonically non-increasing gain coefficient,

$$\alpha(t) = \begin{cases} 1 - 0.99\left(\frac{2t}{N}\right) & \text{if} \quad 0 \le t < \frac{N}{2} \\ 0.01 & \text{if} \quad t \ge \frac{N}{2}, \end{cases} \tag{3.11}$$

referred to as the *cooling schedule* of the learning process.

5. Repeat steps 2-4 for N iterations.

[2] Sequential selection of vectors from ordered data sets was occasionally found to be detrimental to the learning process due to the tendency of a sequence to ‘drag’ a single prototype. Instead, random selection (without replacement) from a small buffer of sequentially selected vectors is used.

Kohonen [51] shows that such an algorithm is a gradient descent procedure for the approximation of an optimal VQ minimising the expected squared reconstruction error

$$E = \int |\mathbf{z} - \mathbf{c}_j|^2 p(\mathbf{z})\, d\mathbf{z}, \tag{3.12}$$

where $p(\mathbf{z})$ is the continuous probability density function underlying the training data and $d\mathbf{z}$ is the volume differential in feature space. Kohonen also cites proof that, for such an optimal VQ, density matching will obey the power law

$$P(\mathbf{c}) \propto p(\mathbf{z})^{\frac{d}{(d+2)}}, \tag{3.13}$$

where $P(\mathbf{c})$ is the prototype's point density function. Thus, when $d \gg 2$, the point density of prototypes will, after learning, approximate the probability density of training data, and each prototype will represent (in a *nearest-neighbour* sense) an approximately equal number of training vectors.

The cooling schedule detailed in step 4 is chosen to give a large initial gain which decreases gradually over the first half of the learning process, allowing prototypes to move rapidly into roughly the desired distribution. During the remainder of the learning process, the relatively small fixed gain allows fine tuning of the prototype distribution as the system reaches equilibrium.

The remaining model parameters must be determined experimentally. The number of iterations required to achieve a reasonable distribution is dependent on the number of prototypes, the dimensionality of the feature space, and the attributes of the distribution being estimated, and, in our experiments, is typically in the order of millions of iterations. In cluster analysis, a 'natural number' of prototypes can be determined by observing the reconstruction error for increasing numbers of prototypes. In the estimation of dispersed distributions, such an analysis is only useful in determining an absolute minimum number of prototypes; the number chosen is then essentially arbitrary, more prototypes giving a more detailed representation.

3.2.2 Improving prototype distribution

Before the VQ algorithm described can be used to learn probability density representations, there are three limitations which must be addressed in order to generate optimal representations:

1. Dynamically changing object characteristics sweep out continuous paths in the corresponding feature spaces. These paths are sampled at regular time instants to generate the ordered

sets of experimental data from which the probability density over each space is estimated. When the speed at which a particular path is swept out is low, the sampled vectors are densely distributed, and when it is high, the vectors are sparsely distributed. This will result in a higher probability density in areas where the rate of movement along a particular path is low.

2. The final distribution of prototypes is extremely sensitive to their initial placement within the feature space. For instance, prototypes can be 'stranded' in areas where they will never take part in the competition, resulting in a sub-optimal distribution. This is a particular problem in sparse distributions such as those to be modelled here.

3. From Equation 3.13 it can be seen that, for low dimensional feature spaces, the VQ approximation to the probability density of training data is highly non-linear (i.e. the density matching is poor). In such low dimensional feature spaces, areas of high probability density will be under-represented and areas of low probability density over-represented.

The first of these limitations can be avoided by re-sampling the piecewise linear interpolant of the experimental data such that adjacent vectors in the resulting ordered set have a constant separation Δ, and are thus evenly distributed along the path. Within this scheme, each new vector sample is generated using geometry to find the point of intersection between a hypersphere of radius Δ centred on the last sample and the next piecewise linear interpolant to cross this boundary. The value of Δ is chosen to be approximately equal to the average distance between vectors in the original experimental data (or less for highly non-linear data). Unless otherwise stated, it is assumed hereafter that all data sets are modified in this manner prior to training. A solution to the remaining two limitations is described below, resulting in a robust VQ algorithm which is insensitive to the initial placement of prototypes and in which density matching is approximately correct, regardless of dimensionality.

3.2.2.1 Adding prototype sensitivity

A number of solutions to the problem of sensitivity to initial prototype placement have been discussed in the literature. The simplest methods involve initialisation of prototype positions from either the first *k* training vectors, or, more robustly, from *k* vectors chosen randomly from the training set. Although generally successful, such approaches will not allow prototypes to relocate during adaptation over time-varying distributions, and do nothing to improve density matching. Rumelhart and Zipser [78] propose a better solution called *leaky learning*, in which prototypes losing the competition also move towards the current training vector, but by a much smaller amount. This results in stranded prototypes drifting towards the mean of the distribution. For sparse distributions, however, this mechanism is inadequate since the mean of the distribution may be unpopulated.

Instead, inspired by the work of Bienenstock *et al.* [8] on adaptive sensitivity to stimuli in neurons, an algorithm is developed which extends VQ to incorporate a prototype sensitivity mechanism. By allowing prototypes to automatically vary their sensitivity to input features, prototypes which are winning too often can decrease their sensitivity and exclude themselves from the competition. In this way stranded prototypes become increasingly sensitive to input features until they too begin to compete, whilst the mechanism also allows exact density matching to be enforced. To distinguish this enhanced algorithm from the standard iterative VQ algorithm, it is referred to as *Altruistic Vector Quantization* (AVQ). We became aware, very recently, of a similar extension - the *Conscience Algorithm* proposed by DeSieno [26].

Sensitivity to input features is realised by associating a *sensitivity value* $S_i(t)$ with each prototype $\mathbf{c}_i$, and subtracting this value from the Euclidean distance when finding the nearest prototype in step 3 of the standard VQ algorithm. In this way a prototype with positive sensitivity is more likely and a prototype with negative sensitivity is less likely to win the competition. Thus Equation 3.9 now becomes

$$|\mathbf{z}(t) - \mathbf{c}_j(t)| - S_j(t) = \min_i\{|\mathbf{z}(t) - \mathbf{c}_i(t)| - S_i(t)\}, \tag{3.14}$$

where $S_i(0) = 0$ and sensitivity values are updated on each iteration using

$$S_i(t+1) = \zeta S_i(t) + A_i, \tag{3.15}$$

where ζ is a *damping coefficient* defined as

$$\zeta = 1 - \frac{\beta}{(k-1)\sqrt{d}}, \tag{3.16}$$

and A_i introduces sensitivity adjustments defined by

$$A_i = \begin{cases} -\beta & \text{if } i = j \\ \frac{\beta}{k-1} & \text{otherwise,} \end{cases} \tag{3.17}$$

where β is a constant in the interval $(0,\ 1)$ specifying the magnitude of adjustments. The value of β should be small relative to distances within the feature space, but large enough to enable stranded prototypes to ‘escape’ early in the learning process.

The form of the sensitivity adjustments in Equation 3.17 ensures that, for correctly distributed prototypes, the mean adjustment will be zero, thus enforcing exact density matching. The coefficient ζ is required to damp dynamically shifting imbalances in sensitivity which are caused initially by stranded prototypes but which tend to persist throughout learning, leading to excessive motion of prototypes. The form of Equation 3.16 ensures that sensitivity values will tend to zero (since $0 < \zeta < 1$), and that $S_i(t+1) \leq \sqrt{d}$, the largest possible separation within a d-dimensional unit hypercube (since $\zeta\sqrt{d} = \sqrt{d} - \frac{\beta}{k-1}$).

3.2.2.2 Density matching - scalar case

Whilst extensive experiments on real data show the AVQ algorithm to be highly successful in removing sensitivity to initial prototype placement (see, for example, Sections 3.3 and 3.4) and in allowing prototype relocation during adaptation over time-varying distributions, the effect of this approach on density matching needs to be more formally quantified.

In a simple experiment (adapted from those performed by Ritter [75]), asymptotic ($k \rightarrow \infty$) density matching has been demonstrated for both the standard VQ and AVQ algorithms, using a series of Monte Carlo simulations on scalar data sampled from the simple ‘ramped’ distribution $p(x) = 2x$, where $0 \leq x \leq 1$. The scalar case was used since, as indicated by Ritter, the standard VQ algorithm is very slow to reach equilibrium, particularly as the dimensionality of feature space increases.

The following experimental procedure was used for each simulation:

1. Divide the scalar feature space x into 10 histogram bins covering intervals $[i\Delta,\ (i+1)\Delta]$, where $0 \leq i < 10$ and $\Delta = 0.1$.

2. Initialise the $k = 100$ scalar prototypes by sampling from a uniform distribution over the unit interval $[0,\ 1]$.

3. Using a static gain coefficient $\alpha(t) = 0.01$, and a value of $\beta = 0.001$ for AVQ, perform

50,000,000 iterations of VQ/AVQ to allow the system to reach equilibrium, sampling training vectors from the distribution $p(x) = 2x$, where $0 \leq x \leq 1$.

4. Assuming equilibrium has been reached, perform a further 50,000,000 iterations of VQ/AVQ, taking a total of 50,000 'snapshots' of the system at 1,000 iteration intervals.

5. Estimate the probability $Q(i)$ of a prototype lying in each bin by summing the number of prototypes within each bin over the 50,000 snapshots and dividing these totals by 5,000,000.

For each algorithm, 10 independent simulations were performed and the mean $\overline{Q}(i)$ and standard deviation $\sigma(i)$ of the probability of a prototype lying in each bin were calculated. The results of this experiment are summarised in Table 3.1, including theoretical probabilities for exact density matching, $P(i)$, and density matching obeying the power law given by Equation 3.13 for the $d = 1$ (scalar) case, $D(i)$. Experimental results are also displayed graphically in Figures 3.4 and 3.5 where the error bars represent $\pm 3\sigma(i)$. The theoretical probabilities were calculated by integrating density functions over each histogram bin, i.e.

$$P(i) = \int_{i\Delta}^{(i+1)\Delta} p(x)\,dx, \tag{3.18}$$

and

$$D(i) = \int_{i\Delta}^{(i+1)\Delta} \frac{1}{a} p(x)^{\frac{1}{3}}\,dx, \tag{3.19}$$

where

$$a = \int_{0}^{1} p(x)^{\frac{1}{3}}\,dx. \tag{3.20}$$

i	$\overline{Q}(i)_{\mathrm{AVQ}} \pm 3\sigma(i)_{\mathrm{AVQ}}$	$P(i)$	$\overline{Q}(i)_{\mathrm{VQ}} \pm 3\sigma(i)_{\mathrm{VQ}}$	$D(i)$
0	0.010000 ± 0.000000	0.01	0.050636 ± 0.003042	0.046416
1	0.030000 ± 0.000000	0.03	0.075151 ± 0.010128	0.070545
2	0.050000 ± 0.000002	0.05	0.089974 ± 0.002815	0.083869
3	0.070001 ± 0.000008	0.07	0.096990 ± 0.005214	0.093893
4	0.090004 ± 0.000015	0.09	0.104243 ± 0.002265	0.102128
5	0.110003 ± 0.000019	0.11	0.108463 ± 0.002109	0.109209
6	0.130000 ± 0.000018	0.13	0.112034 ± 0.003484	0.115473
7	0.149996 ± 0.000033	0.15	0.116940 ± 0.005545	0.121121
8	0.170001 ± 0.000026	0.17	0.120430 ± 0.003174	0.126286
9	0.189992 ± 0.000022	0.19	0.125136 ± 0.008817	0.131060

Table 3.1: Summary of density matching results.

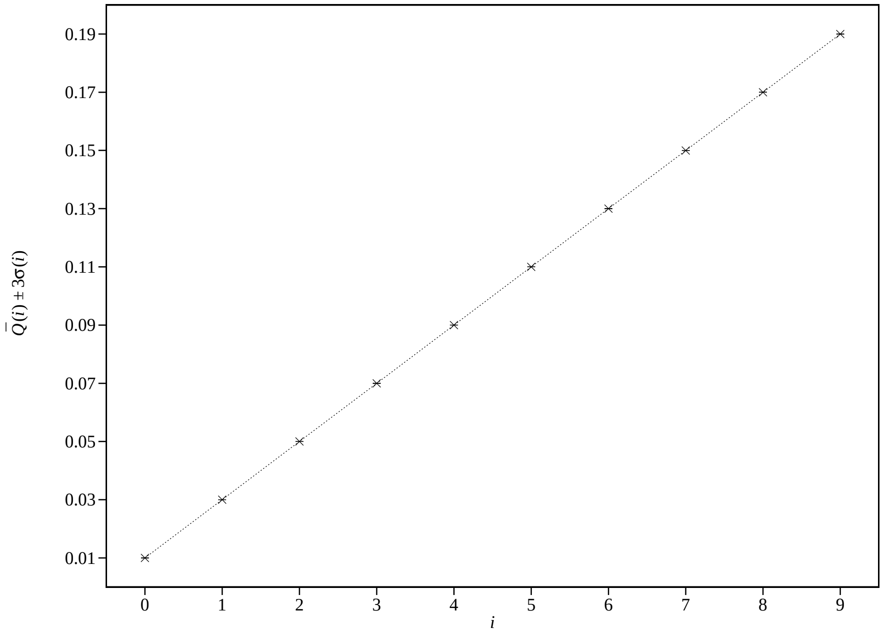


Figure 3.4: Density matching results for AVQ.

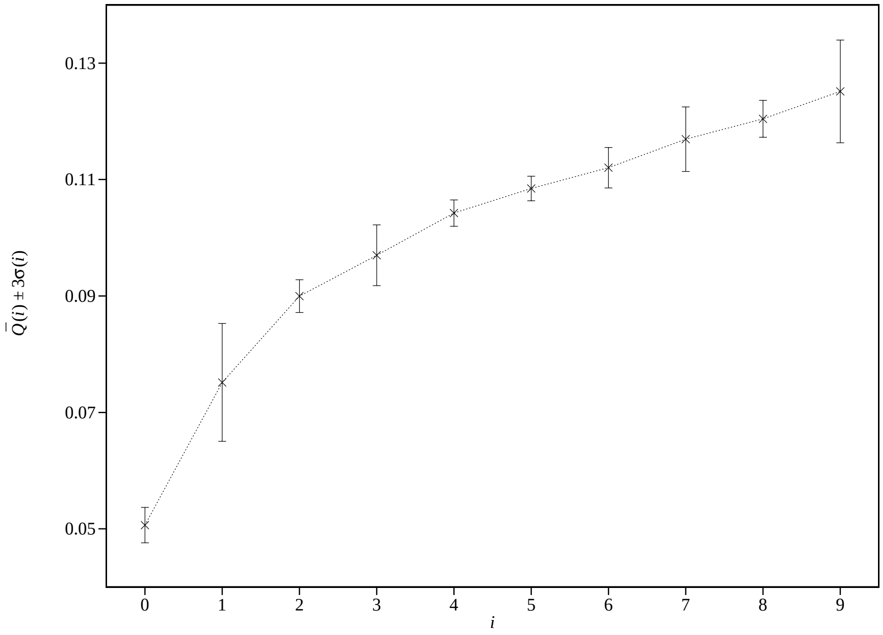


Figure 3.5: Density matching results for VQ.

Inspection of both the results listed in Table 3.1 and the shape of the graphs in Figures 3.4 and 3.5 clearly shows that, whilst the performance of the standard VQ algorithm is consistent with the distortion predicted by Equation 3.13, the addition of the sensitivity mechanism in the AVQ algorithm results in an almost exact density matching, at least for the scalar case. Experimental results in the remainder of the thesis for AVQ over higher dimensional distributions provide visually compelling evidence that density matching may be invariant to dimensionality.

The results of this experiment also identify considerably smaller error margins for the AVQ algorithm, suggesting that equilibrium is reached much faster than for the standard VQ algorithm. This is confirmed by experimental observations and hence, in later experiments with real data, the number of iterations used is approximately one order of magnitude less than used here.

3.3 Learning state models

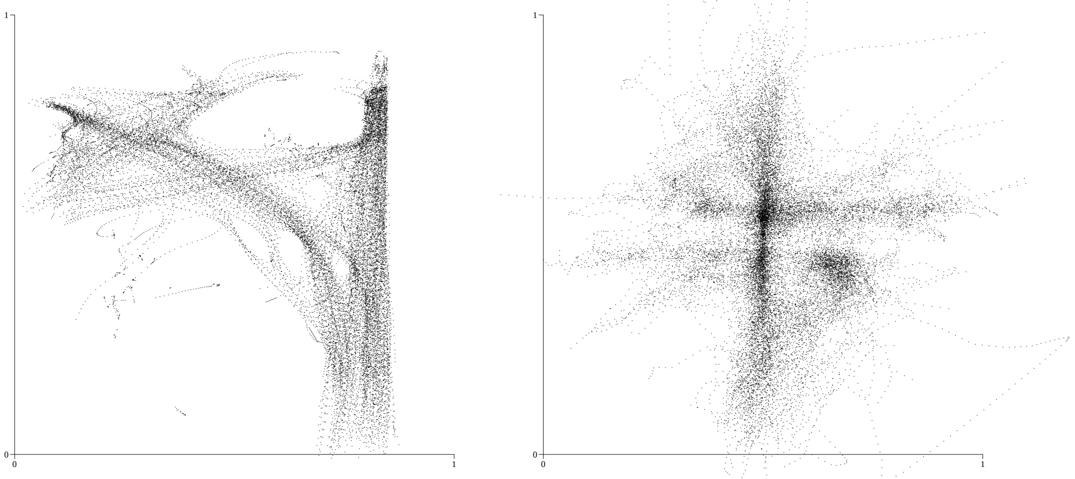
Using the robust Altruistic Vector Quantization (AVQ) algorithm developed in Section 3.2, detailed models of state space probability density can be learnt in an unsupervised manner from the extended observation of vectors from state training sets $\mathcal{F}_j$. The resulting state models comprise sets of u state prototypes $\bar{\alpha}_i$:

$$\mathcal{A} = \{\bar{\alpha}_1,\ \bar{\alpha}_2,\ \ldots,\ \bar{\alpha}_u\}\,. \tag{3.21}$$

In this section, experimental results are presented to demonstrate the acquisition of such models for the two distinct object characteristics detailed in Section 3.1 - object location within the image plane and object silhouette shape. In the following section, the models presented here are used as the basis for a spatio-temporal trajectory representation defining a behaviour space within which characteristic object behaviour can be modelled.

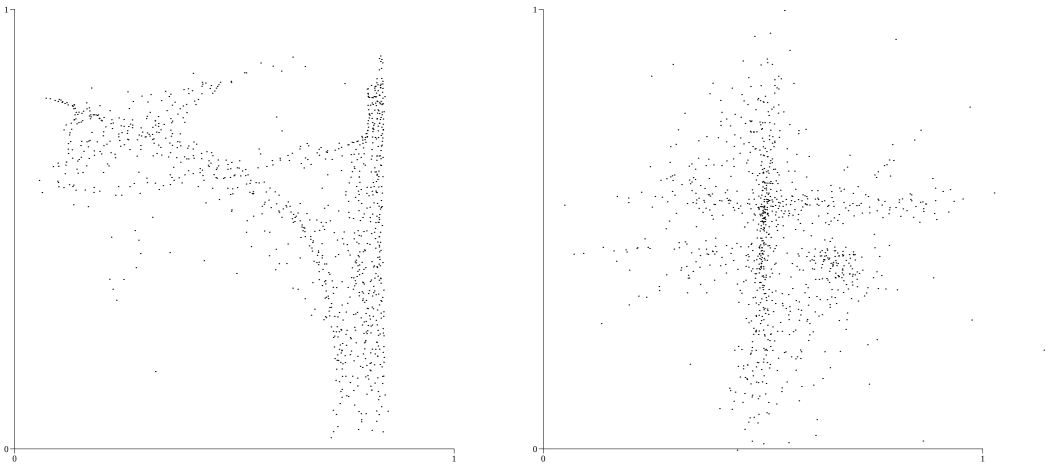
3.3.1 Experimental results - object location

State training sets $\mathcal{F}_j^{\mathrm{loc}}$ were generated from the 622 smoothed, sub-sampled pedestrian trajectories illustrated in Figure 3.1(b). Sub-sampling of the 2-dimensional characteristic vectors $\mathbf{C}_t = (x(t),\, y(t))$ was performed at $0.5s$ intervals and high frequency noise was minimised by smoothing over a moving window of width $w = 5$. To minimise noise due to the tracking of spurious objects, trajectories existing for less than $l = 50$ frames were rejected. 4-dimensional state vectors $\mathbf{F}_t$ were generated using a scaling factor $\lambda = 10$ to scale differential components, and ordered data sets were further re-sampled to improve density representation using a constant separation $\Delta = 0.02$. After pre-processing, training sets $\mathcal{F}_j^{\mathrm{loc}}$ comprised a total of 23,878 state vectors lying approximately within a unit hypercube. Figure 3.6 shows scatter plots of this training data projected onto both the $(x,\ y)$ plane, (a), and the $(\lambda\dot{x}+\frac{1}{2},\ \lambda\dot{y}+\frac{1}{2})$ plane, (b).



(a) (b)

Figure 3.6: State vector distribution - object location: (a) projection onto the position plane, and (b) projection onto the first derivative plane.



(a) (b)

Figure 3.7: State prototype distribution - object location: (a) projection onto the position plane, and (b) projection onto the first derivative plane.

A set $\mathcal{A}^{\mathrm{loc}}$ of 1,000 state prototypes was learnt from 2,000,000 iterations of AVQ over state vectors from the training sets $\mathcal{F}_j^{\mathrm{loc}}$. A constant $\beta = 0.01$ was used for sensitivity adjustments in the AVQ algorithm together with the two-stage cooling schedule described in Section 3.2.1. Figure 3.7 shows scatter plots of the resulting state prototypes projected onto both the $(x,\ y)$ plane, (a), and the $(\lambda\dot{x}+\frac{1}{2},\ \lambda\dot{y}+\frac{1}{2})$ plane, (b). Comparison with the scatter plots of training data clearly shows the results to be plausible and suggests that reasonable density matching is achieved.

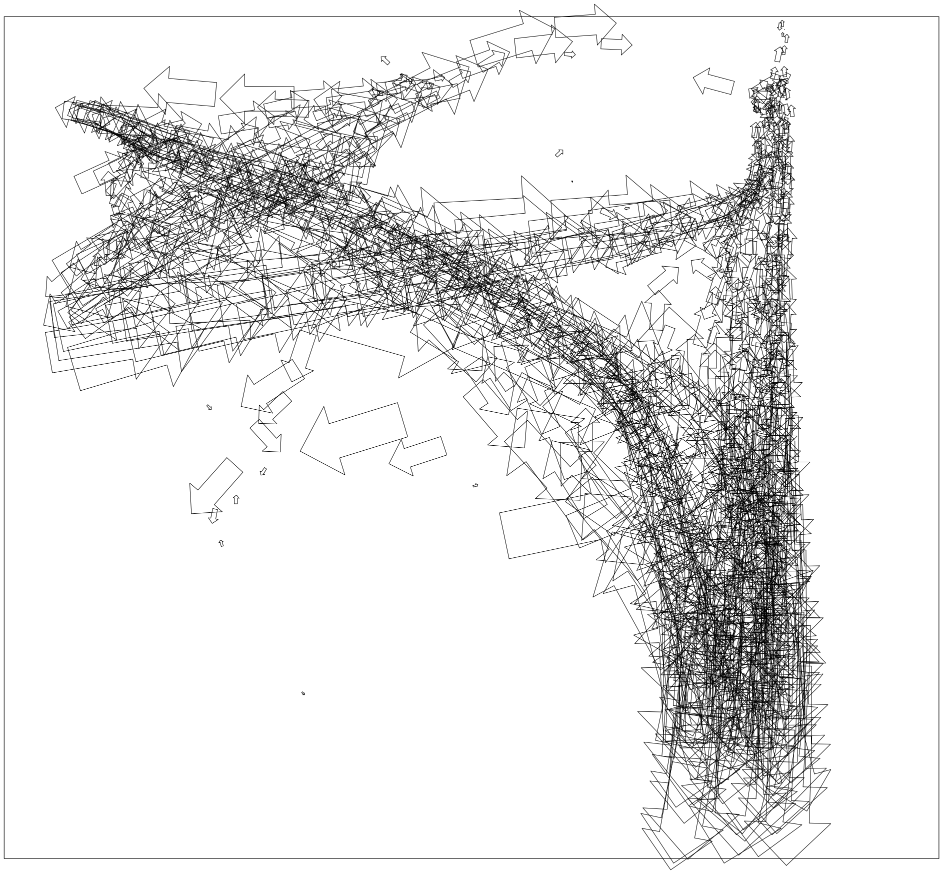


Figure 3.8: Learnt state prototypes - object location.

In Figure 3.8, each of the 1,000 state prototypes is illustrated by a single arrow, the position of which represents the prototype's $(x,\, y)$ components whilst the size and direction represent the prototype's $(\lambda\dot{x}+\frac{1}{2},\, \lambda\dot{y}+\frac{1}{2})$ components scaled a by factor of $\frac{1}{20}$. It is clear from this representation that prototypes lie in the desired areas of the state space and that stranded prototypes have successfully entered the competition.

Although the results presented in Section 3.2.2.2 suggest that the AVQ algorithm is capable of producing an almost exact density matching as the number of prototypes tends to infinity, for highly structured distributions within high dimensional spaces, the accuracy of density matching can be expected to decrease as the ratio of training vectors to prototypes increases. Perhaps the simplest method of assessing the accuracy of density matching within a feature space is to count the number of training vectors which are closest to each prototype by the Euclidean metric and to plot a frequency histogram of the results. For an exact density match, each prototype must represent an equal amount of probability, and thus each prototype will be closest to an equal number $\frac{N}{k}$ of training vectors, where N is the size of the training data set and k is the number of prototypes. As the accuracy of density matching decreases, it is reasonable to expect the distribution to become approximately normal with a mean of $\frac{N}{k}$ and an increasing standard deviation.

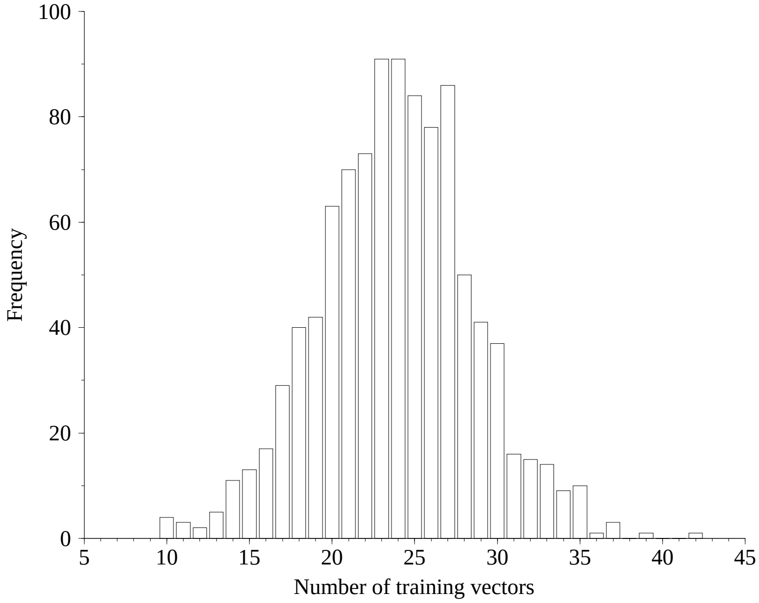
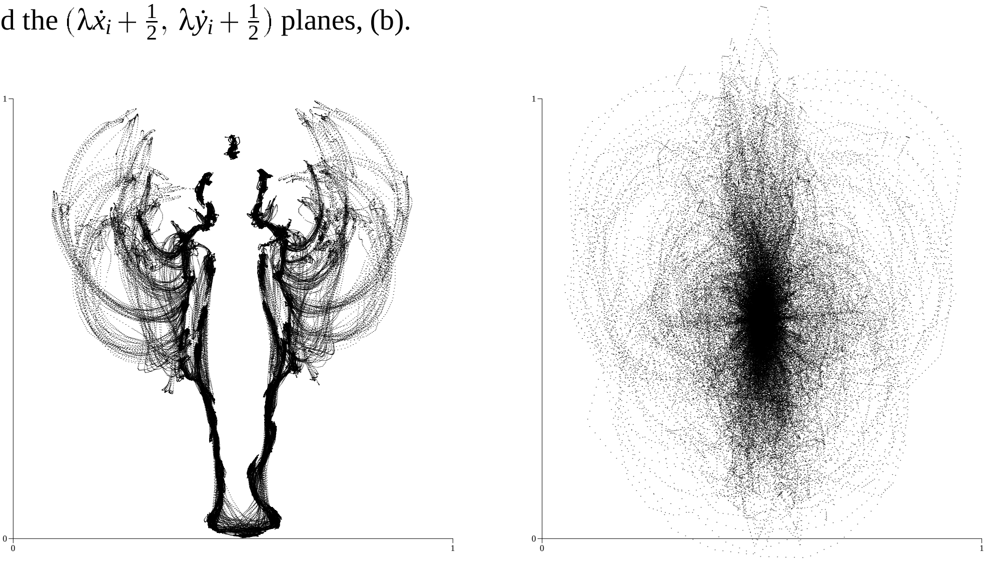


Figure 3.9: Frequency histogram illustrating state prototype density matching - object location.

Figure 3.9 shows such a frequency histogram for the 1,000 state prototypes and 23,878 training vectors used in this experiment. The mean of this approximately normal distribution is between 23 and 24, which is consistent with the expected value of 23.878, whilst the width of the distribution suggests some inaccuracy in density matching.

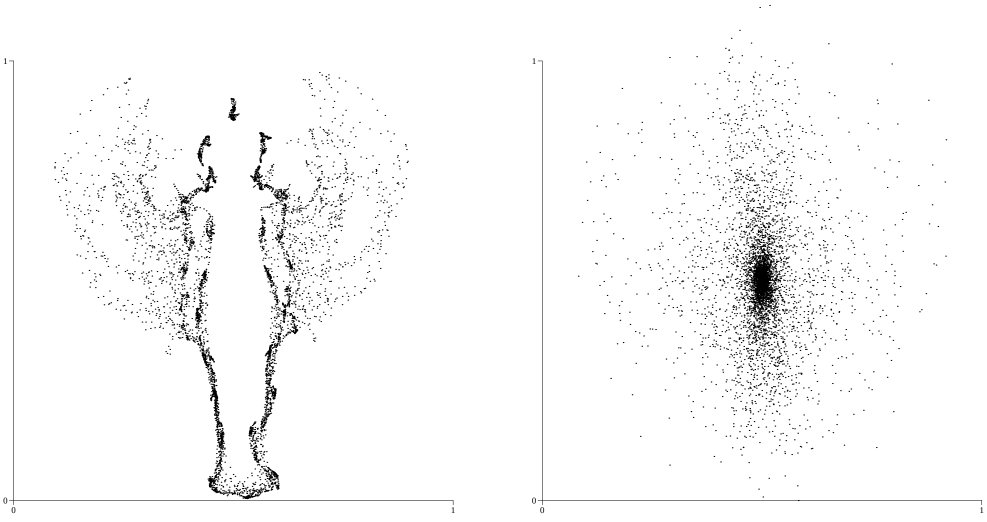
3.3.2 Experimental results - object shape

The state training set $\mathcal{F}^{\text{shape}}$ was generated from the single smoothed, sub-sampled shape sequence partially illustrated in Figure 3.2(b). This exercise routine comprises four main exercises, each of which is repeated four times and then followed by a further four repetitions of a 'sub-exercise'. Sub-sampling of the 64-dimensional characteristic vectors $\mathbf{C}_t = (x_1(t),\ y_1(t),\ x_2(t),\ y_2(t),\ \ldots,\ x_{32}(t),\ y_{32}(t))$ (describing 32 control point B-splines) was performed at 0.02s intervals and high frequency noise was minimised by smoothing vectors over a moving window of width $w = 5$. 128-dimensional state vectors $\mathbf{F}_t$ were generated using a scaling factor $\lambda = 10$ to scale differential components, and the ordered data set was further re-sampled to improve density representation using a constant separation $\Delta = 0.05$. After pre-processing, the training set $\mathcal{F}^{\text{shape}}$ comprised a total of 5,933 state vectors lying approximately within a unit hypercube. Figure 3.10 shows scatter plots of this training data projected onto both the $(x_i,\ y_i)$ planes, (a), and the $(\lambda\dot{x}_i + \frac{1}{2},\ \lambda\dot{y}_i + \frac{1}{2})$ planes, (b).



(a) (b)

Figure 3.10: State vector distribution - object shape: (a) projection onto the position planes, and (b) projection onto the first derivative planes.



(a) (b)

Figure 3.11: State prototype distribution - object shape: (a) projection onto the position planes, and (b) projection onto the first derivative planes.

A set $\mathcal{A}^{\text{shape}}$ of 200 state prototypes was learnt from 2,000,000 iterations of AVQ over state vectors from the training set $\mathcal{F}^{\text{shape}}$. A constant $\beta = 0.01$ was used for sensitivity adjustments in the AVQ algorithm together with the two-stage cooling schedule described in Section 3.2.1. Figure 3.11 shows scatter plots of the resulting state prototypes projected onto both the (x_i, y_i) planes, (a), and the $(\lambda\dot{x}_i + \frac{1}{2}, \lambda\dot{y}_i + \frac{1}{2})$ planes, (b). Comparison with the scatter plots of training data clearly shows the results to be plausible and suggests that reasonable density matching is achieved.

In Figure 3.13, each of the 200 state prototypes is illustrated by a pair of overlapping silhouettes, the upper spline representing the prototype's (x_i, y_i) components whilst the lower spline has been generated by subtracting the prototype's $(\dot{x}_i, \dot{y}_i)$ values from the corresponding (x_i, y_i) components. It is clear from this representation that prototypes lie in the desired areas of the state space and that stranded prototypes have successfully entered the competition.

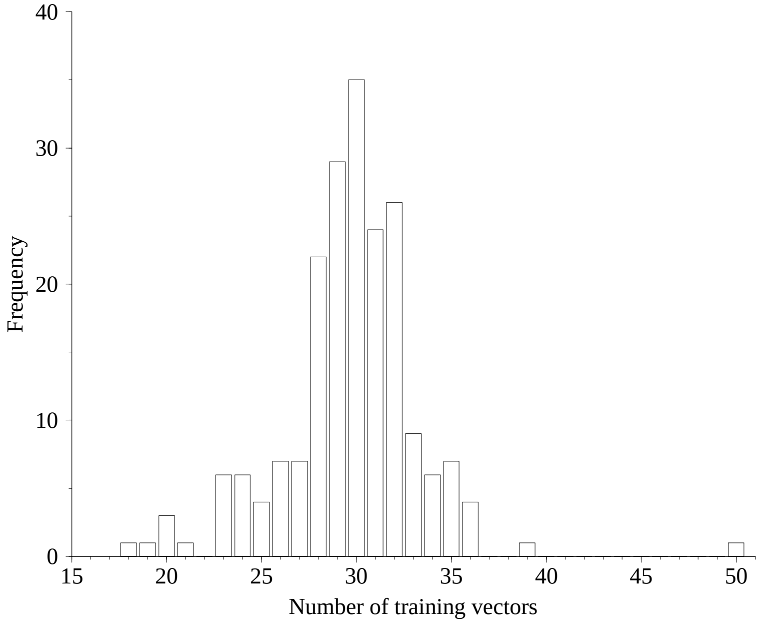


Figure 3.12: Frequency histogram illustrating state prototype density matching - object shape.

Finally, Figure 3.12 shows a frequency histogram illustrating density matching for the 200 prototypes and 5,933 training vectors used in this experiment. The mean of this approximately normal distribution is around 30, which is consistent with the expected value of 29.665, whilst the width of the distribution suggests some inaccuracy in density matching.

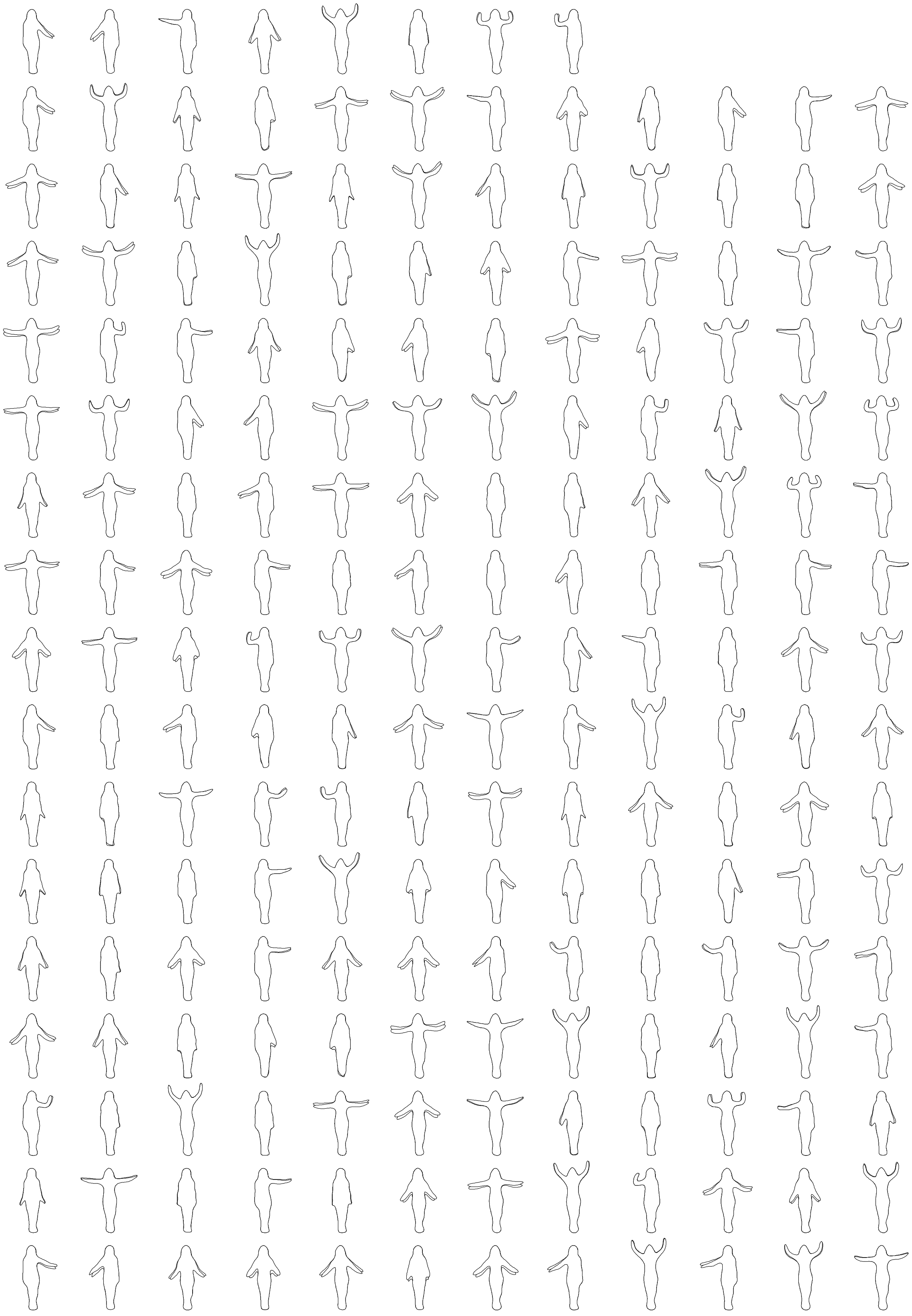


Figure 3.13: Learnt state prototypes - object shape.

3.4 Learning behaviour models

To model probability density over the behaviour space of an object characteristic, a behaviour space representation must be formed which encodes spatio-temporal trajectories of different lengths, and in which similar trajectories are close together (and *vice versa*). In this section, state trajectory representation is achieved using a novel temporal pattern formation strategy which encodes the evolving proximity of state vectors to the corresponding set of state prototypes, using a memory mechanism to maintain a history of close proximities. This strategy results in a representation which is of fixed size, which encodes trajectories of different lengths whilst maintaining a similar level of detail, and which ensures that the separation of any two points in the behaviour space is a measure of the dissimilarity of the trajectories they represent. In the remainder of this section, experimental results are presented to demonstrate the acquisition of behaviour models for both object location and object shape.

3.4.1 Temporal pattern formation

A common approach to sequence representation within the neural network literature is the use of neurons such as the Leaky Integrators of Reiss and Taylor [69] or the neurons of Wang and Arbib [88]. Such neurons implement a simple memory mechanism by allowing activation to decay slowly over a period of time. This leaky characteristic is present in biological neurons where electrical potential on the neuron's surface decays according to a time constant. Typically, when learning simple sequences of discrete tokens, a single neuron is associated with each token, and each neuron's activation gives a measure of the elapsed time since the corresponding token was last seen. In this way, the activation of the entire set of neurons at any given time instant forms an encoding of the token sequence previously presented.

Whilst such an approach could be used to form a representation of state trajectories, using state prototype set $\mathcal{A}$ to define a discrete token alphabet, the representation would not possess the sense of similarity required for a behaviour space, since the representation fails to capture any sense of token similarity - two similar state trajectories could give rise to entirely different token sequences and would thus not lie near one another within the behaviour space. Such discontinuities within the behaviour space would negate the use of the Euclidean distance as a dissimilarity measure resulting in an invalid probability density representation.

Instead, temporal pattern formation is achieved by considering the proximity of successive state vectors from an ordered set $\mathcal{F}$ to the corresponding state prototype set $\mathcal{A}$, where unmodified data sets are used to preserve the constant time interval between successive state vectors. The proximity $p_i(t)$ of a state vector $\mathbf{F}_t$ to a state prototype $\bar{\alpha}_i$ decreases linearly from one to zero as the distance between them increases from zero to the maximum observed separation within the unit hypercube state space:

$$p_i(t) = 1 - \rho\left(\frac{|\mathbf{F}_t - \bar{\alpha}_i|}{\sqrt{d}}\right), \tag{3.22}$$

where d is the dimensionality of the state space and ρ is a scaling factor chosen such that $\frac{\sqrt{d}}{\rho}$ is approximately equal to the maximum observed separation within state space.

If the proximity of successive state vectors to a particular state prototype is observed over a period of time, proximity maxima will occur at instants of closest proximity between the state trajectory and the prototype, whilst the value of each maximum will encode the similarity between the prototype and the state trajectory at these time instants. Applying a conditional decay operator to these continuous valued proximity sequences allows a *trace* of these maxima to be retained in a similar manner to the leaky neuron memory mechanism used in learning discrete token sequences:

$$z_i(t) = \begin{cases} p_i(t) & \text{if } p_i(t) > \gamma z_i(t-1) \\ \gamma z_i(t-1) & \text{otherwise,} \end{cases} \tag{3.23}$$

where γ is a coefficient in the interval $(0,\ 1)$ which governs the rate of decay and thus the *memory* of the representation. $z_i(t)$ will mimic $p_i(t)$ unless proximity values decrease at a rate which is greater than the rate of decay due to γ. Thus, given a slow decay rate (high value of γ), the value of $z_i(t)$ will retain a trace of proximity maxima.

Figure 3.14 illustrates the results of applying a conditional decay operator with $\gamma = 0.99$ to proximity sequences generated using a 500 frame sample from the experimental shape data set, a scaling factor of $\rho = 3.5$, and four of the state prototypes illustrated in Figure 3.13.

Although the value of $z_i(t)$ cannot be employed as a measure of the elapsed time since the last proximity maximum (resulting in a non-reconstructive representation), the evolving pattern formed over the entire set of prototypes does give a trajectory encoding with the properties outlined in the introduction to this section. Thus, at each time instant, a behaviour vector $\mathbf{G}_t \in [0,\ 1]^u$ is formed

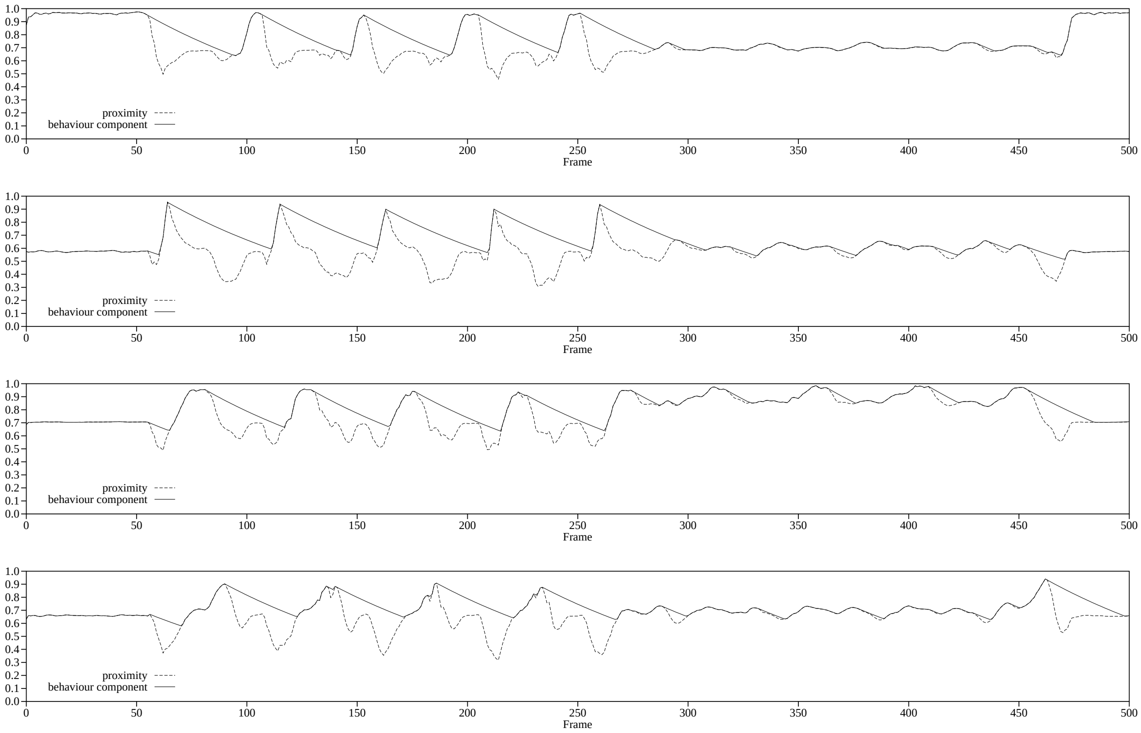


Figure 3.14: Conditional decay operator applied to sample proximity data.

from the set of $z_i(t)$ values associated with the state prototypes, where u is the cardinality of the state prototype set and $z_i(0) = 0$:

$$\mathbf{G}_t = (z_1(t), z_2(t), \ldots, z_u(t)). \tag{3.24}$$

Figure 3.15 illustrates a sample behaviour vector generated from one of the experimental pedestrian trajectories and the state prototypes illustrated in Figure 3.8, using a decay coefficient $\gamma = 0.999$ and a scaling factor $\rho = 1.4$. In this representation, each behaviour vector component is illustrated by a coloured arrow. The arrow indicates which of the state prototypes the component corresponds to whilst the colour (and layering) represents the value of the component, red representing 1.0. In this illustration, the behaviour represented by the trace of proximity maxima is immediately apparent.

Since similar prototypes will give rise to similar behaviour vector components, representations of similar state trajectories will lie close to one another within the behaviour space and *vice versa*. The relative value of maxima (and decayed maxima thereafter) associated with state prototypes surrounding a point on the state trajectory also allows the representation to partially encode the position of the point relative to the prototypes, resulting in a representation which is sensitive to

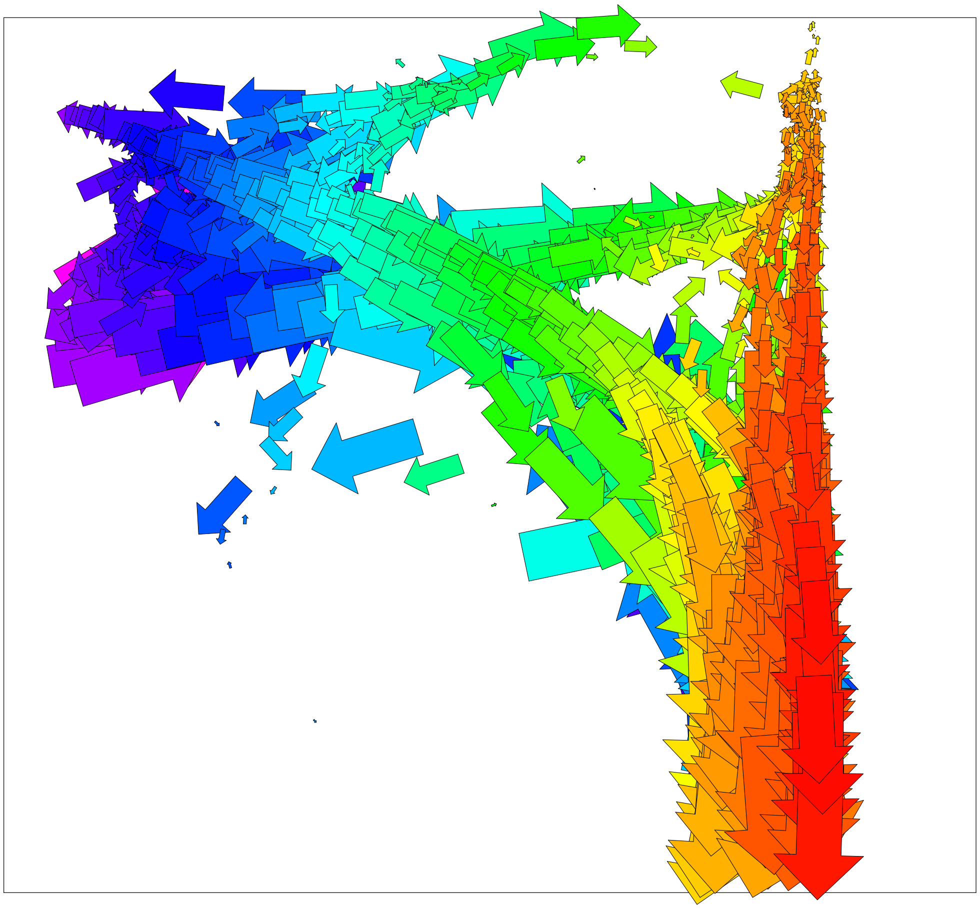


Figure 3.15: Sample behaviour vector - object location.

minor differences between trajectories.

In simple sequences where the state trajectory passes each state prototype no more than once, any length of behaviour can be represented up to a maximum defined by the number of prototypes and the rate of decay due to γ. In more complex sequences, it is often necessary to use a relatively fast decay rate (and thus reduced memory) to prevent the saturation of behaviour vector components which correspond to recurring state prototypes.

For slow decay rates relative to a particular sequence length, decay is approximately linear, resulting in an equal discriminatory ability in both shorter and longer sequences. Thus, in such cases, the representation can be considered to maintain a similar level of detail, independent of sequence length. However, for faster decay rates (or longer sequences), the ability to discriminate the oldest parts of trajectories gradually diminishes or is entirely lost.

3.4.2 Method

Having developed a behaviour space representation, detailed models of probability density over object behaviour space can be learnt in a similar manner to object state models. For each unmodified training set $\mathcal{F}$, an ordered set of behaviour vectors $\mathbf{G}_t \in [0,\ 1]^u$ is generated from the corresponding set $\mathcal{A}$ of u state prototypes and the m state vectors $\mathbf{F}_t$:

$$\mathcal{G} = \left\{\mathbf{G}_1,\ \mathbf{G}_2,\ \ldots,\ \mathbf{G}_m\right\}. \tag{3.25}$$

Thus, at each time instant, a behaviour vector is generated representing the *partial trajectory* in state space of behaviour from the start of the sequence to the present (or, depending on decay rate and sequence complexity, from some earlier time to the present). Using the AVQ algorithm, models of characteristic object behaviours can be learnt in an unsupervised manner from the extended observation of vectors from training sets $\mathcal{G}_j$. The resulting models comprise sets of v behaviour prototypes $\bar{\beta}_i$:

$$\mathcal{B} = \left\{\bar{\beta}_1,\ \bar{\beta}_2,\ \ldots,\ \bar{\beta}_v\right\}. \tag{3.26}$$

3.4.3 Experimental results - object location

Behaviour training sets $\mathcal{G}_j^{\text{loc}}$ were generated from the 622 unmodified state data sets $\mathcal{F}_j^{\text{loc}}$ and the set $\mathcal{A}^{\text{loc}}$ of 1,000 state prototypes generated in the experiment described in Section 3.3.1. The pre-processing of raw pedestrian trajectories was performed using the parameter values given in Section 3.3.1, and 1,000-dimensional behaviour vectors $\mathbf{G}_t$ were generated using a scaling factor $\rho = 1.4$ to scale proximity values and a decay coefficient $\gamma = 0.999$. γ was chosen to give a very slow decay rate relative to average sequence lengths so that behaviour vectors will encode entire trajectories of varying lengths with a similar level of detail. Ordered data sets were further re-sampled to improve density representation using a constant separation $\Delta = 0.15$. After pre-processing, training sets $\mathcal{G}_j^{\text{loc}}$ comprised a total of 23,270 behaviour vectors lying approximately within a unit hypercube.

A set $\mathcal{B}^{\text{loc}}$ of 1,000 behaviour prototypes was learnt from 2,000,000 iterations of AVQ over behaviour vectors from the training sets $\mathcal{G}_j^{\text{loc}}$. A constant $\beta = 0.01$ was used for sensitivity adjustments in the AVQ algorithm together with the two-stage cooling schedule described in Section

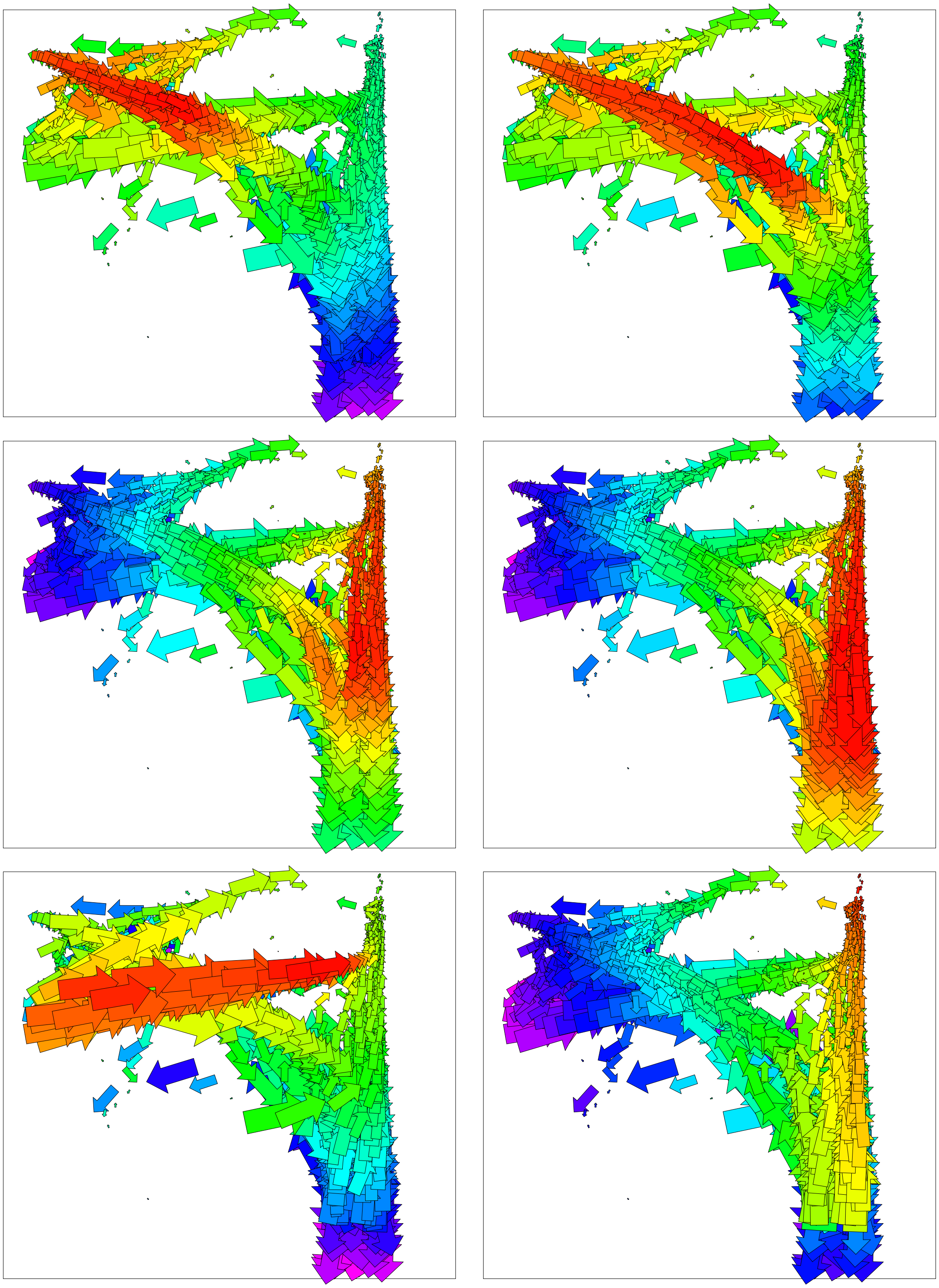


Figure 3.16: Learnt behaviour prototypes - object location.

3.2.1. Figure 3.16 illustrates a selection of the resulting behaviour prototypes, using the representation introduced in Figure 3.15. It is clear from this representation that the prototypes illustrated are plausible encodings of observed behaviours.

In order to further illustrate the results of the experiment, behaviour prototypes were used to partition partial trajectories from the raw data set. Within this scheme, behaviour vectors were generated as described above and, on each iteration, the current smoothed, sub-sampled, partial trajectory was allocated to the behaviour prototype which was closest, by the Euclidean metric, to the current behaviour vector. Figures 3.18 and 3.19 show the resulting partitioning of partial trajectories, where each box corresponds to one of the learnt behaviour prototypes. It is clear from these results that the region of behaviour space represented, in a *nearest-neighbour* sense, by each behaviour prototype encodes a subset of self-similar trajectories where similarity is based on an entire temporal history. It can also be seen that the more commonly occurring trajectories are represented by a greater proportion of the behaviour prototypes, and that there is less variability evident in the trajectories assigned to these prototypes, thus suggesting that density matching has, to some extent, been achieved.

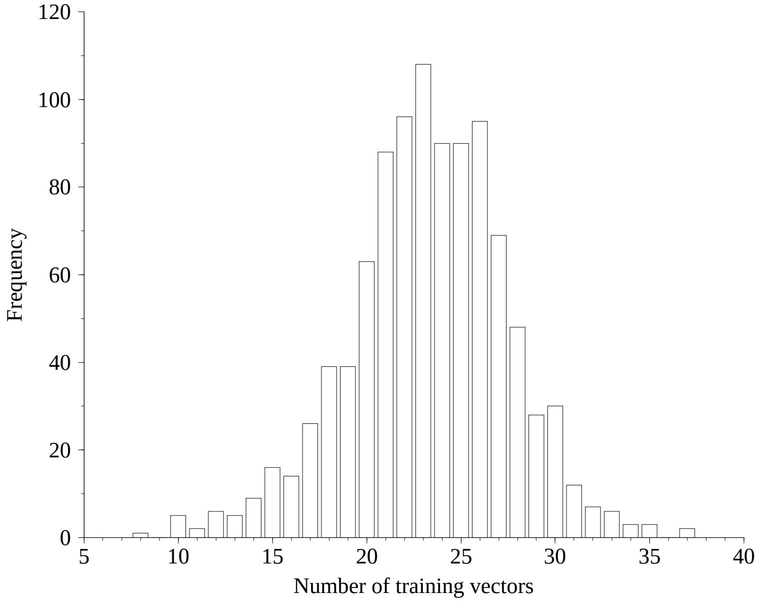


Figure 3.17: Frequency histogram illustrating behaviour prototype density matching - object location.

Finally, Figure 3.17 shows a frequency histogram which further illustrates density matching for the 1,000 behaviour prototypes and 23,270 training vectors used in this experiment. The mean of

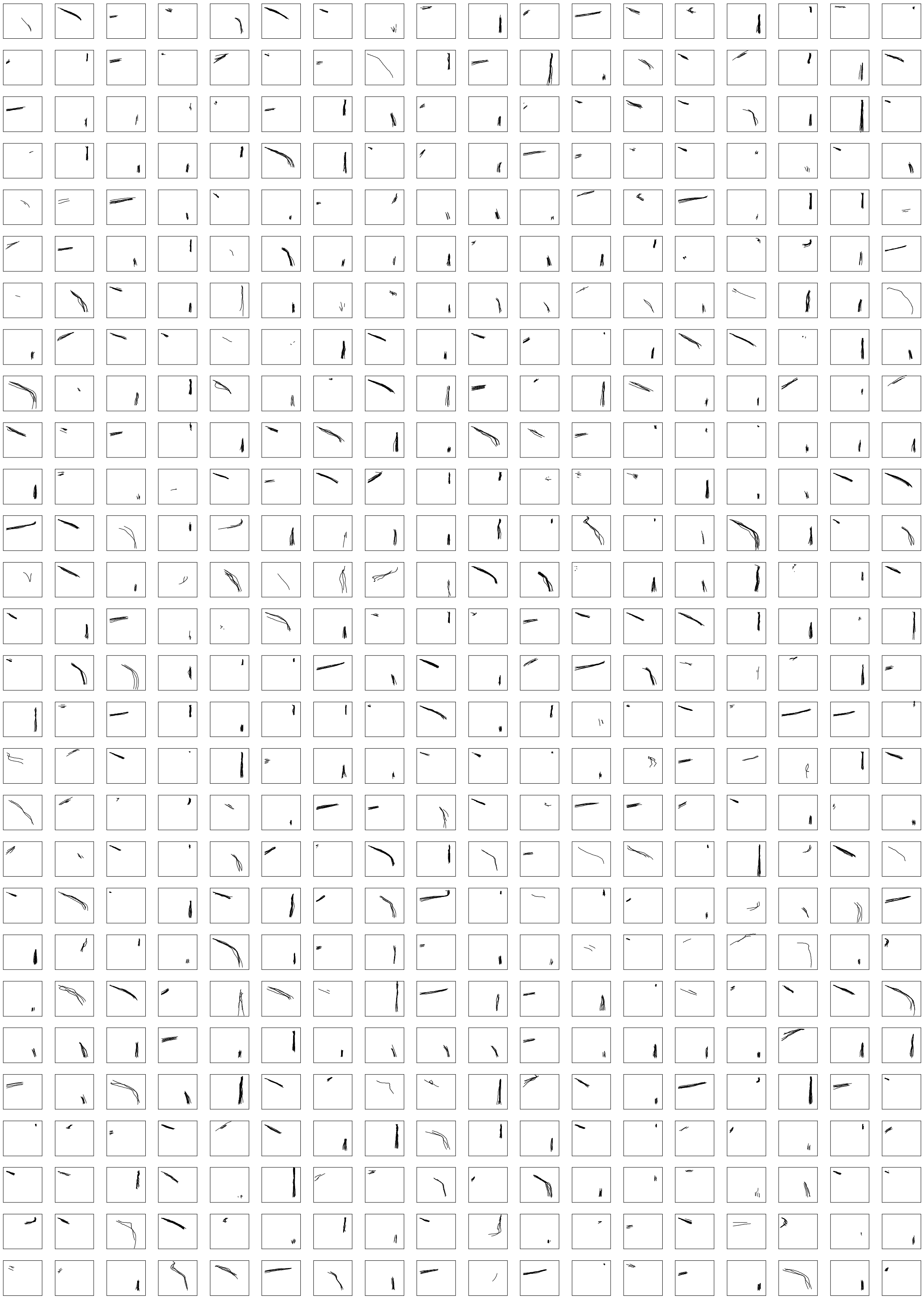


Figure 3.18: Partitioned pedestrian trajectories - prototypes 1–504.

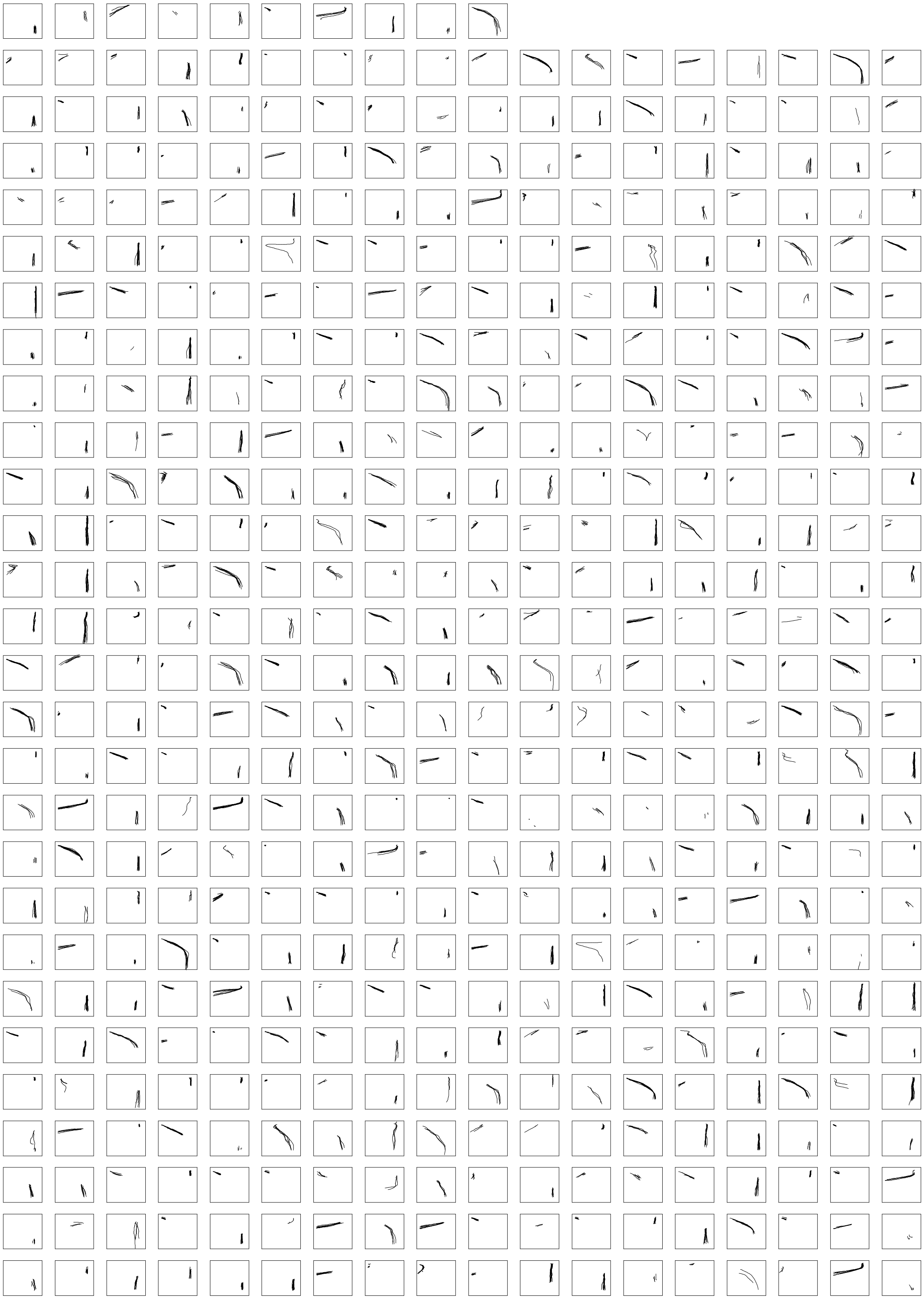


Figure 3.19: Partitioned pedestrian trajectories - prototypes 505–1000.

this approximately normal distribution is around 23, which is consistent with the expected value of 23.27, whilst the width of the distribution suggests some inaccuracy in density matching.

3.4.4 Experimental results - object shape

The behaviour training set $\mathcal{G}^{\text{shape}}$ was generated from the unmodified state data set $\mathcal{F}^{\text{shape}}$ and the set $\mathcal{A}^{\text{shape}}$ of 200 state prototypes generated in the experiment described in Section 3.3.2. The pre-processing of the raw shape sequence was performed using the parameter values given in Section 3.3.1, and 200-dimensional behaviour vectors $\mathbf{G}_t$ were generated using a scaling factor $\rho = 3.5$ to scale proximity values and a decay coefficient $\gamma = 0.999$. γ was chosen to give a relatively fast decay rate relative to the length of the entire sequence, thus avoiding behaviour component saturation during repeated exercise sub-sequences. Relative to the length of each sub-sequence, γ gives a slow enough decay rate for a trace of an exercise to be maintained throughout the four repetitions of the following exercise sub-sequence. In this way, behaviour vectors encode sufficient temporal information to disambiguate both the transitions between exercises and the repeated instances of each exercise sub-sequence. The ordered data set was further re-sampled to improve density representation using a constant separation $\Delta = 0.015$. After pre-processing, the training set $\mathcal{G}^{\text{shape}}$ comprised a total of 5,858 behaviour vectors lying approximately within a unit hypercube.

A set $\mathcal{B}^{\text{shape}}$ of 400 behaviour prototypes was learnt from 2,000,000 iterations of AVQ over behaviour vectors from the training set $\mathcal{G}^{\text{shape}}$. A constant $\beta = 0.01$ was used for sensitivity adjustments in the AVQ algorithm together with the two-stage cooling schedule described in Section 3.2.1. Since no reasonable method could be found to illustrate either the behaviour prototypes or their partitioning of the raw shape data set, only density matching results are presented. Figure 3.20 shows a frequency histogram illustrating density matching for the 400 behaviour prototypes and 5,858 training vectors used in this experiment. The mean of this approximately normal distribution is around 15 which is consistent with the expected value of 14.645, whilst the width of the distribution suggests little inaccuracy in density matching.

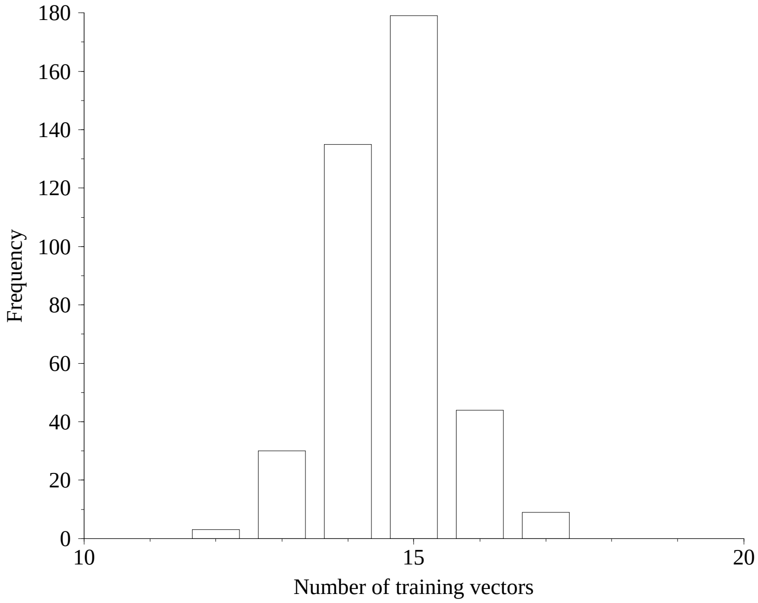


Figure 3.20: Frequency histogram illustrating behaviour prototype density matching - object shape.

3.5 Typicality assessment and incident detection

Having learnt models of probability density over the state and behaviour spaces of object characteristics exhibiting interesting behaviours, the statistical nature of these models can be immediately exploited to provide typicality assessment, where typicality is defined statistically. In addition, attentional control mechanisms which identify interesting incidents can be implemented via the detection of sufficiently atypical behaviours. Using automatically acquired behaviour models to approach such tasks negates the need for *a priori* knowledge and could thus prove powerful within the automated visual surveillance domain where inherently inaccurate hand-crafted knowledge has classically been employed (see Section 2.2). In this section, an effective typicality measure is introduced and experimental results are presented to demonstrate both a relative typicality partitioning of entire pedestrian trajectories, and continuous typicality assessment over the duration of a number of test trajectories. Typicality assessment results are included for both the state and behaviour models of pedestrian trajectories, and the advantages of employing the behaviour model for typicality assessment are demonstrated.

3.5.1 Local density estimation and prototype bounding

Since probability density has been modelled by the distribution of prototype vectors, typicality assessment can be achieved from the estimation of local probability density at each prototype. Assuming an exact density match has occurred, each prototype will represent an equal amount $\frac{1}{k}$ of probability, where k is the number of prototypes. Thus, by estimating the volume v_i within the state or behaviour space which is represented (in a *nearest-neighbour* sense) by prototype $\mathbf{c}_i$, and assuming probability density is constant within this region, an approximation to the local probability density p_i is given by

$$p_i = \frac{1}{kv_i}. \tag{3.27}$$

Unfortunately, even a simple hypercube-based estimate of v_i is impractical for high dimensional spaces due to rapid underflow in the digital floating-point representation. Instead, typicality assessment is achieved by considering the distribution of Euclidean distances between a prototype $\mathbf{c}_i$ and the sample vectors $\mathbf{z}_j$ it represents, using the mean

$$\mu_i = \frac{\sum_{j=1}^{n} |\mathbf{z}_j - \mathbf{c}_i|}{n} \tag{3.28}$$

of each distribution as a measure of relative *atypicality* in the region surrounding the corresponding prototype. Thus the atypicality of a feature $\mathbf{z}(t)$ is given by

$$A_t = \mu_j, \tag{3.29}$$

where

$$\left|\mathbf{z}(t) - \mathbf{c}_j\right| = \min_i \{|\mathbf{z}(t) - \mathbf{c}_i|\}. \tag{3.30}$$

Since a number of prototypes will border unpopulated areas of the distribution, it is necessary to estimate the boundary of the region represented by each prototype such that outlying features can be rejected. By again considering the distribution of Euclidean distances between a prototype and the sample vectors it represents, a simple hyperspherical boundary is realised by estimating the standard deviation

$$\sigma_i = \sqrt{\frac{\sum_{j=1}^{n} |\mathbf{z}_j - \mathbf{c}_i|^2}{n} - \mu_i^2} \tag{3.31}$$

of each distribution and rejecting features for which

$$\left|\mathbf{z}(t) - \mathbf{c}_j\right| > \mu_j + 3\sigma_j, \tag{3.32}$$

under the weak assumption that the distribution of distances is normal[3]. Rejected features are considered to have zero typicality.

Although atypicality values have no clear interpretation which would permit a sensible choice of threshold for the discrimination of atypical features, a sufficiently principled classification can be achieved if μ_i values are used to arrange prototypes in order of increasing probability density (decreasing atypicality). Since each prototype represents an approximately equal proportion of the distribution, comparisons of the form

$$\frac{r_j}{k} < \frac{f}{100}, \tag{3.33}$$

where r_j denotes the rank of the closest prototype $\mathbf{c}_j$ and $1 \leq f \leq 100$, can be used to ascertain whether the feature lies within the f percent of the distribution with least probability density, thus providing an intuitive decision support mechanism. In addition, this ranking allows *normalised* typicality values $T \in [0,\ 1]$ to be generated:

$$T_t = 1 - \frac{\mu_j - \mu_l}{\mu_m - \mu_l}, \tag{3.34}$$

where $r_l = k$ and $r_m = 1$.

If adaptivity is required, both μ_i and σ_i values can be updated during extended learning using either iterative update equations or moving temporal windows, whilst adjustments to prototype ordering can be performed each time a μ_i value changes.

Finally, some post-processing of typicality sequences is required to remove occasional zero-going spikes. These spikes are partly due to inaccuracies in the bounding of prototypes which occasionally results in small ‘holes’ within the distribution which cause features to be rejected and assigned a typicality $T_t = 0$. Spikes also occur more frequently when a trajectory closely skirts the boundary of the distribution, and thus continually moves in and out of the hyperspherical boundaries of the outermost prototypes. Spikes can be minimised by median filtering typicality sequences over a moving temporal window of width w.

[3] Experimental evidence indicates that the distribution of Euclidean distances between a prototype and the sample vectors it represents is often skewed in the direction of increasing distance.

3.5.2 Experimental results - pedestrian trajectories

During a further learning phase, the distributions of Euclidean distances between the set $\mathcal{A}^{\mathrm{loc}}$ of 1,000 state prototypes and the corresponding state vectors from the 622 state training sets $\mathcal{F}_j^{\mathrm{loc}}$ (from the experiment described in Section 3.3.1) were estimated using iterative update equations derived from Equations 3.28 and 3.31. Similarly, distributions were estimated for the set $\mathcal{B}^{\mathrm{loc}}$ of 1,000 behaviour prototypes and the corresponding behaviour vectors from the 622 behaviour training sets $\mathcal{G}_j^{\mathrm{loc}}$ (from the experiment described in Section 3.4.3).

Figures 3.21 and 3.22 illustrate the distribution of μ_i values for the state and behaviour models respectively, where frequency graphs were generated by dividing the range of observed μ_i values into 20 classes of equal width. Both distributions have a similar skewed shape which is intuitively appealing since it indicates that most states and behaviours are reasonably typical whilst very few are highly typical or highly atypical.

To illustrate the types of pedestrian behaviours which correspond to different typicalities, the set of 622 *complete* trajectories illustrated in Figure 3.1(b) were partitioned into four classes based on the atypicality of their final behaviour vector. Figure 3.23 illustrates the results of this partitioning, where trajectories which lie outside the behaviour distribution have not been shown. The changing nature of trajectories over the four classes clearly shows a plausible typicality-based partitioning.

Finally, to demonstrate the continuous typicality assessment of pedestrian trajectories, Figures 3.24 and 3.25 illustrate the results of assessing three normal and three atypical trajectories from test data sets captured soon after the training data. In each figure, trajectories ((a), (c), and (e)) are represented by state vector sequences where each state vector is illustrated by a single arrow as per the state prototypes in Figure 3.8, whilst the corresponding graphs ((b), (d), and (f)) illustrate normalised state and behaviour typicality throughout each sequence. Typicality assessment was performed using a threshold of 5% to reject atypical states and behaviours, whilst spikes were removed from typicality sequences by median filtering over a window of width $w = 5$.

In Figure 3.24, the normal test trajectories are seen to have reasonably high state and behaviour typicalities over the entire duration of each trajectory. However, in Figure 3.25 the advantages of behaviour typicality assessment over state typicality assessment are clearly illustrated. Figure 3.25

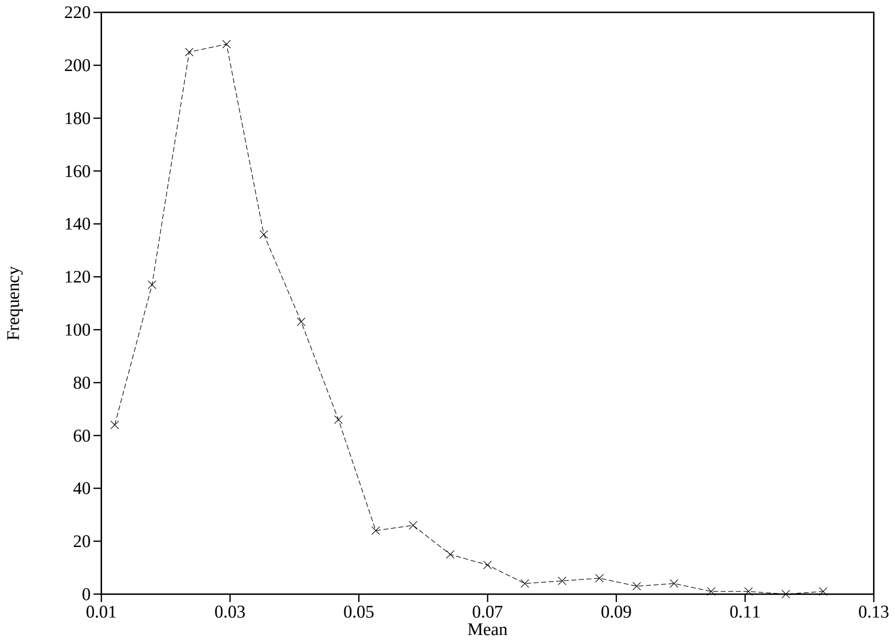


Figure 3.21: State atypicality distribution - object location.

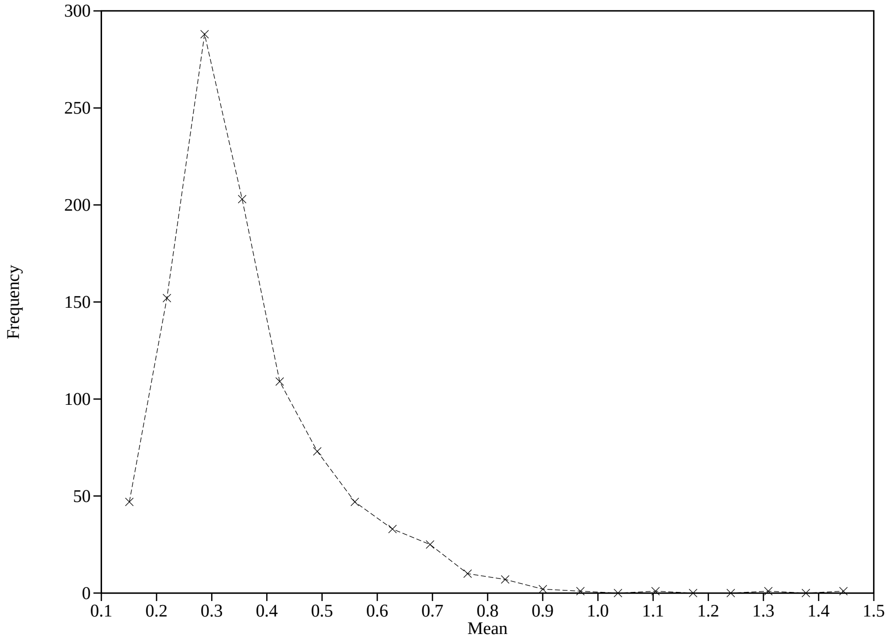
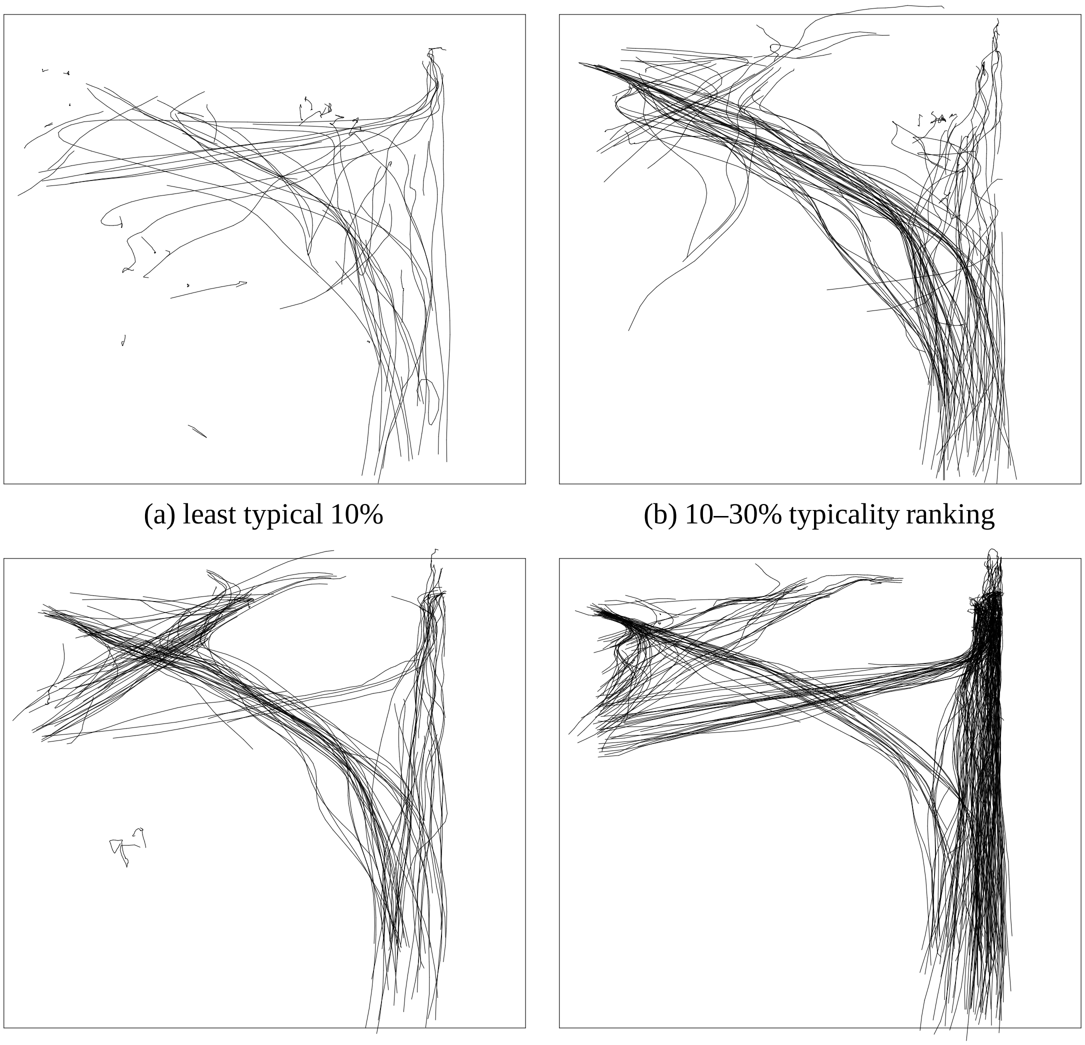


Figure 3.22: Behaviour atypicality distribution - object location.

(a) and (b) again show equal performance for both state and behaviour typicality assessment - both models reject the unusually fast trajectory which actually corresponds to a tracked cyclist! Figure 3.25 (c) and (d) show an atypical trajectory which has two distinct phases, each of which is a part of a typical trajectory. Whilst state typicality only drops slightly during the inter-phase transi-



(a) least typical 10%

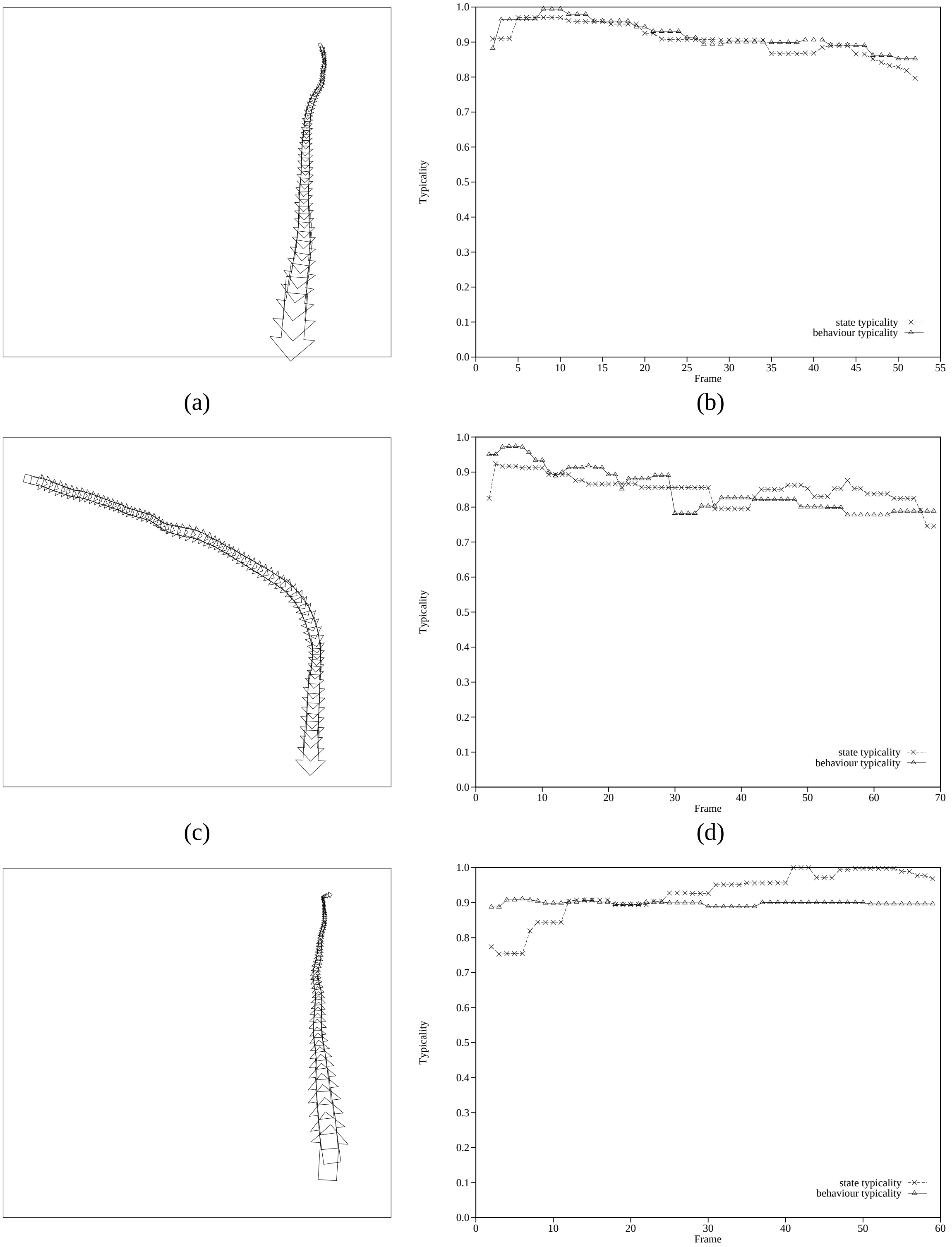
(b) 10–30% typicality ranking

(c) 30–50% typicality ranking

(d) most typical 50%

Figure 3.23: Typicality-based pedestrian trajectory partitioning.

tion, behaviour typicality drops to zero during the transition and remains at zero for the remainder of the trajectory. Similarly, Figure 3.25 (e) and (f) show an atypical trajectory with three distinct phases, the middle phase being previously unseen whilst the first and last phases correspond to the start and end of typical trajectories. Whilst both models perform similarly during the first two phases, state typicality recovers during the final phase whilst behaviour typicality remains at zero.

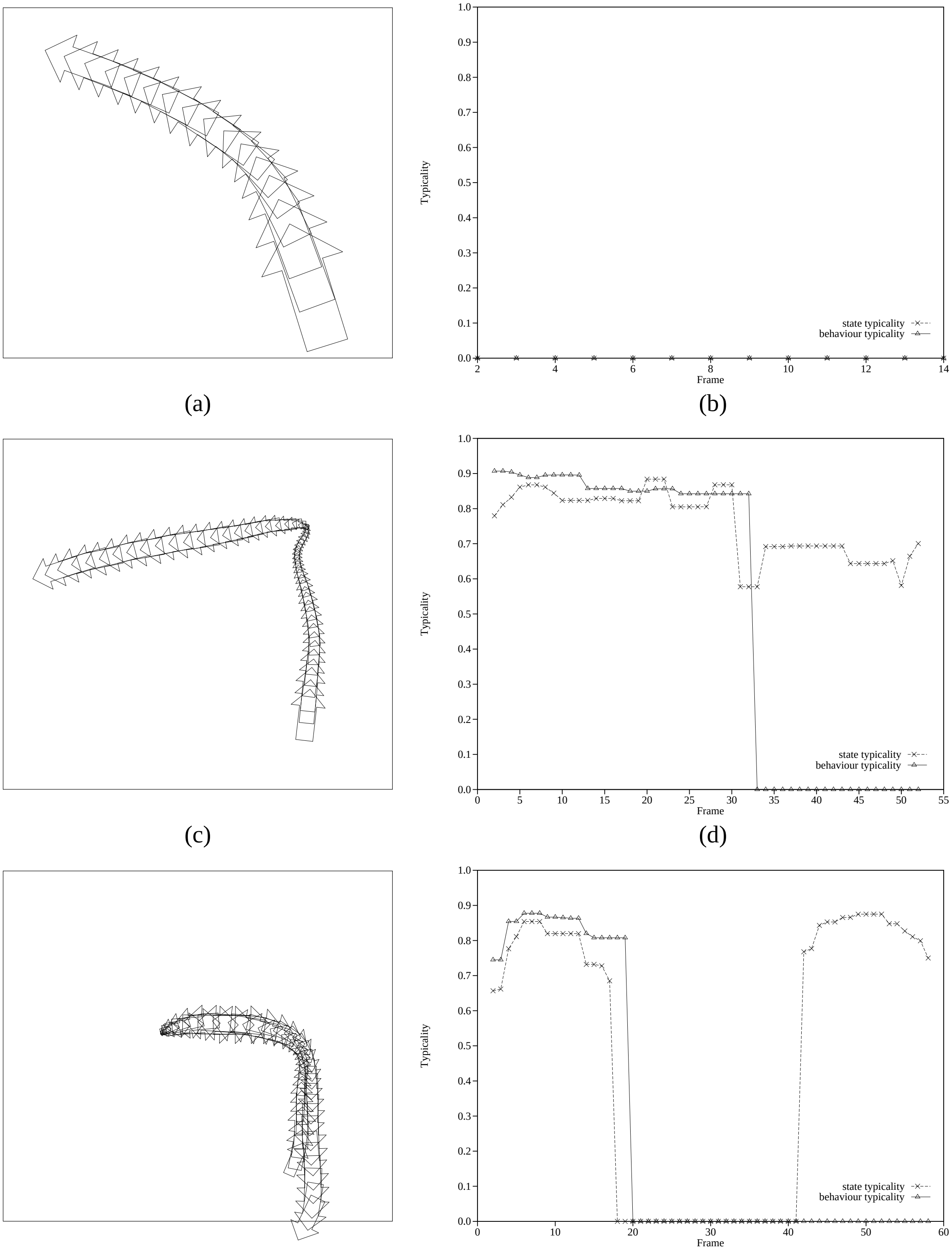


(a) (b)

(c) (d)

(e) (f)

Figure 3.24: Typicality assessment - normal pedestrian trajectories.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 3.25: Typicality assessment - atypical pedestrian trajectories.

3.6 Discussion

In this chapter, techniques have been developed allowing the acquisition of models of characteristic object states and behaviours from the continuous observation of long image sequences, and experimental results presented for two object characteristics with distinctly different properties. Models constitute an optimised sample-set representation of probability density, which is both highly specific and reasonably compact, and are learnt in an unsupervised manner using an extension to the standard iterative VQ algorithm - dubbed Altruistic Vector Quantization (AVQ) - which provides increased robustness and improved density matching (demonstrated experimentally for the scalar case).

The representation of object behaviours over varying temporal intervals has been achieved using a novel temporal pattern formation strategy to encode sequences of state vectors. Using this representation, simple, non-repeating, sequences of varying lengths can be encoded whilst maintaining a similar level of detail, whilst results presented in Chapter 4 will indicate that certain complex sequences involving repeated sub-sequences may also be encoded effectively.

By exploiting the statistical nature of behaviour models, a typicality measure has been derived which allows both the continuous assessment of behaviour typicality and the implementation of an attentional control mechanism through the identification of interesting (sufficiently atypical) incidents. Such capabilities are particularly applicable within the visual surveillance domain, providing objective attention cues to a human operator which are based entirely on the frequency of occurrence of previously observed behaviours.

Although not demonstrated within this thesis, the discrete nature of state and behaviour models allows semantics to be associated with different classes of actions or behaviours, thus facilitating event and gesture recognition as well as providing cues for higher-level reasoning systems. Such semantic labelling could be achieved during a further supervised learning phase, using majority voting to assign prototype labels, and perhaps employing Kohonen's Learning Vector Quantization (LVQ) strategies [51] to derive near-optimal decision boundaries between classes.

3.6.1 Dissimilarity metrics

Within the techniques developed in this chapter, the Euclidean distance is used as a measure of the dissimilarity between points in both state and behaviour spaces. When this dissimilarity metric is applied to sets of B-spline control points, the resulting sense of shape dissimilarity is often contra-intuitive. A better dissimilarity metric could be achieved either from the use of a landmark-based shape representation (see, for example, Cootes *et al.* [22]), or by instead measuring the distance between corresponding points sampled densely over the parametric curves, as suggested by Baumberg [7]. Similarly, when using this dissimilarity metric within behaviour spaces defined by the temporal pattern formation strategy, it is uncertain to how great an extent measured dissimilarities emulate the dissimilarities we perceive.

3.6.2 Temporal adaptation

As stated in Chapter 1, a natural process for the perception of behaviour models should allow gradual temporal adaptation, enabling model evolution with occasional changes in characteristic behaviour. Using the techniques developed in this chapter, such temporal adaptation can be achieved through extended learning, using a low gain coefficient in the AVQ algorithm and iteratively updating prototype typicality values as proposed in Section 3.5.1.

Assuming changes in characteristic behaviour are slow and continuous, state prototypes will adapt smoothly to the changing state distribution. As state prototypes move, temporal proximity patterns will gradually alter, and behaviour prototypes will adapt smoothly to the changing behaviour distribution. Thus, as characteristic behaviour changes, both state and behaviour prototypes will adapt, whilst changes in the probability density local to each prototype will result in the evolution of typicality values.

If changes in behaviour are more rapid or discontinuous, the sensitivity mechanism in the AVQ algorithm will prevent the loss of stranded prototypes and will ensure that prototypes eventually adapt to represent the modified distributions. An extreme case of such changes is encountered at the start of the learning process when prototypes are randomly distributed. Experiments designed to illustrate the concurrent acquisition of both state and behaviour models, using twice the number of iterations in the AVQ algorithm, give comparable results to those presented within this chapter, thus also demonstrating worst-case temporal adaptation.

3.6.3 Self-Organizing Maps

An interesting extension to the Vector Quantization and Competitive Learning paradigms is the Self-Organizing Maps (SOMs) (or Topographic Mappings) of Kohonen [51]. In addition to producing a quantization of feature space, these artificial neural networks undergo a self-organization process which results in a network in which similarity relationships within feature space are preserved in the lattice structure of the prototypes. Self-organization is achieved by defining a temporally shrinking neighbourhood relationship between prototypes and extending the VQ algorithm described in Section 3.2.1 such that, on each iteration, the neighbours of the winning prototype are also moved towards the current input vector.

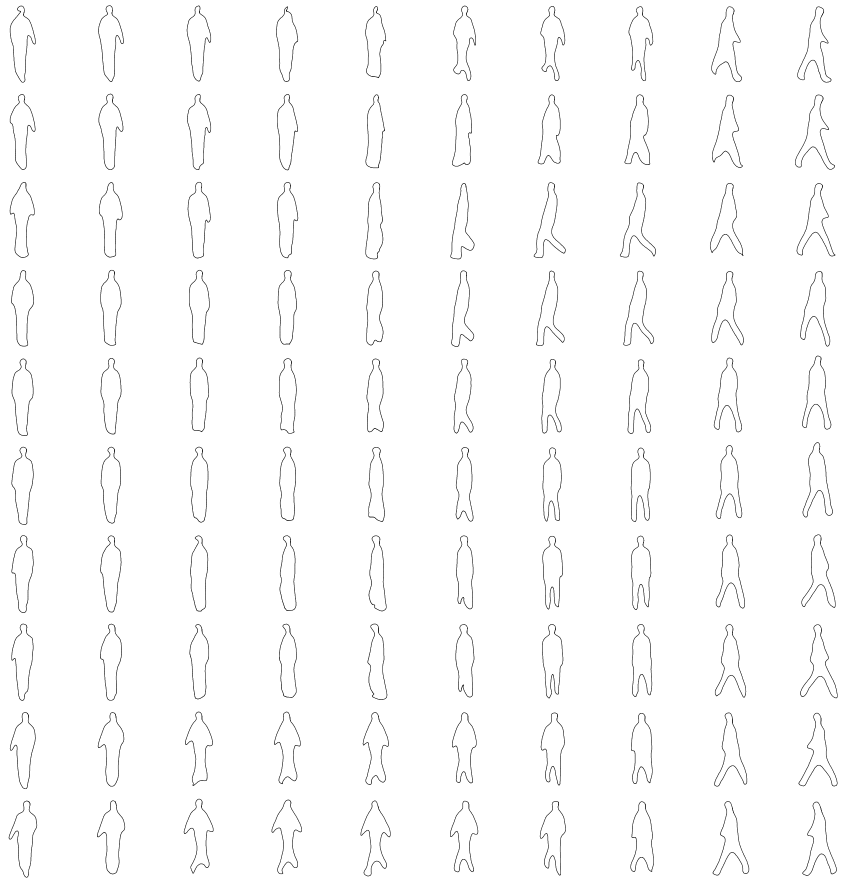


Figure 3.26: 2-dimensional SOM fitted to pedestrian shape data.

It is conceivable that the information provided by such a spatial ordering may be of value to higher-level reasoning systems, whilst a low-dimensional parameterisation may be of use in, for example, object tracking and gesture recognition. Limited experiments have therefore been performed to investigate the fitting of both 1-dimensional (chain) and 2-dimensional (sheet) SOMs to experimental shape data similar to that described in Section 3.1, using the standard SOM algorithm as described by Kohonen [51]. For example, Figure 3.26 illustrates the results of an experiment in which a 2-dimensional map was fitted to pedestrian shape data, and in which a reasonable parameterisation is achieved. Whilst experimental results were generally encouraging, the following factors severely limit the utility of the algorithm:

- As suggested by the limited theoretical density matching results for the SOM algorithm (see, for example, Ritter [75]), density matching is poor with areas of high probability density under-represented and areas of low probability density over-represented.
- When maps are fitted to distributions which are discontinuous or which have a complex structure, the lattice may become 'stretched' across unpopulated regions of feature space, resulting in a sub-optimal distribution and discontinuities in the similarity relationships across the lattice.
- The tendency of chains/sheets to form space-filling curves/surfaces when fitted to higher dimensional distributions distorts similarity relationships since similar features may map to distinct locations on the lattice.

Whilst the first of these limitations may be addressed using the sensitivity mechanism described in Section 3.2.2.1 (or by techniques such as minimum distortion encoding (Luttrell [56]) or nonlinear weight adjustments (Zheng and Greenleaf [94])), the remaining limitations are due to the fixed topology, size, and dimensionality of the lattice. Perhaps the most promising technique for the addition of spatial self-organization to behavioural models is thus the 'cell structure' growing algorithm described by Fritzke [31]. In this approach, both the topology and size of a fixed-dimensional simplex mesh are determined during learning via an iterative process of cell insertion and occasional cell removal which resembles *fractal growth*. Results presented in [31] suggest that the algorithm is capable of providing an efficient representation of complex, possibly discontinuous, distributions whilst achieving reasonable density matching.

Chapter 4

Behaviour generation

This chapter describes the enhancement of the models developed in Chapter 3 to include generative capabilities via the superimposition of learnt probabilistic prediction schemes. Using this technique, both maximum likelihood behaviour extrapolation[1] and the stochastic generation of realistic sample behaviours are demonstrated. To further demonstrate the utility of predictive models, the performance of both state-based and behaviour-based predictors is compared with a linear prediction scheme. Finally, the similarities between the enhanced models and Hidden Markov Models, commonly used for the recognition of gesture and speech, are discussed.

4.1 Generating predictive models

The state and behaviour models developed in Chapter 3 are deficient in the sense that they do not support the performance of generative tasks such as the prediction or extrapolation of future behaviours or the generation of realistic sample behaviours. In state models, this deficiency is simply due to the presence of insufficient temporal information. In behaviour models, sufficient temporal information exists but cannot be exploited due to the limited reconstructive capabilities of the behaviour representation. If it were possible to reconstruct an approximation to the sequences represented by behaviour prototypes, then generative tasks could be achieved via some form of sequence matching process. Unfortunately, as stated in Section 3.4.1, such reconstruction is not possible, al-

[1] The term *extrapolation* refers to the generation of future behaviour over a number of contiguous time instants.

though it is possible to obtain the state prototype associated with the most recent proximity maximum by finding the highest valued component of the behaviour prototype, and thus an estimate of the *current* state can be obtained for each behaviour prototype.

Since both state and behaviour models are discrete representations, the addition of generative capabilities can be achieved during a further learning phase in which the relative probabilities of transitions between prototypes are estimated. Thus model prototypes are associated with the states of a time-homogeneous finite Markov chain (see, for example, Lawler [53]), and the state vector associated with each prototype (i.e. the prototype itself in state models and an estimate of the current state in behaviour models) becomes the token associated with the corresponding chain state.

4.1.1 Markov chain acquisition

The Markov chain $\mathcal{M}$ superimposed on a set of state or behaviour prototypes is defined by the 4-tuple

$$\mathcal{M} = \langle \mathcal{E},\, \mathcal{S},\, \pi,\, \mathcal{T} \rangle, \tag{4.1}$$

where

$$\mathcal{E} = \{e_1,\, e_2,\, \ldots,\, e_{k+1}\} \tag{4.2}$$

is the set of chain states, each of which corresponds to a state or behaviour prototype except e_{k+1} which represents the *end state*,

$$\mathcal{S} = \{\bar{\alpha}(e_1),\, \bar{\alpha}(e_2),\, \ldots,\, \bar{\alpha}(e_k)\} \tag{4.3}$$

is the set of state vector tokens associated with the chain states,

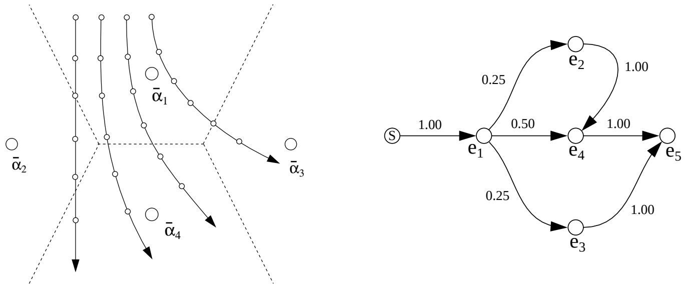
$$\pi = \{\pi_1,\, \pi_2,\, \ldots,\, \pi_k\}\,,\; \pi_i = \mathrm{P}(e_i \text{ at step } r = 0) \tag{4.4}$$

defines the initial state distribution, and finally,

$$\mathcal{T} = \begin{bmatrix} \mathcal{T}_{1,1} & \ldots & \mathcal{T}_{1,k+1} \\ \vdots & \ddots & \vdots \\ \mathcal{T}_{k,1} & \ldots & \mathcal{T}_{k,k+1} \end{bmatrix},\; \mathcal{T}_{i,j} = \mathrm{P}(e_j \text{ at step } r+1 \mid e_i \text{ at step } r) \tag{4.5}$$

is a matrix defining the state transition distribution. Thus, if the Markov chain is superimposed on a set of state prototypes, then $e_i \mapsto \bar{\alpha}_i$ and $\bar{\alpha}(e_i) = \bar{\alpha}_i$, whereas, if the chain is superimposed on a set of behaviour prototypes, then $e_i \mapsto \bar{\beta}_i$ and each $\bar{\alpha}(e_i)$ is an estimate of the current state.

The initial state distribution π and state transition distribution $\mathcal{T}$ are estimated from training sets $\mathcal{F}_j$ or $\mathcal{G}_j$ during a further learning phase by observing the closest prototype, in a *nearest neighbour* sense, to the current training vector at each time instant. Thus π is estimated from the relative frequency of starting at each prototype, whilst $\mathcal{T}$ is estimated from the relative frequency of the transitions between prototypes, considering only transitions which cause state changes (i.e. $\mathcal{T}_{i,i} = 0$).



$$\begin{aligned} \mathcal{E} &= \{e_1,\, e_2,\, e_3,\, e_4,\, e_5\} \\ \mathcal{S} &= \{\bar{\alpha}_1,\, \bar{\alpha}_2,\, \bar{\alpha}_3,\, \bar{\alpha}_4\} \\ \pi &= \{1,\, 0,\, 0,\, 0\} \\ \mathcal{T} &= \begin{bmatrix} 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

(a) (b) (c)

Figure 4.1: Markov chain acquisition.

The acquisition of Markov chains is illustrated in Figure 4.1 using a simple example. Figure 4.1(a) depicts a 2-dimensional state space containing four prototypes. In this illustration, broken lines delimit the Voronoi regions [35] about each prototype (corresponding to a *nearest neighbour* partitioning of state space) and the paths of four training sequences are shown. Figure 4.1(b) gives a graphical representation of the Markov chain acquired from observing the paths the four training sequences trace through the state space, whilst Figure 4.1(c) enumerates the members of the corresponding 4-tuple $\mathcal{M}$.

4.1.1.1 Typicality-based transition pruning

When acquiring initial state and state transition distributions from training data, atypical training sequences may have a detrimental effect on the learnt prediction models. For instance, training sequences which lie entirely outside the boundary of a particular state or behaviour distribution will give rise to *transition noise* in the form of misleading or apparently impossible transitions as sequences cross the Voronoi regions of the bounding prototypes, whilst transitions between prototypes within areas of minimal probability density will be of little practical use since quantization is most coarse in these regions.

Continuous typicality assessment during the acquisition of Markov chains allows transitions which occur when typicality is below a given threshold percentage f to be rejected. Thus, with a suitable choice of f, transitions involving prototypes within areas of minimal probability density can be effectively pruned whilst transition noise is minimised.

4.1.1.2 Markovian property

The Markov chain defined by Equations 4.1–4.5 forms a first-order stochastic process description since the state transition distribution is conditioned only on the current chain state, under the assumption that there is no higher-order state dependency (the *Markovian property*). Clearly, when such a chain is superimposed on a set of state prototypes, the acquired transition distribution fails to represent any higher-order temporal dependencies which exist within the training data. If, however, a Markov chain is superimposed on a set of behaviour prototypes, then higher-order temporal dependencies are successfully represented, since the activation of each behaviour prototype requires that a particular history has been observed.

Thus, if temporal dependencies are inherent in training sequences, a predictor based on a behaviour model will encode these temporal dependencies within its transition structure and is consequently more powerful than the corresponding state-based predictor. Since a higher-order process description (in which the transition distribution, represented as a tensor, is conditioned on a number of previous states) can be expanded to form an equivalent Markov chain, a behaviour-based predictor will closely resemble the predictor generated if a sufficiently high-order process description, superimposed on the corresponding set of state prototypes, is expanded.

4.1.2 Generating maximum likelihood and stochastic extrapolations

Prediction, extrapolation, and the generation of sample behaviours are achieved by traversing a Markov chain until the end state is reached, selecting either the most likely transition (maximum likelihood extrapolation) or sampling from the transition distribution (stochastic extrapolation) on each iteration. Maximum likelihood extrapolation is achieved by selecting each transition randomly from the (generally singleton) set of transitions with equally maximal probability, whilst stochastic extrapolation is achieved by selecting transitions via sampling (using a partitioning of unity) from the transition distribution. Traversal of a Markov chain results in an ordered set of state vector tokens $\bar{\alpha}(e_{i_r})$ associated with the visited chain states:

$$\mathcal{Q} = \left\{\bar{\alpha}(e_{i_0}),\ \bar{\alpha}(e_{i_1}),\ \ldots,\ \bar{\alpha}(e_{i_l})\right\}, \tag{4.6}$$

where the time interval between successive state vectors is initially unspecified and e_{i_0}, the initial chain state, is identified using the state or behaviour model when performing prediction or extrapolation and is selected from the initial state distribution when generating sample behaviours. Selection from the initial state distribution is again achieved either by selecting randomly from the (generally singleton) set of equally maximal probability start states or by basing the selection on sampling (using a partitioning of unity) from the initial state distribution. When performing prediction or extrapolation, $\bar{\alpha}(e_{i_0})$ is replaced by the current state vector $\mathbf{F}_t$ to ensure a smooth join between previous behaviour and the extrapolation.

4.1.2.1 State sequence interpolation

Since the time interval between successive state vectors in $\mathcal{Q}$ is initially unspecified, depending to a great extent on the local probability density within the corresponding state or behaviour model, a regularly sampled extrapolation requires the interpolation of $\mathcal{Q}$ and the approximation of the time interval between successive state vectors. Assuming constant acceleration and a linear path between state vectors $\bar{\alpha}(e_{i_r})$ and $\bar{\alpha}(e_{i_{r+1}})$, the time interval δ_r can be approximated from the mean speed and separation of the constituent characteristic vectors:

$$\delta_r = 2\frac{|\mathbf{C}_{r+1} - \mathbf{C}_r|}{|\dot{\mathbf{C}}_{r+1}| + |\dot{\mathbf{C}}_r|}, \tag{4.7}$$

where $\delta_r = 0$ if the denominator is 0.

A piecewise linear interpolation of $\mathcal{Q}$ will fail to express the non-linear changes which may occur between state vectors separated by large time intervals, and thus a higher-degree polynomial may be more appropriate. Since state vectors can place four constraints on each polynomial, two endpoints and two tangent vectors, a Hermite (cubic) interpolation is used. In an extension to the standard Hermite curve definition (see, for example, Foley *et al.* [30]), both characteristic vectors and their differentials are interpolated. Thus each parametric curve segment

$$\mathbf{Q}_r(t) = \begin{bmatrix} \mathbf{C}(t) \\ \delta_r\dot{\mathbf{C}}(t) \end{bmatrix} \tag{4.8}$$

is defined over the interval $0 \leq t \leq 1$ by

$$\mathbf{Q}_r(t) = \mathbf{TBE} = \begin{bmatrix} t^3 & t^2 & t & 1 \\ 3t^2 & 2t & 1 & 0 \end{bmatrix} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{C}_r \\ \mathbf{C}_{r+1} \\ \delta_r \dot{\mathbf{C}}_r \\ \delta_r \dot{\mathbf{C}}_{r+1} \end{bmatrix}, \tag{4.9}$$

where **B** is the *Hermite basis matrix*, **E** is the *Hermite geometry matrix*, and the differentials of characteristic vectors have been transformed into tangent vectors using a scaling factor δ_r.

A temporally regular extrapolation $\mathcal{P}$ is thus produced by sampling the Hermite interpolant of $\mathcal{Q}$ at regular time instants:

$$\mathcal{P} = (\mathbf{F}_{t+1},\, \mathbf{F}_{t+2},\, \ldots,\, \mathbf{F}_{t+l}), \tag{4.10}$$

where the time interval between the start of each curve segment and the desired sample instant is transformed into the curve's unit time scale using a scaling factor $\frac{1}{\delta_r}$, and successive state vectors are reconstructed from the components of **Q**.

4.1.3 Improving behaviour-based prediction

When a Markov chain is superimposed on a set of behaviour prototypes, the estimates of current state, obtained by finding the highest valued component of each behaviour prototype, are often found to be rather poor when compared to the actual mean current state of the behaviours represented by each of the prototypes, thus adding to spatio-temporal inaccuracy within models. Further, it is often found that the same state prototype is associated with sequential chain states, and thus traversal of the chain results in sequences in which adjacent state vector tokens may be identical. Whilst these identical state vectors do not affect the interpolation of sequences, since the approximation of the time interval between identical vectors will yield a value of zero, their presence does indicate a loss of detail in the representation of extrapolations, thus further adding to spatio-temporal inaccuracy within models.

To eliminate these additional spatio-temporal inaccuracies, current state estimates are replaced during the learning of the Markov chain distributions with the actual mean current state of the behaviours represented by each of the behaviour prototypes:

$$\bar{\alpha}(e_i) = \frac{\sum_{j=1}^{n} \mathbf{F}_j}{n}, \tag{4.11}$$

where the $\mathbf{F}_j$ are the current state vectors associated with the behaviours represented by behaviour prototype $\bar{\beta}_i$. If typicality-based transition pruning is being performed, then the current state vectors associated with atypical behaviours are not included in this summation.

4.1.4 Stochastic behaviour perturbation

When using stochastic predictions or extrapolations to aid in tracking or to produce realistic sample behaviours, it may be advantageous to perturb each state vector token using an additive noise process associated with the corresponding chain state, thus better representing the variation in sequences represented by the underlying state or behaviour model. This technique would be of particular benefit if the predictor were being used within a stochastic tracking algorithm such as Isard and Blake's CONDENSATION [46]. The inclusion of noise models extends the definition of Markov chains to the 5-tuple

$$\mathcal{M} = \langle \mathcal{E},\, \mathcal{S},\, \pi,\, \mathcal{T},\, \mathcal{N} \rangle\,, \tag{4.12}$$

effectively a Hidden Markov Model (see Section 4.4.1), where

$$\mathcal{N} = \{\mathbf{n}_1,\, \mathbf{n}_2,\, \ldots,\, \mathbf{n}_k\}\,, \tag{4.13}$$

and each $\mathbf{n}_i$ is a vector of noise model parameters. Traversal of such an enhanced chain thus results in an ordered set of perturbed state vector tokens:

$$\mathcal{Q} = \left\{\bar{\alpha}(e_{i_0}) + \mathbf{w}_{i_0},\, \bar{\alpha}(e_{i_1}) + \mathbf{w}_{i_1},\, \ldots,\, \bar{\alpha}(e_{i_l}) + \mathbf{w}_{i_l}\right\}\,, \tag{4.14}$$

where each $\mathbf{w}_i$ is sampled from the corresponding noise model $\mathbf{n}_i$.

If state prototypes are used as tokens within the chain, then rudimentary noise models can be generated by considering the distribution of Euclidean distances between each state prototype and the sample vectors it represents, using the parameters estimated for typicality assessment in Section 3.5.2 to generate noise vectors whose magnitude is normally distributed (i.e. $|\mathbf{w}_i| \sim \mathrm{N}(\mu_i,\, \sigma_i)$). Unfortunately, such isotropic noise processes are probably inadequate, particularly for state spaces such as the shape model described in Section 3.3.2 in which the distribution of sample vectors represented by each state prototype is typically elongated parallel to sample trajectories. To resolve such problems, more refined noise models could be generated by estimating the parameters of multivariate normal distributions which better represent the distribution of sample vectors around each state vector token $\bar{\alpha}(e_i)$, resulting in multivariate normally distributed noise vectors $\mathbf{w}_i \sim \mathrm{N}(\bar{\mu}_i,\, \Sigma_i)$.

4.1.5 Assessing predictor performance

As well as generating qualitative experimental results such as extrapolations and entirely hypothetical sample behaviours which demonstrate the utility of predictive models, it is also possible to generate quantitative experimental results which demonstrate predictor performance. The experiment described here allows state-based and behaviour-based predictors to be assessed by observing the deterioration in mean performance over time, using the linear prediction scheme:

$$\mathbf{C}_{t+T} = \mathbf{C}_t + T\dot{\mathbf{C}}_t, \tag{4.15}$$

to provide a standard comparison in which prediction is based entirely on the value of the current state vector $\mathbf{F}_t$.

For each prediction scheme (linear, state-based, and behaviour-based), a set of root mean square (RMS) errors is calculated to quantify the mean performance in predicting the value of the characteristic vector on each future time instant:

$$E_T = \sqrt{\frac{\sum_{j=1}^{n} \left| \mathbf{C}_{t+T} - \mathbf{C}^*_{t+T} \right|_j^2}{n}}, \tag{4.16}$$

where the error E_T in predicting T time steps into the future is averaged over predictions generated on every frame of every test sequence and $\mathbf{C}^*_{t+T}$ denotes the *ground truth* characteristic vector at time $t+T$ as given by the test data. State-based and behaviour-based predictions are only generated if the current state or behaviour falls within the bounds of the corresponding distribution, and errors are only updated if both a prediction and the ground truth exist for the particular T.

Unlike the linear prediction scheme, the predictive models developed in this chapter are non-deterministic in nature, and thus their mean performance should represent a probabilistic weighting of the errors given by all possible predictions. Unfortunately, it is not, in general, possible to enumerate the entire set of possible predictions from a particular chain state due to the possibility of cycles within the transition structure. Instead, mean performance is calculated using Monte Carlo simulation, generating a large number of stochastic predictions on each frame and allowing their relative frequency to provide probabilistic weighting within the calculation of RMS errors.

4.2 Experimental results - object location

A 1,001-state Markov chain $\mathcal{M}_\alpha^{\text{loc}}$ was superimposed on the set $\mathcal{A}^{\text{loc}}$ of 1,000 state prototypes generated in the experiment described in Section 3.3.1, using state prototypes as the token set (i.e. $\mathcal{S}_\alpha^{\text{loc}} = \mathcal{A}^{\text{loc}}$). Initial state and state transition distributions were estimated from the 622 state training sets $\mathcal{F}_j^{\text{loc}}$, using a typicality threshold of 5% to minimise transition noise and prune transitions involving atypical prototypes. A value of $\Delta = 0.01$, half that used in Section 3.3.1, was used to re-sample training sets as described in Section 3.2.2, thus reducing the tendency to omit transitions associated with brief entry into a prototype's Voronoi region.

A 1,001-state Markov chain $\mathcal{M}_\beta^{\text{loc}}$ was superimposed on the set $\mathcal{B}^{\text{loc}}$ of 1,000 behaviour prototypes generated in the experiment described in Section 3.4.3, estimating the token set $\mathcal{S}_\beta^{\text{loc}}$ during learning as described in Section 4.1.3. Initial state and state transition distributions were estimated from the 622 behaviour training sets $\mathcal{G}_j^{\text{loc}}$, using a typicality threshold of 5% to minimise transition noise and prune transitions involving atypical prototypes. A value of $\Delta = 0.075$, half that used in Section 3.4.3, was used to re-sample training sets as described in Section 3.2.2, thus reducing the tendency to omit transitions associated with brief entry into a prototype's Voronoi region.

4.2.1 Predictor performance

Using the learnt Markov chains $\mathcal{M}_\alpha^{\text{loc}}$ and $\mathcal{M}_\beta^{\text{loc}}$, the experiment described in Section 4.1.5 was performed to assess predictor performance, using test data sets captured soon after the training data. In this experiment 50 stochastic predictions were generated (without perturbation) on each frame to account for the non-deterministic nature of the state-based and behaviour-based predictors. Since the test data was captured primarily to evaluate typicality assessment, it contains a high percentage of 'artificial' behaviours which are initially typical but rapidly become bizarre, and thus provides a rather exacting test of predictor performance.

Figure 4.2 illustrates mean predictor performance over a range $t+T$, $1 \leq T \leq 30$, of future time instants, averaged over all stochastic predictions for all frames in the test sets. As expected, graphs indicate both that the mean performance of all predictors diminishes as the time interval to the prediction increases, and that the linear predictor is generally less powerful, although graphs also iden-

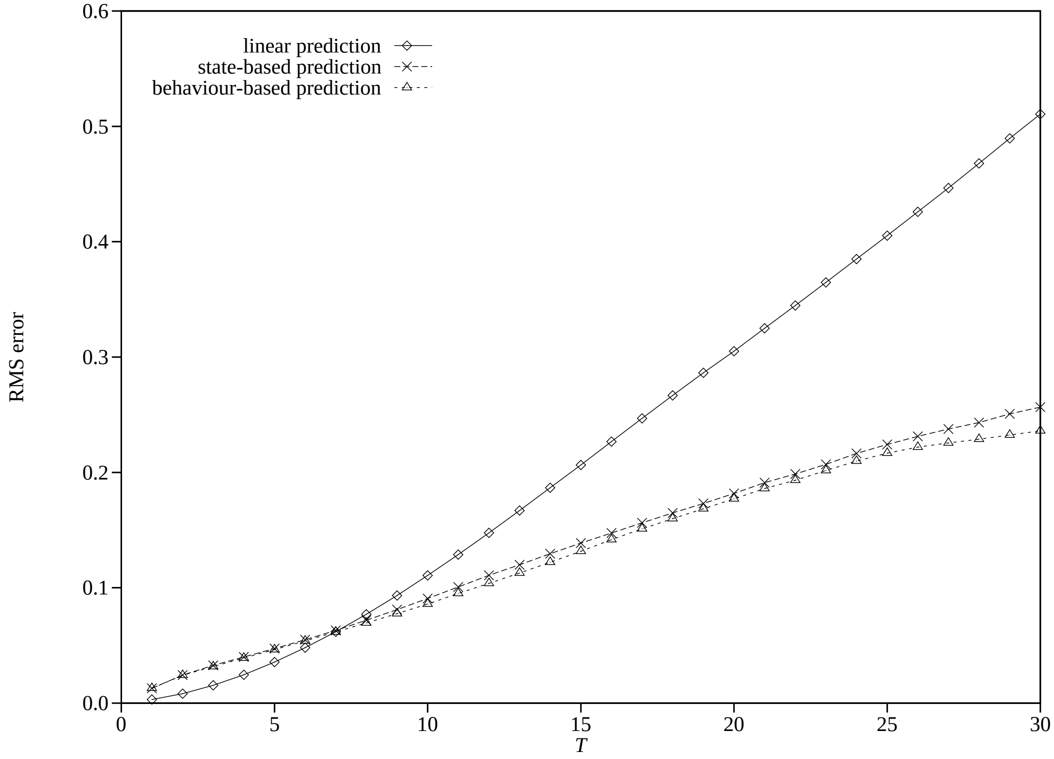


Figure 4.2: Location prediction errors.

tify two rather surprising characteristics.

The first surprising characteristic is that the linear prediction scheme actually has a superior mean performance for predictions of up to about $T = 6$ time instants. This superior performance is probably due to two factors - firstly, the generally locally-linear nature of pedestrian trajectories which will ensure reasonable accuracy for short-term linear predictions, and secondly, the quantization errors which are inherent in the state-based and behaviour-based predictors.

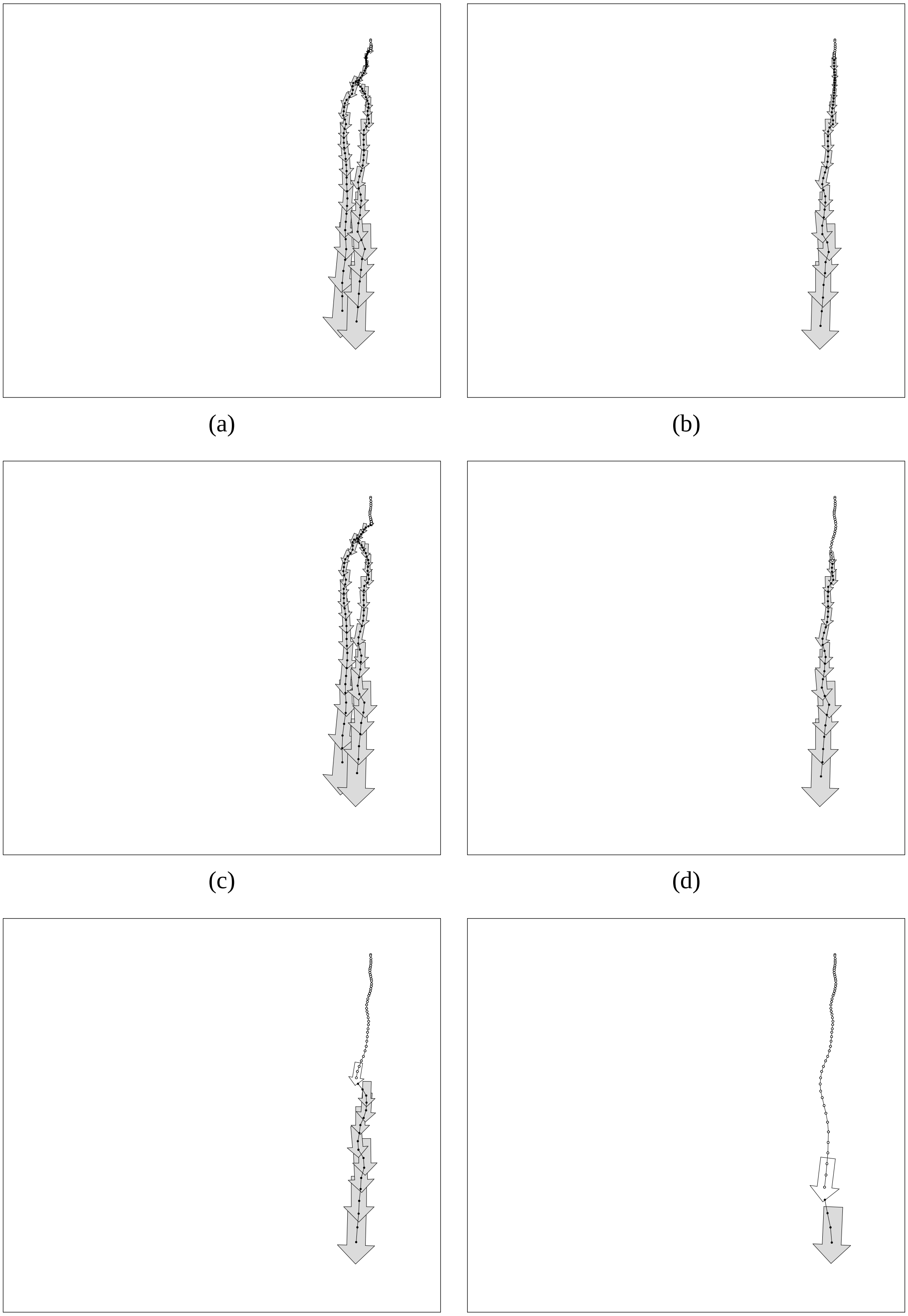
The second surprising characteristic is that the mean performance of the behaviour-based predictor is only marginally superior to that of the state-based predictor, although graphs appear to diverge after about $T = 25$ time instants. The absence of distinctly superior performance in the behaviour-based predictor suggests that the temporal evolution of pedestrian trajectories in the test data has little dependence on past behaviour. Since a similar result is obtained if the experiment is performed on training data, it is reasonable to assume that this is an inherent characteristic of behaviours within this scene.

4.2.2 Maximum likelihood behaviour-based extrapolation

To demonstrate extrapolation, the learnt behaviour-based predictor $\mathcal{M}_{\beta}^{\mathrm{loc}}$ was used to generate maximum likelihood extrapolations during the evolution of three pedestrian trajectories selected from the test data sets. Figure 4.3, Figure 4.4, and Figure 4.5 illustrate extrapolation at selected time instants during each of the three sequences, where all equally maximal probability extrapolations are illustrated and each extrapolation is terminated prematurely if a previously visited chain state is reached, thus avoiding infinite cycles. In each figure, different aspects of the extrapolation process are illustrated as follows:

- The entire current trajectory on which the extrapolation depends is illustrated by a set of small unfilled circles joined with lines.
- The current state vector $\mathbf{F}_t$ which replaces $\bar{\alpha}(e_{i_0})$ is illustrated by an unfilled arrow.
- The sets of state vector tokens generated from each traversal of the chain are illustrated by filled arrows.
- The extrapolations generated by sampling the Hermite interpolants of state vector token sets at regular time instants are illustrated by sets of small filled circles joined with lines.

It is clear from these experimental results that the behaviour-based Markov chain forms an effective encoding of the evolution of spatio-temporal behaviours. Extrapolated trajectories are both spatially and temporally continuous and there is reasonable spatio-temporal continuity where observed behaviour and extrapolations join. Trajectories follow plausible paths through the scene, and temporal characteristics such as the apparent gradual increase or decrease in speed as a pedestrian approaches or retreats relative to the camera are clearly visible, as illustrated in, for example, Figure 4.3(b) and Figure 4.4(b). It is also revealing to observe the changes which occur in the maximum likelihood extrapolations as trajectories progress and alternative future behaviours become more appropriate. In particular, instability is sometimes evident around decision points, causing extrapolations to flit rapidly between alternate possible futures, as illustrated in Figure 4.3(a)–(d) and Figure 4.4(a)–(c)).



(a)

(b)

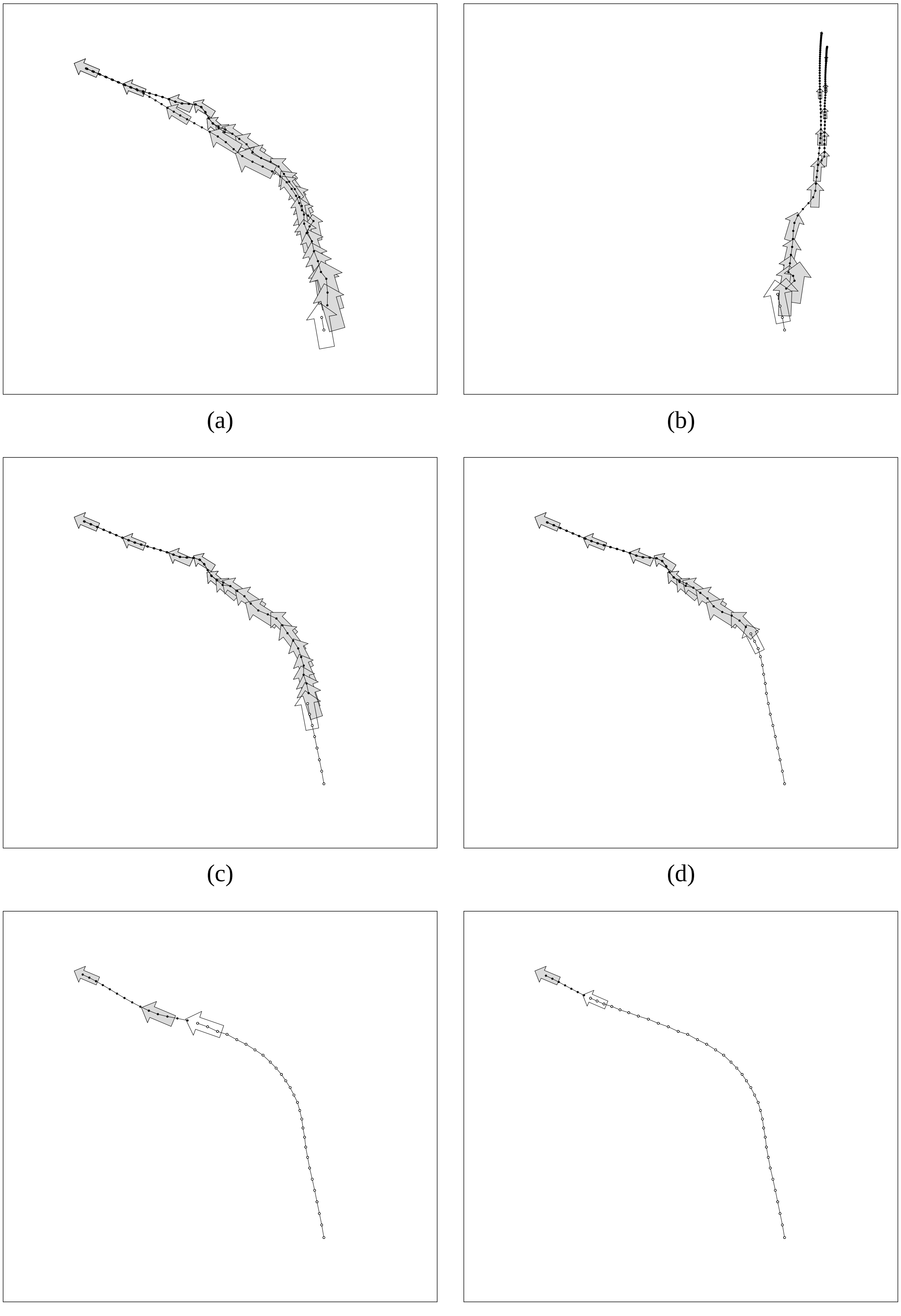
(c)

(d)

(e)

(f)

Figure 4.3: Maximum likelihood location extrapolation - trajectory 1.


(a)

(b)

(c)

(d)

(e)

(f)

Figure 4.4: Maximum likelihood location extrapolation - trajectory 2.

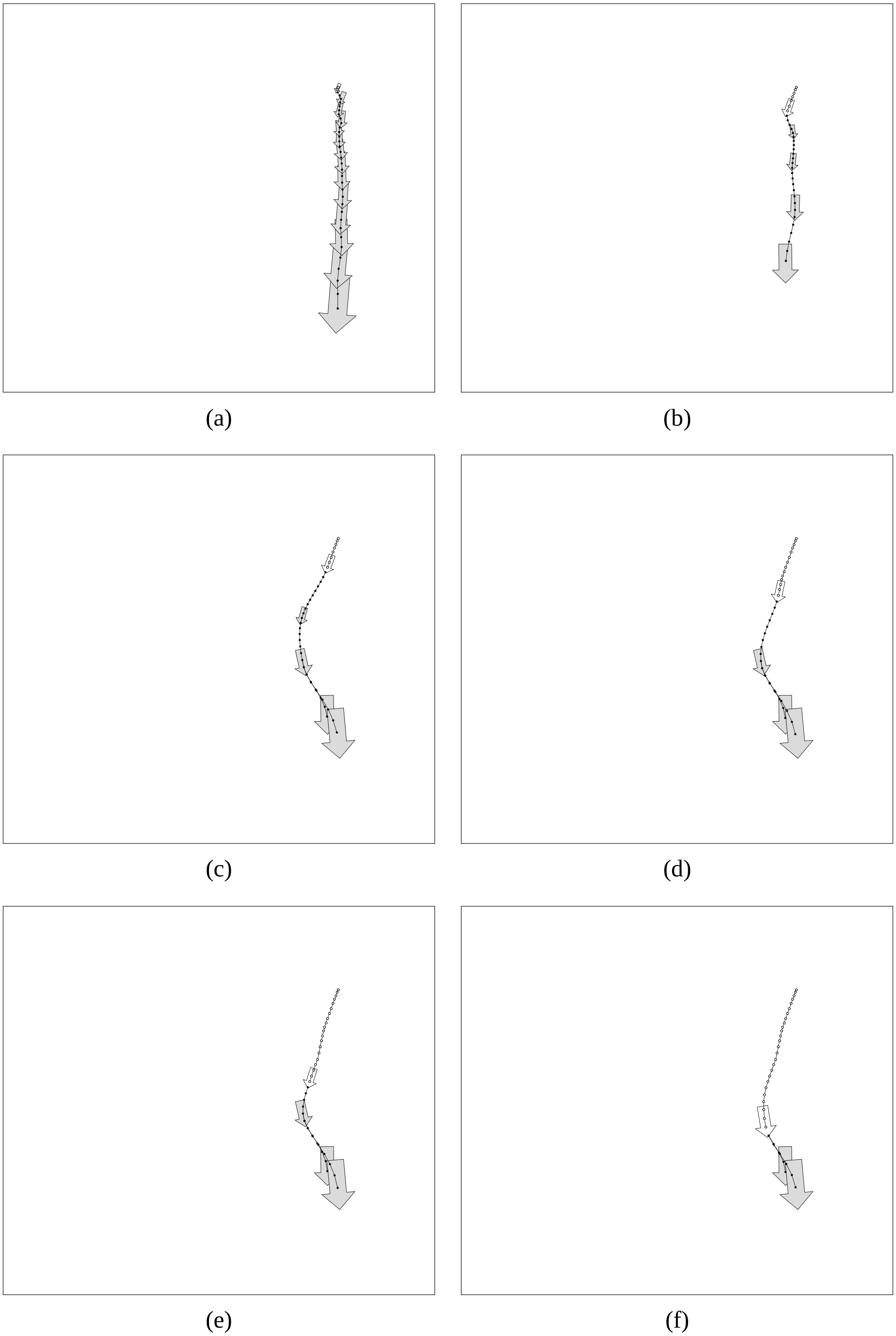


Figure 4.5: Maximum likelihood location extrapolation - trajectory 3.

4.2.3 Stochastic behaviour-based generation

To demonstrate the generation of realistic sample behaviours, stochastic extrapolation and stochastic selection of initial chain states from the learnt behaviour-based predictor $\mathcal{M}_\beta^{\text{loc}}$ were used to produce a set of entirely hypothetical pedestrian trajectories. Figure 4.6 illustrates 504 hypothetical trajectories generated in this way.

Although occasionally rather noisy, sample trajectories are generally both spatially and temporally continuous and exhibit plausible spatio-temporal characteristics. Comparison between the results of this experiment and the training data illustrated in Figure 3.1 suggests that the set of hypothetical sequences forms a plausible random sample of pedestrian behaviour within the scene.

4.3 Experimental results - object shape

A 201-state Markov chain $\mathcal{M}_\alpha^{\text{shape}}$ was superimposed on the set $\mathcal{A}^{\text{shape}}$ of 200 state prototypes generated in the experiment described in Section 3.3.2, using state prototypes as the token set (i.e. $\mathcal{S}_\alpha^{\text{shape}} = \mathcal{A}^{\text{shape}}$). Initial state and state transition distributions were estimated from the single state training set $\mathcal{F}^{\text{shape}}$, disregarding typicality-based transition rejection as the entire sequence is considered to be typical. A value of $\Delta = 0.025$, half that used in Section 3.3.2, was used to re-sample training sets as described in Section 3.2.2, thus reducing the tendency to omit transitions associated with brief entry into a prototype's Voronoi region.

A 401-state Markov chain $\mathcal{M}_\beta^{\text{shape}}$ was superimposed on the set $\mathcal{B}^{\text{shape}}$ of 400 behaviour prototypes generated in the experiment described in Section 3.4.4, estimating the token set $\mathcal{S}_\beta^{\text{shape}}$ during learning as described in Section 4.1.3. Initial state and state transition distributions were estimated from the single behaviour training set $\mathcal{G}^{\text{shape}}$, again disregarding typicality-based transition rejection. A value of $\Delta = 0.0075$, half that used in Section 3.4.4, was used to re-sample training sets as described in Section 3.2.2, thus reducing the tendency to omit transitions associated with brief entry into a prototype's Voronoi region.

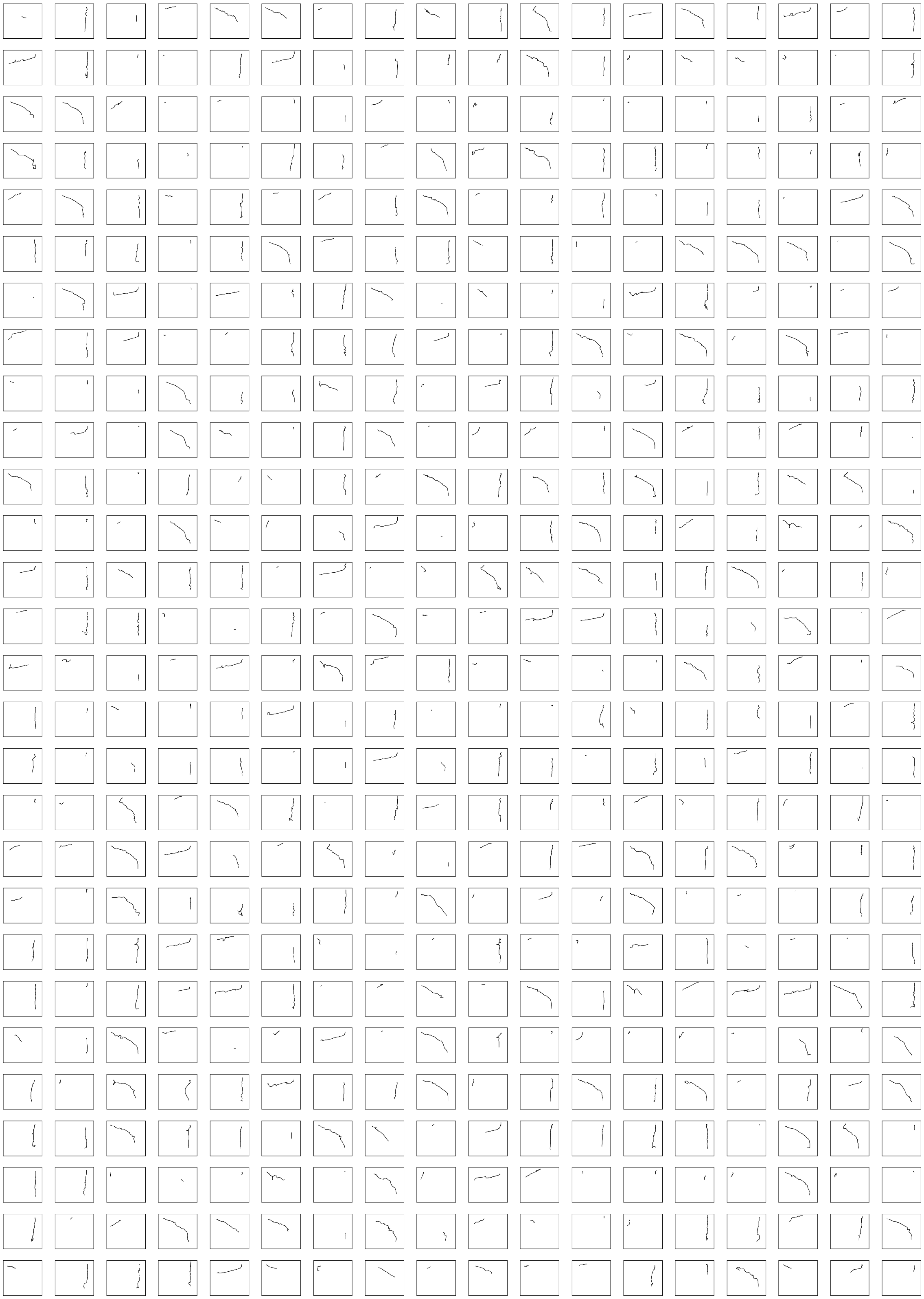


Figure 4.6: Sample pedestrian trajectories generated from the behaviour-based predictor.

4.3.1 Predictor performance

Using the learnt Markov chains $\mathcal{M}_\alpha^{\text{shape}}$ and $\mathcal{M}_\beta^{\text{shape}}$, the experiment described in Section 4.1.5 was performed to assess predictor performance, using training data due to the absence of a test sequence. In this experiment 50 stochastic predictions were generated (without perturbation) on each frame to account for the non-deterministic nature of the state-based and behaviour-based predictors.

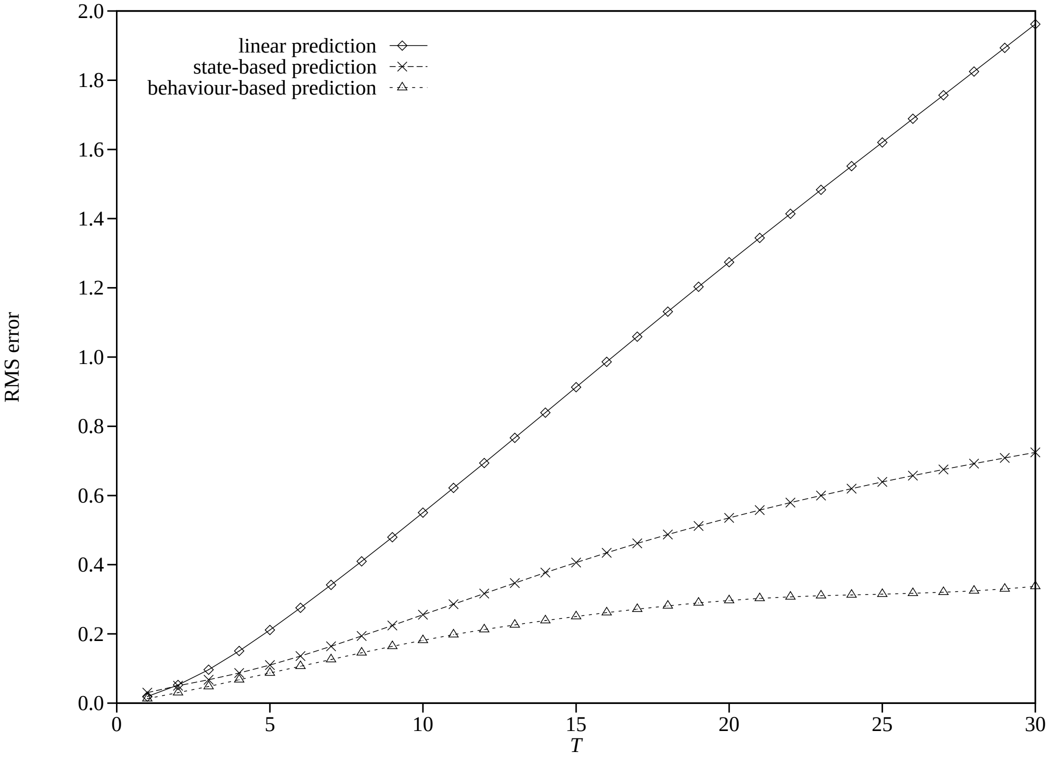


Figure 4.7: Shape prediction errors.

Figure 4.7 illustrates mean predictor performance over a range $t+T$, $1 \leq T \leq 30$, of future time instants, averaged over all stochastic predictions for all frames in the training set. As expected, graphs indicate both that the mean performance of all predictors diminishes as the time interval to the prediction increases, and that the linear predictor is less powerful than the state-based and behaviour-based predictors. Unlike the results obtained in Section 4.2.1, linear prediction of object shape has consistently inferior mean performance due to the highly non-linear nature of shape sequences. Also significant in these results is the markedly superior mean performance of behaviour-based prediction which is indicative of the temporal dependencies inherent in the exercise routine and illustrates the power of the behavioural representation.

4.3.2 Maximum likelihood behaviour-based extrapolation

To demonstrate extrapolation, the learnt behaviour-based predictor $\mathcal{M}_{\beta}^{\text{shape}}$ was used to generate maximum likelihood extrapolations during the evolution of the exercise routine. Figure 4.8 and Figure 4.9 illustrate extrapolation at selected time instants during the sequence, where each extrapolation was chosen randomly from the (generally singleton) set of equally maximal probability extrapolations. In each figure, recent behaviour is illustrated by a set of 12 filled contours, the shade of which indicates recency, the lightest being the current shape. The first 12 frames of each extrapolation are illustrated by a set of unfilled contours overlaying the recent behaviour, the shade of which indicates the progression of behaviour, the lightest being the furthest advanced.

It is clear from these experimental results that the behaviour-based Markov chain forms an effective encoding of the evolution of spatio-temporal behaviours. Extrapolated sequences are both spatially and temporally continuous and there is good spatio-temporal continuity where observed behaviour and extrapolations join. Even the relatively short-term extrapolations illustrated exhibit highly non-linear changes in the positions of B-spline control points, particularly exemplified by extrapolations such as those shown in Figure 4.8(a) and Figure 4.9(q), whilst temporal characteristics such as accelerations and decelerations in arm movements are clearly evident, as illustrated in, for example, Figure 4.8(b) and Figure 4.9(p). These experimental results thus clearly illustrate the utility of behaviour-based models for the representation of complex, non-linear dynamics.

Whilst extrapolations are clearly plausible continuations of recently observed behaviours, comparison with the evolving shape sequence indicates that longer-term temporal dependencies have been encoded within the structure of the Markov chain. At the start of the sequence, illustrated in Figure 4.8(a), and in the transitions between the four exercises, illustrated in Figure 4.8(f), Figure 4.9(j), and Figure 4.9(o), the subsequent exercise is consistently predicted. Further, during each repetition of an exercise or sub-exercise, the subsequent repetition or transition to a new exercise or sub-exercise is also consistently predicted.

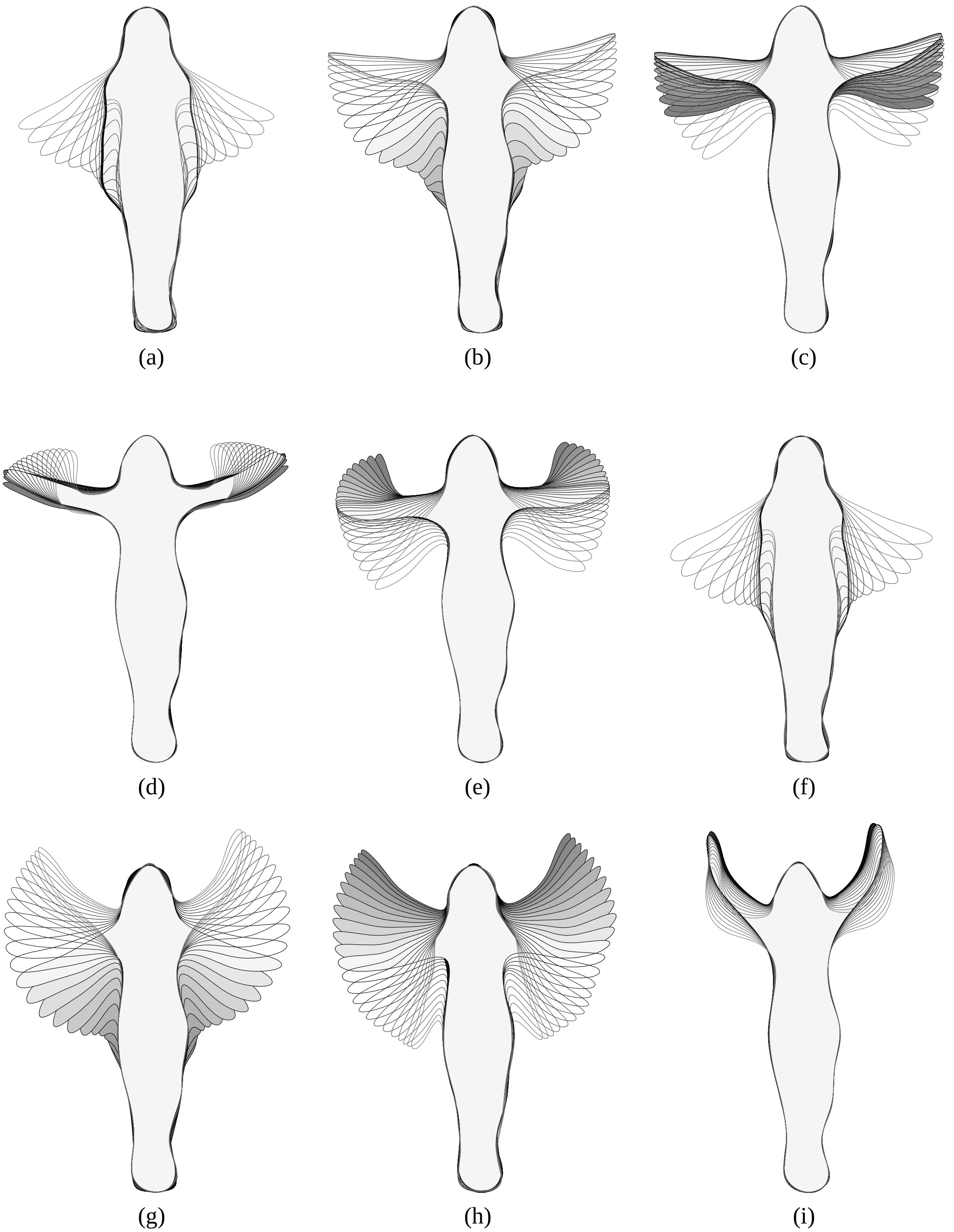


Figure 4.8: Maximum likelihood shape extrapolation (a)–(i).

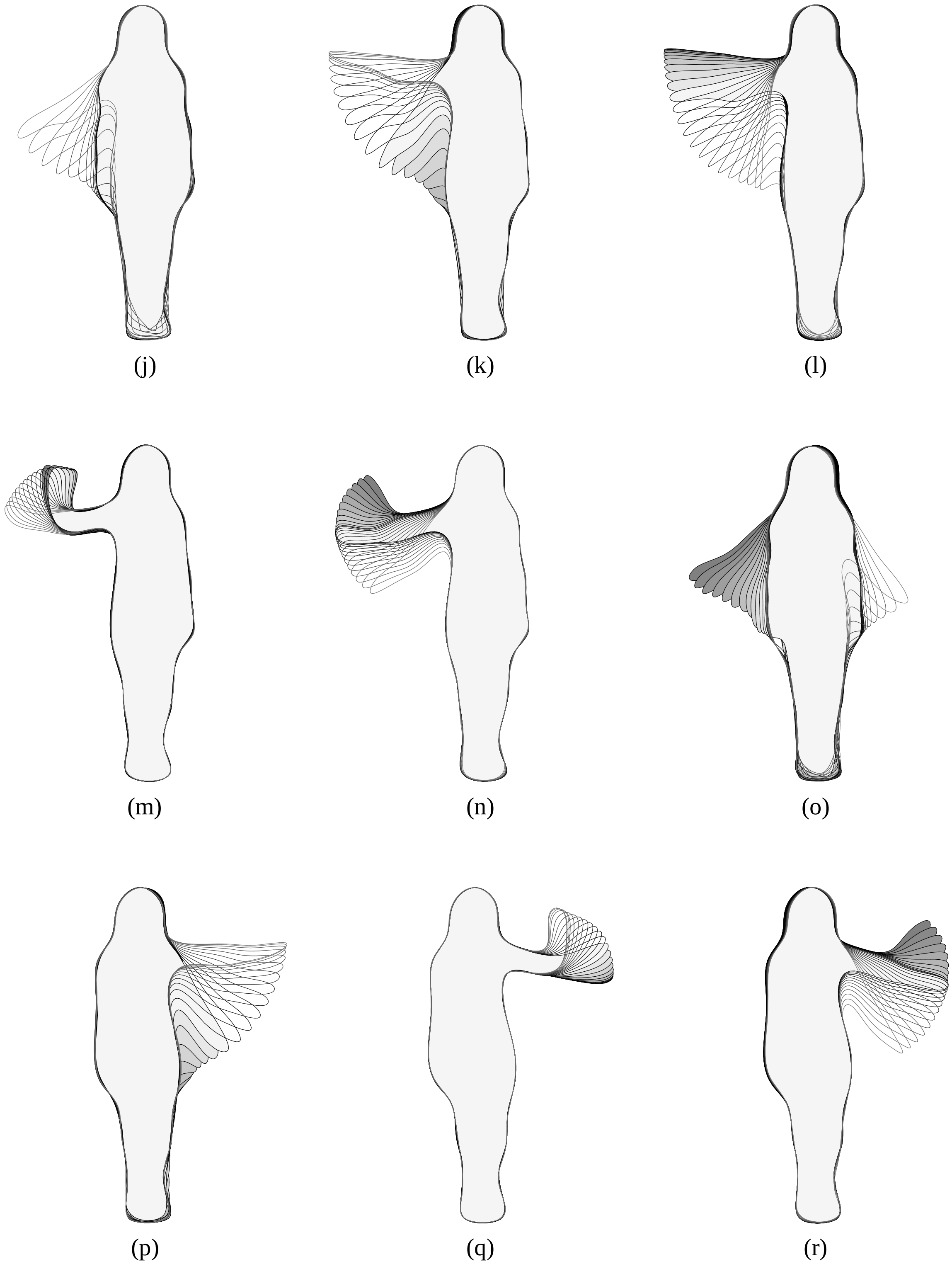


Figure 4.9: Maximum likelihood shape extrapolation (j)–(r).

4.3.3 Stochastic state-based and behaviour-based generation

To demonstrate the generation of realistic sample behaviours, stochastic extrapolation and stochastic selection of initial chain states from both the learnt state-based predictor $\mathcal{M}_{\alpha}^{\text{shape}}$ and the learnt behaviour-based predictor $\mathcal{M}_{\beta}^{\text{shape}}$ were used to produce two entirely hypothetical shape sequences. Figure 4.10 illustrates the sequence generated using the state-based predictor, whilst Figure 4.11 illustrates the sequence generated using the behaviour-based predictor. In each figure, sequences are illustrated by a set of filled contours. Sequences start at the top-left corner of each figure, and progress in a top-to-bottom, left-to-right order. Vertical layering of contours has been used in these illustrations since it produces less occlusion of the arms than a horizontal layering.

Although the sequence illustrated in Figure 4.10 is both spatially and temporally continuous, it identifies a number of weaknesses in the state-based predictor. Throughout the sequence, small aberrations are evident (indicated by circles) which suggest instability in prediction due to spatio-temporal ambiguities, whilst excessive periods of static behaviour demonstrate the state-based predictor's failure to encode the temporal extent of approximately static behaviours. The absence of any encoding of longer-term temporal dependencies is clearly illustrated by the apparently random order in which the separate exercises and sub-exercises are generated, and by the failure to generate the correct number of repetitions of each exercise and sub-exercise.

The sequence illustrated in Figure 4.11 is both spatially and temporally continuous and clearly illustrates the superior performance of the behaviour-based predictor. None of the aberrations or excessive periods of static behaviour which were evident in the state-based sequence are present, thus suggesting more stability in prediction and the ability to encode the temporal extent of approximately static behaviours. The encoding of longer-term temporal dependencies is clearly illustrated by the correct progression from one exercise or sub-exercis e to the next, and by the generation of the correct number of repetitions of each exercise and sub-exercise. The encoding of these longer-term temporal dependencies also clearly indicates that the spatio-temporal behaviour representation developed in Section 3.4.1 is not restricted to the representation of simple, non-repeating sequences.

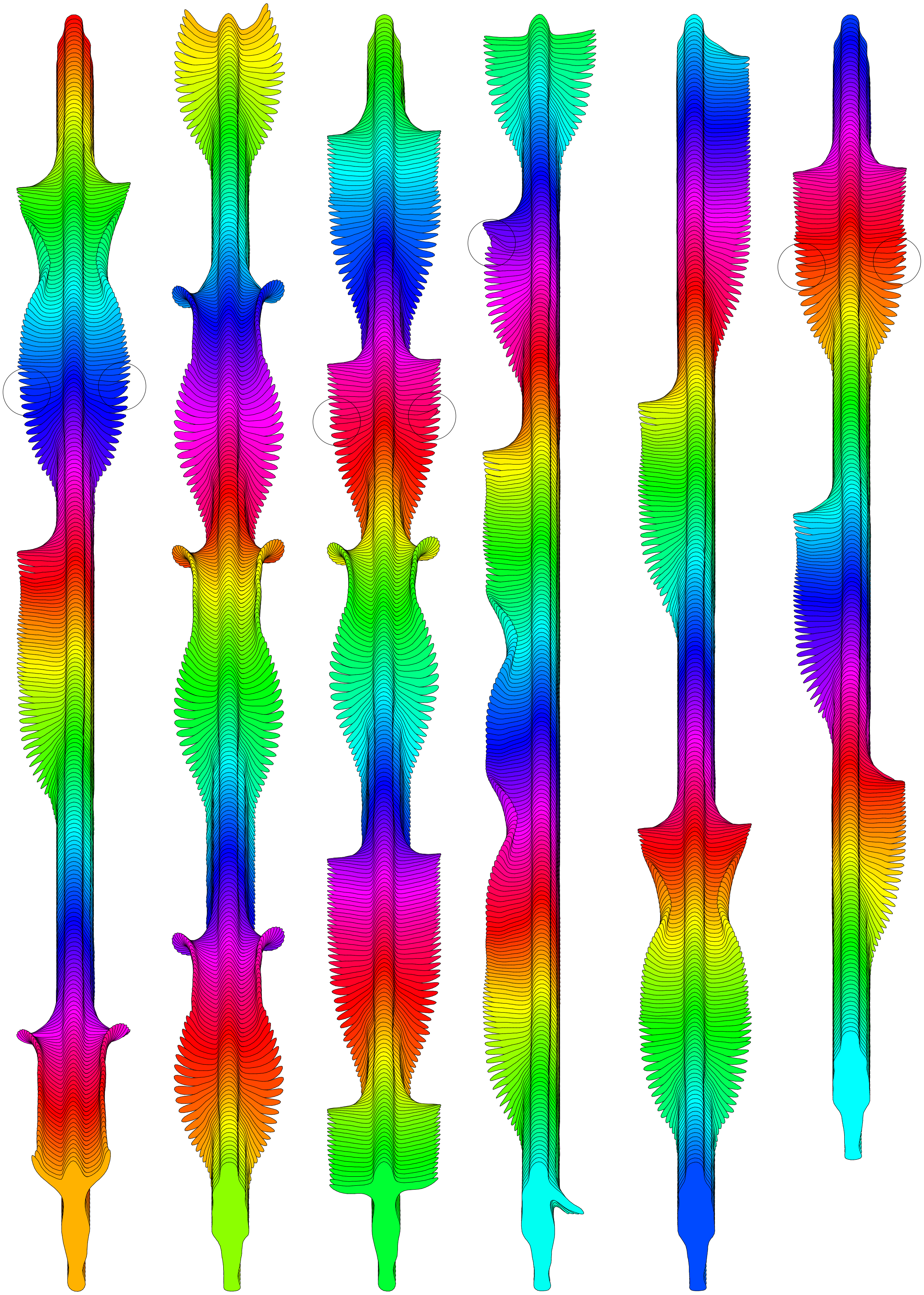


Figure 4.10: Sample shape sequence generated from the state-based predictor.

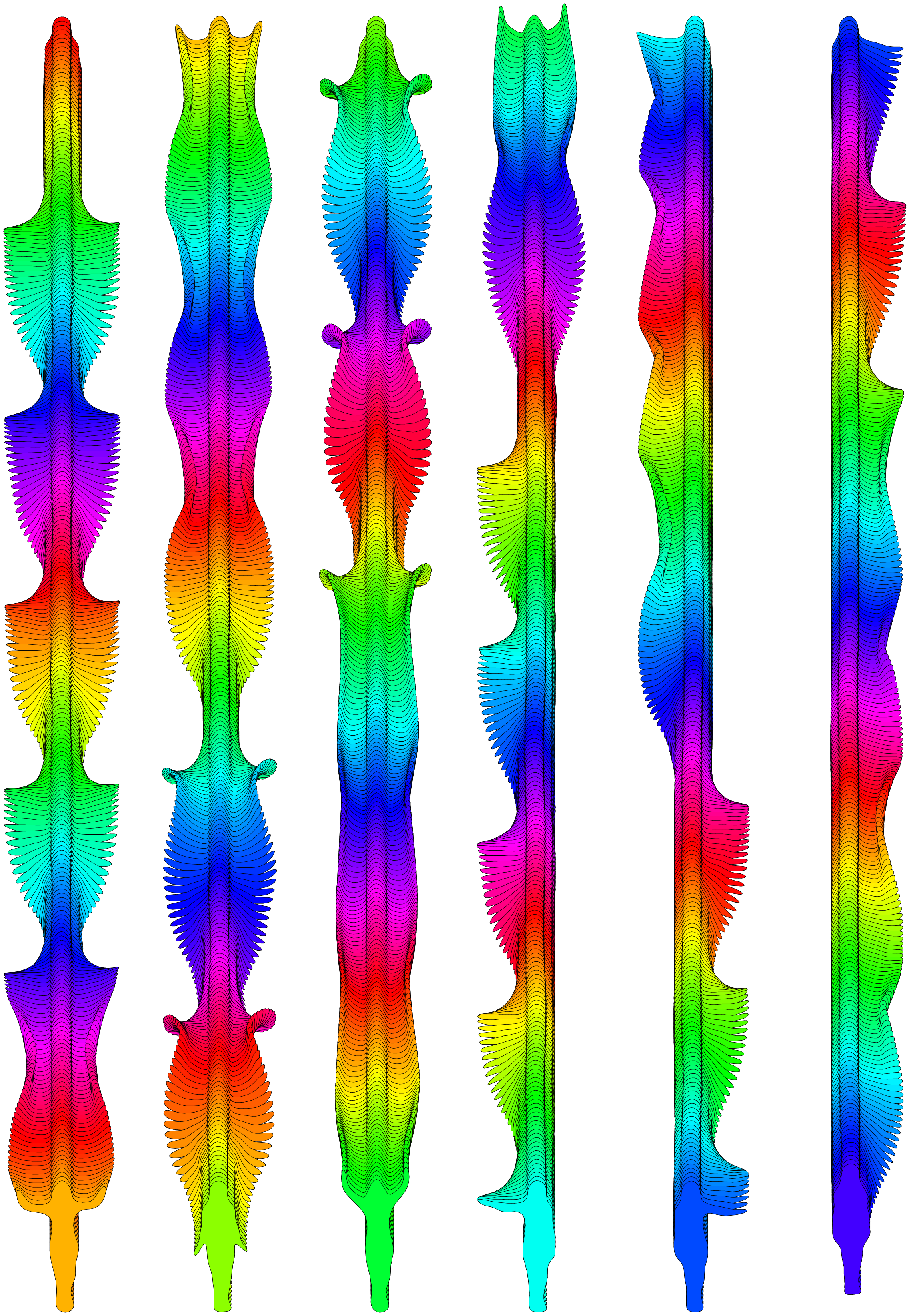


Figure 4.11: Sample shape sequence generated from the behaviour-based predictor.

4.4 Discussion

In this chapter, techniques have been developed which allow the models of characteristic object states and behaviours developed in Chapter 3 to be enhanced to include generative capabilities via the superimposition of Markov chains, the parameters of which are acquired automatically during a further learning phase. Experimental results presented for two object characteristics with distinctly different properties clearly illustrate the utility of enhanced models for the generation of predictions, extrapolations, and realistic sample behaviours, and demonstrate the advantages of behaviour-based predictors in which temporal dependencies are encoded within the transition structure of the Markov chain.

The generative capabilities demonstrated within this chapter could clearly be exploited to increase the robustness and efficiency of object tracking systems, and would be particularly effective if used within a stochastic tracking algorithm such as Isard and Blake's CONDENSATION [46], where behaviour-based predictors with learnt noise models would provide a powerful stochastic prediction mechanism. In addition, such generative capabilities could be exploited to enhance reasoning during partial occlusions, and, since extrapolations remain plausible and reasonably accurate over extended periods of time, could be exploited to facilitate tracking over prolonged periods of complete occlusion, such as when a pedestrian walks behind a large vehicle.

The generation of entirely hypothetical sequences from learnt behaviour models could provide a powerful mechanism for the automatic generation of realistic object behaviours within animations, virtual worlds, or computer generated film sequences. In addition to the generation of isolated characteristics, models describing a number of different characteristics, such as pedestrian location, shape, and texture, could be probabilistically *coupled*, as described by Brand *et al.* [14], thus allowing realistic behaviours of entire objects to be generated.

Although not demonstrated within this thesis, the transition structure of behaviour-based predictors can be exploited to yield details of the regularities inherent in certain behaviours. For instance, through careful choice of the decay coefficient used in temporal pattern formation, the cyclic nature of behaviours such as the experimental shape sequence is replicated in the transition structure of the corresponding behaviour-based predictor. Such structural information may be effective in addressing problems such as the segmentation of gestures.

4.4.1 A comparison with Hidden Markov Modelling

Hidden Markov Models (HMMs) are a popular mechanism for describing the temporal structure of time-varying processes. These models have been extensively used for continuous speech recognition tasks (see, for example, Huang *et al.* [45]) and have recently become popular for describing the temporal structure of actions and gestures (see Section 2.3). HMMs are an extension to the Markov chain process description in which each state has an associated discrete or continuous observation distribution which governs the production of observation tokens, and are thus doubly stochastic processes in which the underlying stochastic process (the Markov chain) is hidden.

As described within the extensive body of HMM literature (see Rabiner and Juang [68] or, for example, Huang *et al.* [45]), there are three key problems in HMM use, commonly referred to as the *estimation*, *evaluation*, and *decoding* problems. Given an instance of an HMM and a number of sequences of observation tokens, estimation describes the process of adjusting model parameters to maximise the conditional probability of observing the training sequences given a particular model. Since an analytic solution to this training problem is not known, iterative optimisation techniques must be used, typically Baum-Welch re-estimation. Having trained an HMM, evaluation describes the process of calculating the probability of a particular sequence of observation tokens, and is achieved using the Forward-Backward algorithm. Finally, given a sequence of observation tokens, decoding describes the process of finding a corresponding state sequence which is in some sense optimal. If maximisation of state sequence probability is used as an optimality criterion, then decoding is achieved using the Viterbi algorithm.

Whilst the enhanced Markov chains developed in Section 4.1.4 for stochastic behaviour perturbation are equivalent to Hidden Markov Models with noise models defining the observation distributions, attempts to directly acquire such models using iterative optimisation techniques are unlikely to succeed. As widely reported within the HMM literature, local optima are frequently encountered by iterative optimisation techniques when learning HMMs with many free parameters, and thus model topology and size are often highly constrained prior to training (see Section 2.3). For example, Yamato *et al.* [93] report the existence of local optima when using even small 36-state HMMs with unconstrained topology to model individual tennis swings. Thus, due to the very large number of free parameters, it is highly unlikely that iterative optimisation techniques would yield near-optimal models such as the behaviour-based predictors in which temporal dependencies are encoded within the transition structure. In addition to enabling the acquisition of such large and near-optimal HMMs, the behaviour modelling approach presented within this thesis provides significantly more efficient mechanisms for typicality assessment and behaviour recognition than those provided by the Forward-Backward and Viterbi algorithms.

4.4.2 Temporal adaptation

As stated in Chapter 1, a natural process for the perception of powerful behaviour models should allow gradual temporal adaptation, enabling model evolution with occasional changes in characteristic behaviour. In Section 3.6.2, extended learning and the adjustment of prototype typicality values was proposed as a mechanism through which the temporal adaptation of models of characteristic object states and behaviours could be achieved. Since models have been enhanced to include generative capabilities, temporal adaptation of enhanced models would also require the adjustment of both Markov chain distributions and noise model parameters during extended learning, using either iterative update equations or moving temporal windows.

Chapter 5

Object interaction

Throughout the development of the behaviour modelling framework, it is the behaviours of single objects which have been considered. To extend the utility of this framework, the modelling of object interaction is also investigated. Object interaction is a particularly interesting form of behaviour since it allows reasoning to be extended from individuals to groups of objects, whilst providing a machine with the ability to learn and use models of natural interaction may prove beneficial to the provision of natural user-machine interaction. This chapter describes two approaches to binary interaction modelling using the models developed in Chapter 3 and Chapter 4. The first approach considers the statistical co-occurrence of events within models of the state or behaviour of individual objects, whilst the second approach attempts to explicitly model interaction as joint behaviour. This latter approach is used within a stochastic tracking algorithm to demonstrate how a learnt joint behaviour model can be used to equip a virtual object with the ability to interact in a natural way.

5.1 State and behaviour co-occurrence

The discrete nature of the models developed in Chapter 3 allows interaction typicality to be assessed by considering the co-occurrence of events within models of the state or behaviour of individual objects. Within this scheme, an event represents the activation of a particular state or behaviour prototype by one of a set of concurrent objects, where all pairs of concurrent objects are considered to be interacting, regardless of proximity or other cues.

Event co-occurrence in a state or behaviour model with k prototypes is modelled by a $k \times k$ symmetric *co-occurrence matrix*

$$\mathcal{C} = \begin{bmatrix} \mathcal{C}_{1,1} & \cdots & \mathcal{C}_{1,k} \\ \vdots & \ddots & \vdots \\ \mathcal{C}_{k,1} & \cdots & \mathcal{C}_{k,k} \end{bmatrix}, \ \mathcal{C}_{i,j} = \mathcal{C}_{j,i}, \ \sum_{i=1}^{k}\sum_{j=1}^{k} \mathcal{C}_{i,j} = 1, \tag{5.1}$$

where the probabilities $\mathcal{C}_{i,j}$ are estimated during a further learning phase by observing the relative frequency with which each combination of state or behaviour events occurs over synchronised training sets.

The probability of an interaction which causes the co-occurrence of events A and B, corresponding to the activation by concurrent objects of prototypes $\bar{\alpha}_a$ and $\bar{\alpha}_b$ or prototypes $\bar{\beta}_a$ and $\bar{\beta}_b$, is thus given by

$$\mathrm{P}(A \cap B) = \mathcal{C}_{a,b}. \tag{5.2}$$

Since the probability of a single event occurring is given by

$$\mathrm{P}(A) = \sum_{j=1}^{k} \mathcal{C}_{a,j}, \tag{5.3}$$

the conditional probability

$$\begin{aligned} \mathrm{P}(A|B) &= \frac{\mathrm{P}(A \cap B)}{\mathrm{P}(B)} \\ &= \frac{\mathcal{C}_{a,b}}{\sum_{j=1}^{k} \mathcal{C}_{b,j}} \end{aligned} \tag{5.4}$$

can also be evaluated. Assuming good density matching has been achieved, each prototype will represent an approximately equal amount of probability, and thus $\mathrm{P}(A) \approx \frac{1}{k}$ and $\mathrm{P}(A|B) \approx k\mathcal{C}_{a,b}$.

5.1.1 State and behaviour dependence

In addition to the evaluation of event co-occurrence and conditional event occurrence probabilities, a measure of the extent to which events A and B are statistically dependent can also be derived from the co-occurrence matrix:

$$d(A,\, B) \quad = \quad \frac{\mathrm{P}(A \cap B)}{\mathrm{P}(A)\mathrm{P}(B)}$$

$$= \frac{\mathcal{C}_{a,b}}{\sum_{j=1}^{k} \mathcal{C}_{a,j} \; \sum_{j=1}^{k} \mathcal{C}_{b,j}}, \tag{5.5}$$

where, assuming good density matching has been achieved, $d(A,\ B) \approx k^2 \mathcal{C}_{a,b}$.

As illustrated in Figure 5.1, the dependence measure $d(A,\ B)$ can be used to classify interactions into four distinct classes[1], of which the two fundamentally different classes of dependent events are of particular interest. If events are negatively dependent, then the occurrence of one event reduces the probability that the other event has occurred, thus implying weak mutual exclusivity. If, however, events are positively dependent, then the occurrence of one event increases the probability that the other event has occurred, thus implying reliance.

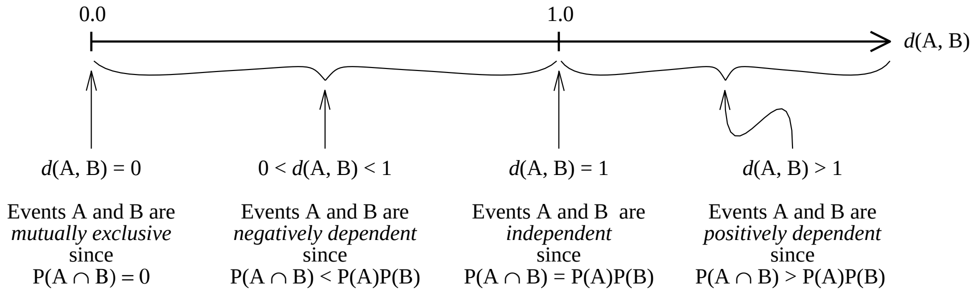


Figure 5.1: Interaction classification through event dependence.

The classification illustrated in Figure 5.1 provides a useful mechanism for filtering interactions and for providing attentional control. For instance, the co-occurrence of mutually exclusive and negatively dependent events is probably indicative of unusual behaviour which may merit further investigation, whilst the co-occurrence of positively dependent events is probably indicative of genuinely interactive behaviours which can thus be selected for further analysis.

5.2 Modelling joint behaviour

The techniques developed within Section 5.1 facilitate both the assessment of interaction typicality and the identification of genuinely interactive behaviours. In addition, event co-occurrence could be used to assess the probability of concurrent predictions from models of the state or behaviour of individual objects, thus providing a mechanism for the extrapolation of future interactive be-

[1] In practice, the classification of mutually exclusive and independent events must tolerate a small margin of error.

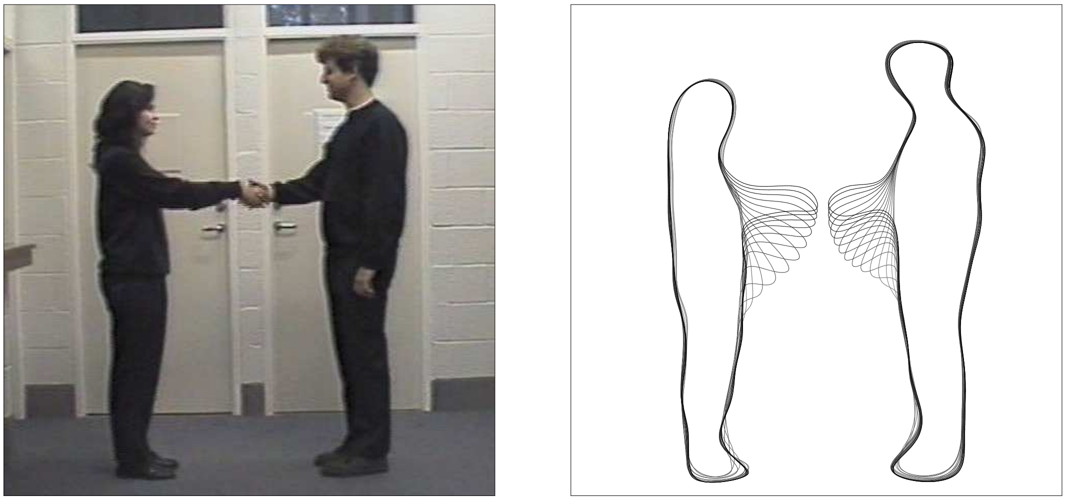
haviours. Unfortunately, event co-occurrence models are often inadequate since they provide a representation of interactive behaviours in which the level of detail is proportional to the typicality of the *individual* behaviours, and not the typicality of the interaction.

A more effective representation of binary interactive behaviours can be achieved by considering the joint (combined) behaviour of pairs of interacting objects, using the behaviour modelling framework developed in Chapter 3 and Chapter 4 to provide a detailed representation which is both analytic and generative. Within this scheme, candidate interactions could be filtered to yield a set of genuinely interactive joint behaviours for training, either using proximity cues or using event co-occurrence to select interactions involving positively dependent events. In the experiments described in this chapter, however, image sequences have been selected by hand, thus negating the need for such filtering.

5.2.1 Joint behaviour representation

When modelling joint behaviour, raw interaction data consisting of ordered sets of characteristic vectors must encode the evolving characteristics associated with both interacting objects. Since the location within a scene at which an interactive behaviour occurs is probably of less relevance than the interaction itself, non-scene-specific representations are probably appropriate. In addition, many interactions are typified not only by the evolution of a particular object characteristic, but also by the evolving spatial relationships between interacting objects, and thus such relationships must also be encoded within characteristic vectors.

For the experiments described in this chapter, a relatively common human interaction has been used - that of shaking hands. Since a typical handshake is a rather brief and relatively simple interaction, experiments are based on individuals performing exaggerated handshakes which comprise a varying number of ‘shakes’, thus introducing a cyclic component within the behaviour, whilst the observation of multiple handshake sequences introduces variation in handshake style. In these experiments, individuals are viewed such that their interaction can be described in terms of the shape of the left-hand and right-hand individuals together with their separation and relative size. Shape data is generated using the silhouette extraction method described in Section 3.1.2, and thus individuals were tracked in uncluttered indoor scenes wearing dark clothing. Figure 5.2 shows two individuals performing an exaggerated handshake, (a), and a number of smoothed shapes from a sequence representing their interaction, (b).



(a) (b)

Figure 5.2: Sample interaction data: (a) handshake scene, and (b) some shapes from the handshake sequence.

The evolution of a handshake interaction is thus represented by an ordered set of characteristic vectors $\mathbf{C} \in [0,\,1]^{4n+2}$:

$$\mathcal{C} = \{\mathbf{C}_0,\, \mathbf{C}_1,\, \ldots,\, \mathbf{C}_m\}\,, \tag{5.6}$$

where

$$\mathbf{C}_t = \left(\mathbf{S}^{\mathrm{L}}(t),\, \mathbf{S}^{\mathrm{R}}(t),\, d(t),\, s(t)\right). \tag{5.7}$$

$\mathbf{S}^{\mathrm{L}}$ and $\mathbf{S}^{\mathrm{R}}$ are normalised shape vectors describing the silhouette boundaries of the left-hand and right-hand individuals, transformed into actor centred coordinates and scaled by their respective heights to enable the integration of data from different sequences whilst ensuring that all components lie approximately in the interval $[0,\,1]$:

$$\mathbf{S}(t) = (x_1(t),\, y_1(t),\, x_2(t),\, y_2(t),\, \ldots,\, x_n(t),\, y_n(t)), \tag{5.8}$$

where

$$x_n(t) \;=\; \frac{x'_n(t) - X(t)}{h(t)} + \frac{1}{2}, \tag{5.9}$$

$$y_n(t) \;=\; \frac{y'_n(t) - Y(t)}{h(t)} + 1, \tag{5.10}$$

(x_i', y_i') are spline control points and h is the height of an individual's silhouette boundary, both provided by the tracker in image plane coordinates, and $(X = x_1', Y = y_1')$ is the spline reference point defining the individual's position.

Finally, d and s are components describing relative horizontal actor separation and relative actor scale, defined as follows:

$$d(t) = \frac{X^{\mathrm{R}}(t) - X^{\mathrm{L}}(t)}{h^{\mathrm{L}}(t)}, \tag{5.11}$$

$$s(t) = \frac{h^{\mathrm{R}}(t)}{h^{\mathrm{L}}(t)}. \tag{5.12}$$

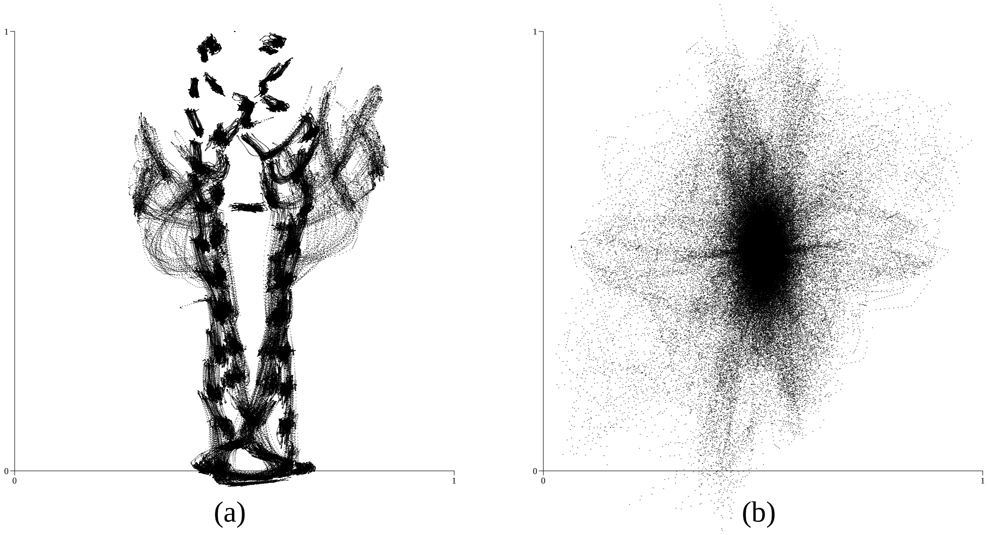
5.2.2 Learning joint behaviour models

Having developed a characteristic vector representation describing the evolution of the joint behaviour of interacting individuals, powerful models of interactive behaviour can be acquired from observation using the framework developed in Chapter 3 and Chapter 4.

5.2.2.1 Experimental results - learning state models

State training sets $\mathcal{F}_j^{\mathrm{int}}$ were generated from the 13 smoothed, sub-sampled handshake sequences, one of which is partially illustrated in Figure 5.2(b). Sub-sampling of the 130-dimensional characteristic vectors $\mathbf{C}_t$ (describing a pair of 32 control point B-splines together with their separation and relative size) was performed at $0.04s$ intervals and high frequency noise was minimised by smoothing vectors over a moving window of width $w = 5$. 260-dimensional state vectors $\mathbf{F}_t$ were generated using a scaling factor $\lambda = 10$ to scale differential components, and ordered data sets were further re-sampled to improve density representation using a constant separation $\Delta = 0.1$. After pre-processing, training sets $\mathcal{F}_j^{\mathrm{int}}$ comprised a total of 4,407 state vectors lying approximately within a unit hypercube. Figure 5.3 shows scatter plots of this training data projected onto both the $(x_i^{\mathrm{L}}, y_i^{\mathrm{L}})$, $(x_i^{\mathrm{R}}, y_i^{\mathrm{R}})$, and (d, s) planes, (a), and the $(\lambda\dot{x}_i^{\mathrm{L}} + \frac{1}{2}, \lambda\dot{y}_i^{\mathrm{L}} + \frac{1}{2})$, $(\lambda\dot{x}_i^{\mathrm{R}} + \frac{1}{2}, \lambda\dot{y}_i^{\mathrm{R}} + \frac{1}{2})$, and $(\lambda\dot{d} + \frac{1}{2}, \lambda\dot{s} + \frac{1}{2})$ planes, (b).

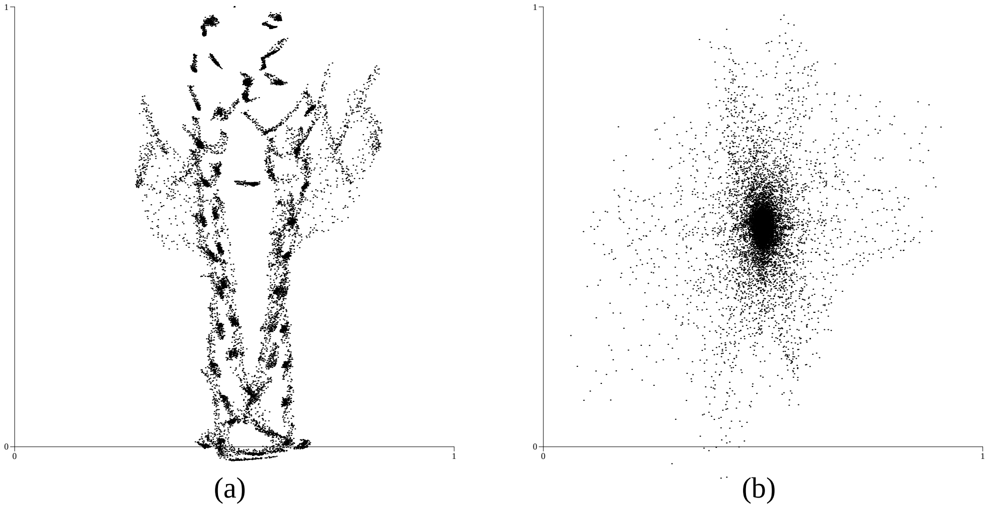
A set $\mathcal{A}^{\mathrm{int}}$ of 200 state prototypes was learnt from 2,000,000 iterations of AVQ over state vectors from the training sets $\mathcal{F}_j^{\mathrm{int}}$, using a constant $\beta = 0.01$ for sensitivity adjustments. Figure 5.4 shows scatter plots of the resulting state prototypes projected onto both the $(x_i^{\mathrm{L}},\, y_i^{\mathrm{L}})$, $(x_i^{\mathrm{R}},\, y_i^{\mathrm{R}})$, and $(d,\, s)$ planes, (a), and the $(\lambda\dot{x}_i^{\mathrm{L}}+\frac{1}{2},\, \lambda\dot{y}_i^{\mathrm{L}}+\frac{1}{2})$, $(\lambda\dot{x}_i^{\mathrm{R}}+\frac{1}{2},\, \lambda\dot{y}_i^{\mathrm{R}}+\frac{1}{2})$, and $(\lambda\dot{d}+\frac{1}{2},\, \lambda\dot{s}+\frac{1}{2})$ planes, (b). Comparison with the scatter plots of training data clearly shows the results to be plausible and suggests that reasonable density matching is achieved.



(a)

(b)

Figure 5.3: State vector distribution - object interaction: (a) projection onto the position (and spatial relationship) planes, and (b) projection onto the first derivative planes.



(a)

(b)

Figure 5.4: State prototype distribution - object interaction: (a) projection onto the position (and spatial relationship) planes, and (b) projection onto the first derivative planes.

In Figure 5.5, each of the 200 state prototypes is illustrated by two overlapping pairs of silhouettes, the upper splines representing the prototype's $(x_i^{\mathrm{L}},\, y_i^{\mathrm{L}})$, $(x_i^{\mathrm{R}},\, y_i^{\mathrm{R}})$, and $(d,\, s)$ components whilst the lower splines have been generated by subtracting the prototype's $(\dot{x}_i^{\mathrm{L}},\, \dot{y}_i^{\mathrm{L}})$, $(\dot{x}_i^{\mathrm{R}},\, \dot{y}_i^{\mathrm{R}})$, and $(\dot{d},\, \dot{s})$

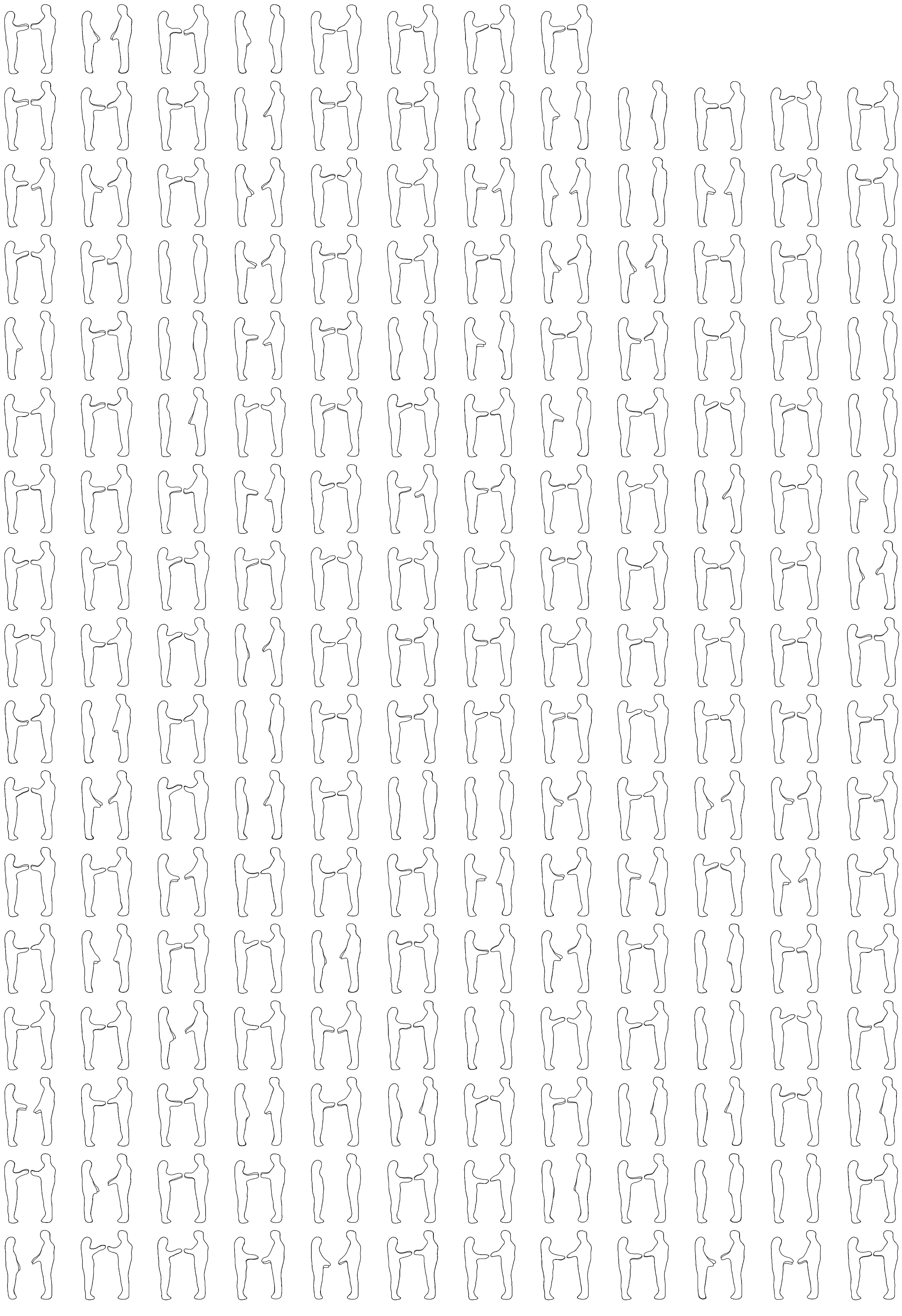


Figure 5.5: Learnt state prototypes - object interaction.

values from the corresponding $(x_i^{\mathrm{L}}, y_i^{\mathrm{L}})$, $(x_i^{\mathrm{R}}, y_i^{\mathrm{R}})$, and (d, s) components. It is clear from this representation that prototypes lie in the desired areas of the state space.

Figure 5.6 shows a frequency histogram illustrating density matching for the 200 state prototypes and 4,407 state training vectors used in this experiment. The mean of this approximately normal distribution is around 22, which is consistent with the expected value of 22.035, whilst the width of the distribution suggests some inaccuracy in density matching.

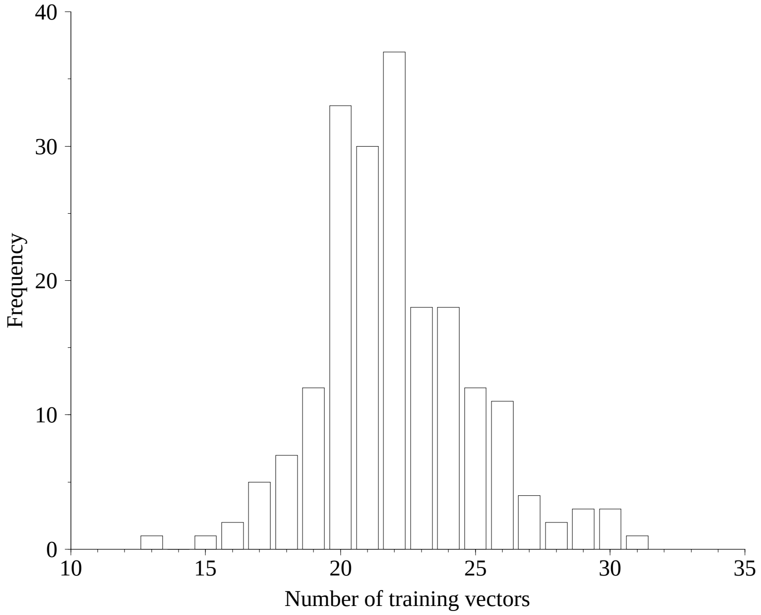


Figure 5.6: Frequency histogram illustrating state prototype density matching - object interaction.

5.2.2.2 Experimental results - learning behaviour models

Behaviour training sets $\mathcal{G}_j^{\mathrm{int}}$ were generated from the 13 unmodified state data sets $\mathcal{F}_j^{\mathrm{int}}$ and the set $\mathcal{A}^{\mathrm{int}}$ of 200 state prototypes generated in the experiment described in Section 5.2.2.1. The pre-processing of raw sequences was performed using the parameter values given in Section 5.2.2.1, and 200-dimensional behaviour vectors $\mathbf{G}_t$ were generated using a scaling factor $\rho = 4.3$ to scale proximity values and a decay coefficient $\gamma = 0.99$. γ was chosen to give a relatively fast decay rate relative to the length of each sequence, thus avoiding behaviour component saturation during repeated ‘shakes’. Ordered data sets were further re-sampled to improve density representation using a constant separation $\Delta = 0.06$. After pre-processing, training sets $\mathcal{G}_j^{\mathrm{int}}$ comprised a total of

4,973 behaviour vectors lying approximately within a unit hypercube.

A set $\mathcal{B}^{\text{int}}$ of 400 behaviour prototypes was learnt from 2,000,000 iterations of AVQ over behaviour vectors from the training sets $\mathcal{G}_j^{\text{int}}$. A constant $\beta = 0.01$ was used for sensitivity adjustments in the AVQ algorithm together with the two-stage cooling schedule described in Section 3.2.1. Figure 5.7 shows a frequency histogram illustrating density matching for the 400 behaviour prototypes and 4,973 training vectors used in this experiment. The mean of this approximately normal distribution is around 12 which is consistent with the expected value of 12.4325, whilst the width of the distribution suggests little inaccuracy in density matching.

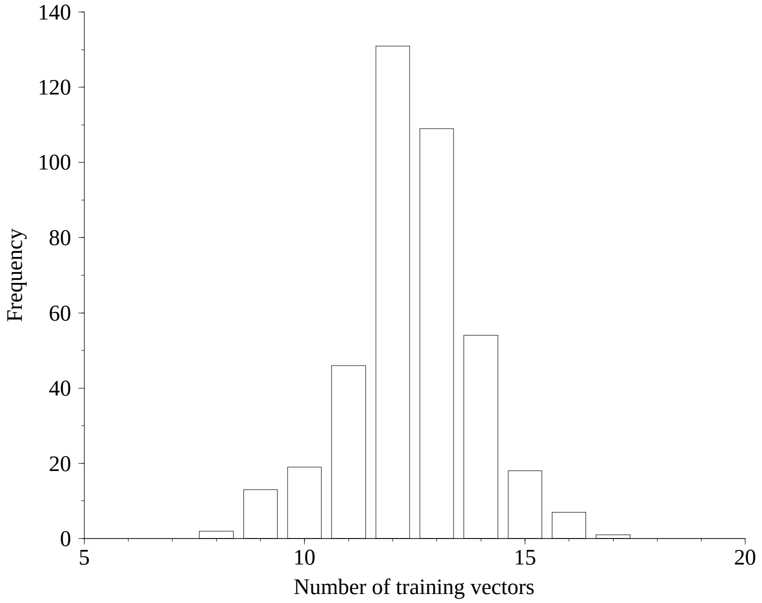
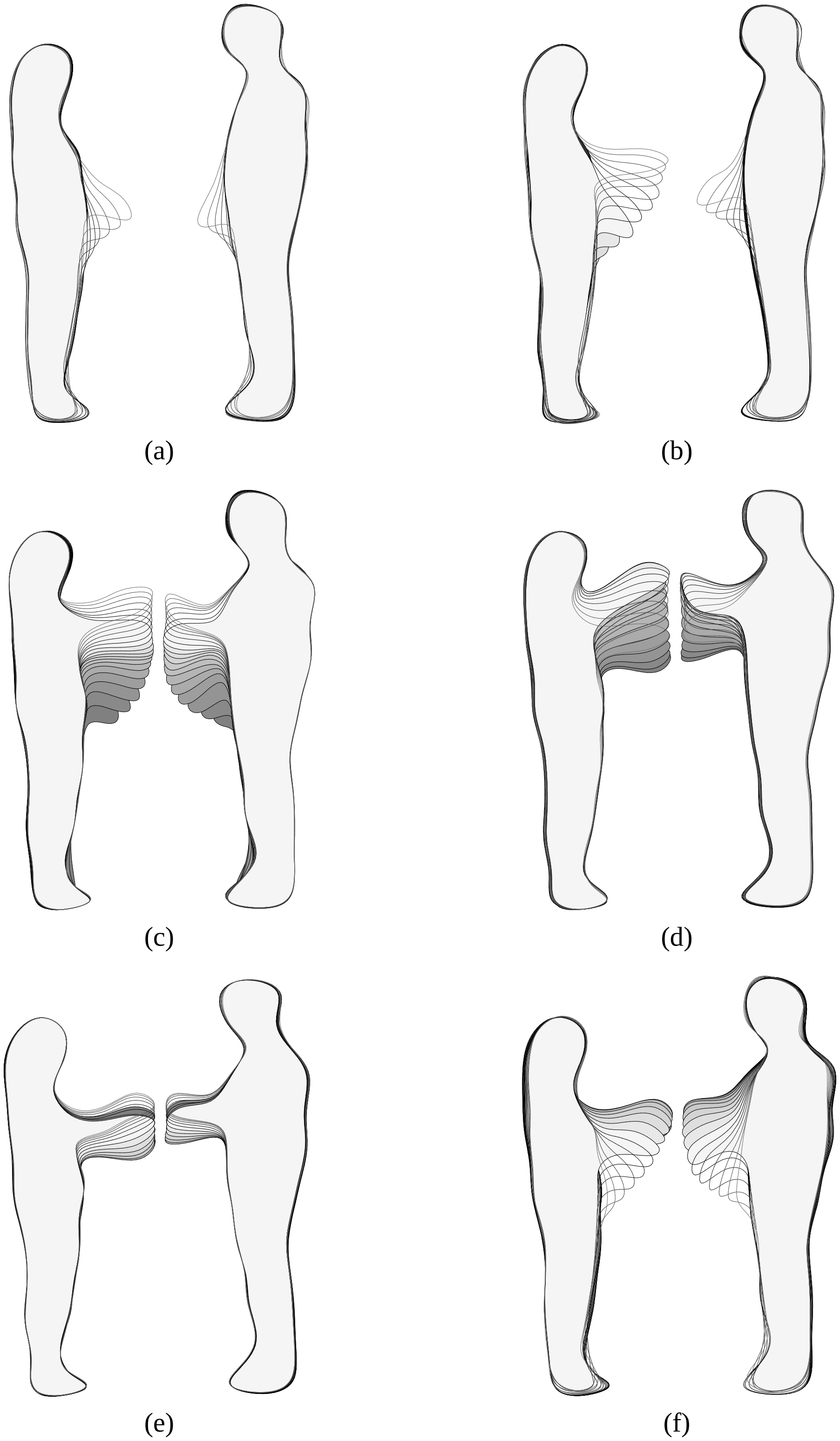


Figure 5.7: Frequency histogram illustrating behaviour prototype density matching - object interaction.

A 401-state Markov chain $\mathcal{M}_\beta^{\text{int}}$ was superimposed on the set $\mathcal{B}^{\text{int}}$ of 400 behaviour prototypes, estimating the token set $\mathcal{S}_\beta^{\text{int}}$ during learning as described in Section 4.1.3. Initial state and state transition distributions were estimated from the 13 behaviour training sets $\mathcal{G}_j^{\text{int}}$, disregarding typicality-based transition rejection as all training sequences were considered to be entirely typical. A value of $\Delta = 0.03$, half that used when learning behaviour prototypes, was used to re-sample training sets as described in Section 3.2.2, thus reducing the tendency to omit transitions associated with brief entry into a prototype's Voronoi region.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5.8: Maximum likelihood interaction extrapolation.

To demonstrate behaviour extrapolation, the learnt behaviour-based predictor $\mathcal{M}_{\beta}^{\text{int}}$ was used to generate maximum likelihood extrapolations during the evolution of handshake training sequences. Figure 5.8 illustrates extrapolation at selected time instants during the sequences, where each extrapolation was chosen randomly from the (generally singleton) set of equally maximal probability extrapolations. In each figure, recent behaviour is illustrated by a set of 12 pairs of filled contours, the shade of which indicates recency, the lightest being the current shape. The first 6 frames of each extrapolation are illustrated by a set of pairs of unfilled contours overlaying the recent behaviour, the shade of which indicates the progression of behaviour, the lightest being the furthest advanced. It is clear from these experimental results that the behaviour-based Markov chain forms an effective encoding of the evolution of spatio-temporal interactive behaviours. Extrapolated sequences are both spatially and temporally continuous and there is good spatio-temporal continuity where observed behaviour and extrapolations join.

5.3 Interacting with a virtual human

In recent years, many researchers have become interested in the development of techniques to allow a more natural form of interface between the user and the machine. In achieving this goal, it is essential that the machine is able to detect and recognise a wide range of human movements and gestures, and this has been a principal avenue of research, using a variety of spatio-temporal behaviour modelling techniques such as those reviewed in Section 2.3. An alternative approach to the provision of natural user-machine interaction is to provide the machine with the ability to learn models of natural interaction from the observation of humans, and using these acquired models, to equip a virtual human with the ability to interact in a natural way. As demonstrated in Section 5.2, the behaviour modelling framework developed in Chapter 3 and Chapter 4 allows the machine to acquire models of interactive behaviours from the extended observation of interactions between humans. Such models may also be used to simulate the evolving shape of a plausible partner during an interaction with a person.

As well as allowing prediction, extrapolation, and the generation of realistic sample behaviours, behaviour-based Markov chains form a powerful representation of the space of learnt behaviours. If such a chain is learnt from a fair sample of an interaction population then any natural interaction will follow one of the possible paths through the chain. A virtual human's behaviour can therefore be entirely defined by a behaviour-based Markov chain, where natural interaction with a human is achieved by providing responses such that the resulting sequence of state vectors forms a valid path through the chain. In this section, two approaches to following a valid path through a Markov chain are presented - a simple deterministic approach and a more robust approach utilising a stochastic tracking algorithm. Since no behaviour recognition or typicality assessment is required in either approach, the Markov chain is used in isolation from the behaviour model which it enhanced.

5.3.1 Propagating a single hypothesis

One approach to simulating the evolving shape of a plausible partner during an interaction with a person is to propagate a single *interaction state hypothesis* $\mathcal{H}$ through the Markov chain, using the evolving shape of the tracked person to determine the start state and state transitions when required. An interaction state hypothesis is defined by the pair

$$\mathcal{H}(t) = \langle \mathbf{F}_t,\ f_t \rangle , \tag{5.13}$$

where $\mathbf{F}_t$ identifies the current state of the interaction and f_t identifies the chain state towards which interaction is proceeding. At each time instant, the transformed shape vector $\mathbf{S}^{\mathrm{H}}(t)$, position $(X^{\mathrm{H}}(t),\ Y^{\mathrm{H}}(t))$ and height $h^{\mathrm{H}}(t)$ of the human are extracted from the current image as described in Section 3.1.2 and Section 5.2.1.

Within this scheme, start state and state transitions are chosen by assessing the extent to which a candidate hypothesis $\mathcal{H}_i(t)$ is consistent with the current shape of the real human $\mathbf{S}^{\mathrm{H}}_t$, using an *error measure* based on the Euclidean distance between shape vectors:

$$E(\mathbf{F}_t,\ \mathbf{S}^{\mathrm{H}}(t)) = \min \left\{ |\mathbf{S}^{\mathrm{L}}(t) - \mathbf{S}^{\mathrm{H}}(t)|,\ |\mathbf{S}^{\mathrm{R}}(t) - \mathbf{S}^{\mathrm{H}}(t)| \right\} , \tag{5.14}$$

where $\mathbf{S}^{\mathrm{L}}(t)$ and $\mathbf{S}^{\mathrm{R}}(t)$ are extracted from the hypothetical interaction state vector $\mathbf{F}_t$, and the minimisation also identifies the real human's position (either left-hand or right-hand) within the interaction.

Interaction with a virtual human is achieved using the following algorithm:

1. Select the initial hypothesis $\mathcal{H}(0)$ from the set $\mathcal{X}_0$ of all potential initial hypotheses such that the error $E(\mathbf{F}_0,\ \mathbf{S}^{\mathrm{H}}(0))$ is minimised. The potential hypotheses $\mathcal{X}_0$ are selected from *valid* initial chain states where $\pi_j \neq 0$.

2. Produce the virtual human's response $\mathbf{S}^{\mathrm{V}}(t)$ from the current hypothesis $\mathcal{H}(t)$.

3. Select the future hypothesis $\mathcal{H}(t+1)$ from the set $\mathcal{X}_{t+1}$ of all potential future hypotheses such that the error $E(\mathbf{F}_{t+1},\ \mathbf{S}^{\mathrm{H}}(t+1))$ is minimised. The potential hypotheses $\mathcal{X}_{t+1}$ are extrapolations at time $t+1$ from the current hypothesis $\mathcal{H}(t)$.

4. Repeat steps 2–3 until the end state is reached.

The virtual human's response $\mathbf{S}^{\mathrm{V}}(t)$ is produced from hypothesis $\mathcal{H}(t)$ by scaling and translating the shape vector which gave rise to the *maximum* error in Equation 5.14 (i.e. the shape vector which was not identified as being the real human). This transformation is achieved by re-arranging Equations 5.11 and 5.12 to yield the height and position of the virtual human, substituting the values of $d(t)$ and $s(t)$ from the current state vector hypothesis $\mathbf{F}_t$, and the values of $X^{\mathrm{H}}(t)$ and $h^{\mathrm{H}}(t)$ provided by the tracker.

Within this scheme, the chain state towards which interaction is proceeding (rather than the current chain state) is stored within a hypothesis, thus ensuring that the selection of the fittest transition at decision points within the Markov chain is only performed once, the decision being reflected in the stored successor state. This ensures that the interaction will precisely follow the interpolant of the interaction state vectors associated with a valid chain state sequence, whilst eliminating the need to assess when chain state membership changes. Extrapolation is achieved by sampling the Hermite interpolant of the current state vector $\mathbf{F}_t$, the state vector associated with successor chain state f_t, and, if required, the state vectors associated with successively selected chain states. Thus the set $\mathcal{X}_{t+1}$ will only contain *multiple* potential hypotheses when a decision point within the Markov chain is reached before time $t+1$.

When propagating a single state hypothesis, the selection of the start state and each successor state permanently restricts the range of possible future behaviours. In the presence of noisy data or an inaccurate or incomplete model, recovery from an undesirable selection is thus impossible, resulting in an erroneous restriction of future behaviour which may cause the interaction to fail unless the real human adopts an appropriately modified behaviour.

5.3.2 Propagating multiple hypotheses via CONDENSATION

A more robust form of interaction is achieved if *multiple* state hypotheses $\mathcal{H}_i(t)$ are propagated through the Markov chain. Within this scheme, stochastic hypothesis extrapolation allows possible alternative paths to be explored with a level of detail proportional to their probability. Further, weighted hypothesis re-sampling using a fitness function based on hypothesis error allows unfit hypotheses to be pruned whilst reinforcing the level of detail around fit hypotheses. Using a large hypothesis set, this process is equivalent to the propagation of a conditional density representation via the CONDENSATION tracking algorithm of Isard and Blake [46], where, assuming the behaviour-based Markov chain encodes high-order temporal dependencies, the propagated density will be conditioned on an entire history of observation.

In this Bayesian approach to tracking an interaction from incomplete (partially occluded) observations, the point density of state vectors from the set of multiple hypotheses approximates $p(\mathbf{F}_t \mid \mathbf{S}^{\mathrm{H}}(t),\, \ldots,\, \mathbf{S}^{\mathrm{H}}(0))$, the conditional density describing the probability of interaction state given an observation history, where

$$p(\mathbf{F}_t \mid \mathbf{S}^{\mathrm{H}}(t),\, \ldots,\, \mathbf{S}^{\mathrm{H}}(0)) \propto p(\mathbf{S}^{\mathrm{H}}(t) \mid \mathbf{F}_t)p(\mathbf{F}_t \mid \mathbf{S}^{\mathrm{H}}(t-1),\, \ldots,\, \mathbf{S}^{\mathrm{H}}(0)), \tag{5.15}$$

and where $p(\mathbf{S}^{\mathrm{H}}(t) \mid \mathbf{F}_t)$ is a fitness function measuring the *likelihood* of a state $\mathbf{F}_t$ giving rise to the observation $\mathbf{S}^{\mathrm{H}}(t)$, and $p(\mathbf{F}_t \mid \mathbf{S}^{\mathrm{H}}(t-1),\, \ldots,\, \mathbf{S}^{\mathrm{H}}(0))$ is the *prior* density representing predictions from $p(\mathbf{F}_{t-1} \mid \mathbf{S}^{\mathrm{H}}(t-1),\, \ldots,\, \mathbf{S}^{\mathrm{H}}(0))$, the *posterior* density from the previous time step. Isard and Blake identify three phases over each discrete time-step of this conditional density propagation process - deterministic drift and stochastic diffusion occur during the prediction step due to the deterministic and random components of stochastic models of dynamics, whilst reinforcement occurs during the multiplication of the likelihood and prior due to influence of measurements. Within the CONDENSATION algorithm, the posterior density is approximated by using the likelihood (fitness function) to weight sampling from the prior - a random sampling method known as *factored sampling*.

Interaction with a virtual human is achieved using a Gaussian likelihood function

$$p(\mathbf{S}^{\mathrm{H}}(t) \mid \mathbf{F}_t) = \exp\left(-\frac{E(\mathbf{F}_t,\, \mathbf{S}^{\mathrm{H}}(t))^2}{2\sigma^2}\right), \tag{5.16}$$

based on the extent to which a hypothesis is consistent with the current shape of the real human, where hypotheses are propagated using the following algorithm:

1. Generate a set $\mathcal{X}_0$ of N hypotheses to represent the initial prior, where $\mathcal{X}_0$ is obtained under sampling with replacement from the initial state distribution π.

2. For each $\mathcal{H}_i(t) \in \mathcal{X}_t$, use the error $E(\mathbf{F}_t,\ \mathbf{S}^{\mathrm{H}}(t))$ to calculate the likelihood of the hypothesis using Equation 5.16.

3. Use relative likelihood values to weight sampling from $\mathcal{X}_t$, the prior, resulting in a set $\mathcal{Y}_t$ of N hypotheses representing the posterior distribution.

4. Produce the virtual human's response $\mathbf{S}^{\mathrm{V}}(t)$ from the hypothesis $\mathcal{H}_i(t) \in \mathcal{Y}_t$ with maximum likelihood.

5. Generate a new set $\mathcal{X}_{t+1}$ of N hypotheses to represent the new prior, where each $\mathcal{H}_i(t+1) \in \mathcal{X}_{t+1}$ is a stochastic extrapolation at time $t+1$ from $\mathcal{H}_i(t) \in \mathcal{Y}_t$.

6. Repeat steps 2–5 until the interaction is complete.

The virtual human's response $\mathbf{S}^{\mathrm{V}}(t)$ is generated as described in the single hypothesis propagation approach. Since the behaviour-based Markov chains have not been extended to include noise models as described in Section 4.1.4, noise is introduced to the time interval approximation (Equation 4.7) during the generation of stochastic extrapolations. The inclusion of temporal noise allows model uncertainty to be represented and results in a more reasonable prior and thus more robust tracking. Spatial noise is omitted since, without a reasonable noise model, perturbed shapes are unlikely to appear natural. Temporal noise is sampled from a uniform distribution over $[-\frac{1}{2}\delta,\ \frac{1}{2}\delta]$ and added to the approximated time interval δ.

The propagation of multiple hypotheses representing a conditional density forms a robust statistical approach to simulating the evolving shape of a plausible partner during an interaction with a person. The algorithm described does not fully realise this potential in one respect - the virtual human's response is generated from the hypothesis with *maximum likelihood*, and not that with *maximum a posteriori* probability. Although the maximum likelihood and maximum a posteriori hypotheses typically coincide, transient maximum likelihood hypotheses associated with local maxima in the posterior density will cause the virtual human's response to skip between states. Since the posterior is represented by the $\mathcal{H}_i(t) \in \mathcal{Y}_t$, the maximum could be located (although rather expensively) by calculating the number of state vector hypotheses that fall within a hypersphere of radius Δ centred on each individual state vector hypothesis, selecting the hypothesis corresponding to

$$\max_i \{ |\{ \mathcal{H}_j(t) \, : \, |\mathbf{F}_t^j - \mathbf{F}_t^i| < \Delta, \;\; j \neq i \}| \}, \tag{5.17}$$

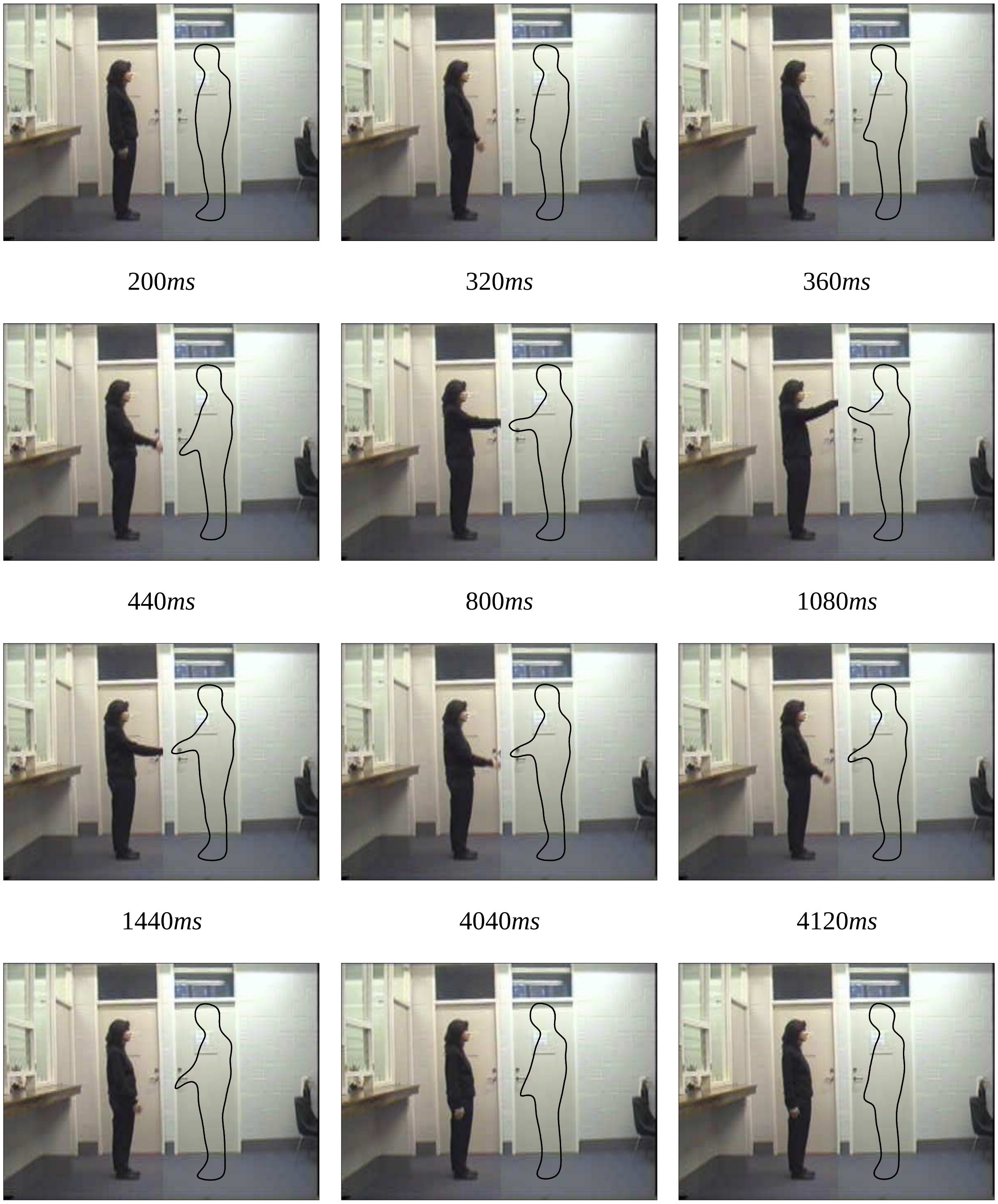
where the value of Δ could be determined experimentally.

5.3.3 Experimental results

Due to the computational requirements of the multiple hypothesis propagation algorithm, initial experiments have been performed off-line. Attempts to generate test data by capturing sequences of a single person performing a ‘blind’ handshake produced generally poor results and it was soon discovered that the behaviour exhibited was markedly different to that exhibited in real interactions. To compensate for this inability to behave naturally in the absence of an interacting partner, test sequences involving two individuals were captured and one of the individuals was masked before object tracking was performed.

Interaction experiments were performed using the 401-state Markov chain $\mathcal{M}_\beta^{\mathrm{int}}$ generated in the experiment described in Section 5.2.2.2 and the multiple hypothesis propagation algorithm. A total of 500 hypotheses were propagated using a value of $\sigma = 0.5$ in the likelihood function. A large number of prototypes, artificially high noise, and an artificially wide likelihood function were found to be necessary to compensate for inadequacies in the behaviour model caused by the lack of sufficiently representative training data. As noted by Isard and Blake [48] (in the context of robustly tracking sudden movements), artificially high noise increases the extent of predicted hypothesis clusters, thus requiring more hypotheses to effectively populate these enlarged regions. In addition to addressing temporal inaccuracies via artificially high temporal noise, inadequate behavioural description can be partially addressed by allowing hypotheses to propagate through less likely paths in the Markov chain, thus relaxing temporal dependencies to some extent. An artificially wide likelihood function and an increased number of hypotheses increases the probability of re-sampling hypotheses on less likely paths, thus increasing the extent to which these less likely paths are traversed.

Figure 5.9 shows a selection of frames from an interaction using the masked test sequence. In each frame, the virtual human’s response is displayed as a black silhouette, clearly indicating the provision of a plausible (if rather spectral) interacting partner. Since the interaction is governed by a



200*ms* 320*ms* 360*ms*

440*ms* 800*ms* 1080*ms*

1440*ms* 4040*ms* 4120*ms*

4200*ms* 4320*ms* 4560*ms*

Figure 5.9: Interaction with a virtual human.

stochastic algorithm, occasionally variation is evident in the response generated to the same test sequence. Observation of the entire set of hypotheses during an interaction suggests a distribution with a mode at the maximum likelihood state and further transient modes describing alternative paths from decision points in the chain. This distribution gradually tails off along past and future paths, whilst modes describing alternative paths tend to gradually diminish once the current shape of the real human becomes sufficiently inconsistent with the hypotheses.

5.4 Summary

In this chapter, two approaches to binary interaction modelling using the behaviour models developed in Chapter 3 and Chapter 4 have been presented - the modelling of event co-occurrence and the modelling of joint behaviour. In addition, a technique has been demonstrated which allows learnt models of natural human interaction to be used to equip a virtual human with the ability to interact in a natural way. Whilst only binary interaction has been considered, both event co-occurrence and joint behaviour modelling could easily be extended to incorporate any given number of interacting objects, thus further extending the potential for reasoning over groups of objects.

Modelling the statistical co-occurrence of events within models of the state or behaviour of individual objects allows the typicality of interactions to be assessed and provides a mechanism by which generative capabilities could be realised. Although the level of detail in such a representation is not proportional to the typicality of an interaction, the assessment of event dependence provides a useful mechanism for the filtering of candidate interactions, thus negating the need for less principled filtering mechanisms based on cues such as object proximity.

The modelling of the joint behaviour of pairs of interacting objects, using the behaviour models developed in Chapter 3 and Chapter 4, provides more detailed models of binary interaction which are both analytic and generative. Experimental results presented for a relatively simple human interaction clearly illustrate that the behaviour modelling framework is equally applicable to the modelling of such joint behaviours.

Interaction with a virtual object has been achieved using a stochastic tracking algorithm to propagate multiple interaction state hypotheses, the density of which forms a representation of interaction state probability conditioned on an entire history of observation. Since this approach allows an interaction to be robustly tracked when only one of the interacting objects is observed, the technique could also be applied to tracking single occluded objects over image sequences. Experimental results presented for a relatively simple human interaction show the successful generation of a plausible virtual partner.

Chapter 6

Conclusions

The research described in this thesis was motivated by a desire to provide a unified framework allowing the perception of effective models of characteristic object behaviours from the continuous observation of long image sequences. Using a low-level statistical modelling approach, a behaviour modelling framework has been developed in which detailed behavioural knowledge is acquired from observation, where the resulting behaviour models are both analytic and generative.

In Chapter 3, the core of the behaviour modelling framework has been developed - a hierarchical approach to behaviour modelling in which models of the probability density in behaviour space are learnt using a novel temporal pattern formation strategy which utilises models of the probability density in state space. Models constitute an optimised sample-set representation of probability density which is both highly specific and reasonably compact, and are learnt in an unsupervised manner using an extension to the standard iterative Vector Quantization algorithm. By exploiting the statistical nature of behaviour models, a typicality measure has been derived which allows both the continuous assessment of behaviour typicality and the implementation of an attentional control mechanism.

The utility of the behaviour modelling framework has been extended in chapter 4 via the superimposition of a Markov chain, the parameters of which are acquired automatically during a further learning phase. The inclusion of generative capabilities via the addition of a stochastic process model allows predictions, extrapolations and realistic sample behaviours to be generated. Since behaviour prototypes can encode entire temporal sequences, the superimposed Markov chains encode temporal dependencies within their transition structure and thus form an effective representation of the underlying dynamic processes.

Two approaches to modelling object interaction using the behaviour modelling framework have been presented in Chapter 5. The first approach considers the statistical co-occurrence of events within models of the state or behaviour of individual objects, which, via the assessment of event dependence, provides a useful mechanism for filtering candidate interactions. In the second approach, the joint behaviour of pairs of interacting objects is modelled directly and a technique is developed which, using learnt models of human interaction, enables a plausible virtual partner to be simulated during interaction between a user and the machine.

6.1 Discussion

Underlying the research described in this thesis is the belief that many useful tasks in machine vision and related disciplines which have previously been addressed using hand-crafted, often high-level, knowledge can in fact be successfully addressed using detailed, low-level statistical behaviour models which have been acquired from observation alone. In addition to demonstrating the acquisition of such models, the experimental results presented within this thesis provide some evidence of the validity of this belief. For instance, the assessment of pedestrian trajectory typicality has demonstrated the successful identification of interesting incidents - a task which has classically been approached within the automated visual surveillance domain by employing detailed hand-crafted knowledge of a scene and the behaviours of objects within it.

In addition to removing dependence on costly and inherently inaccurate hand-crafted knowledge, learnt behaviour models may, in the future, offer a mechanism by which a machine's perception of its users and environment could be enhanced. For instance, experimental results demonstrating the simulation of a plausible interactive partner using learnt models of natural human interaction suggest that learnt behaviour models may be capable of providing the basis of a novel framework within which a more natural form of user-machine interface could be developed.

In the future, it is possible to envisage the use of more detailed models of individuals and their behaviours, capable of much richer kinds of interaction - a kind of *Virtual Immortality*?

6.2 Future research

The behaviour modelling framework described within this thesis presents a number of possible avenues for future research, many of which have been identified within the body of the thesis. In this final section, a few of the more promising avenues for future research are summarised.

A natural process for the perception of behaviour models should allow gradual temporal adaptation, enabling model evolution with occasional changes in characteristic behaviour. Although the techniques developed within this thesis are capable of such adaptivity, further research is required to assess both its stability and its effectiveness over extended periods of time. In addition, many behaviours which appear to change more frequently, such as the trajectories of pedestrians within a city centre, are in fact dependent on a temporal context, and it would be interesting to investigate the inclusion of such dependencies within behaviour models.

As indicated by experimental results presented within this thesis, the behaviour modelling framework may be applicable to a wide range of tasks within machine vision and related disciplines. Such tasks include event recognition and incident identification within automated visual surveillance systems, increasing the robustness and efficiency of object tracking systems, providing recognition and segmentation within gestural interfaces, and the automatic generation of realistic object behaviours within animations, virtual worlds, and computer generated film sequences. Further research is clearly required to realise this potential over the range of possible applications.

References

[1] E. André, G. Herzog, and T. Rist. On the Simultaneous Interpretation of Real World Image Sequences and their Natural Language Description: The System SOCCER. In *Proc. 8th European Conference on Artificial Intelligence*, pages 449–454, August 1988.

[2] N. I. Badler. Real-Time Virtual Humans. In *Proc. 5th Pacific Conference on Computer Graphics and Applications*, October 1997.

[3] N. I. Badler, C. B. Phillips, and B. L. Webber. *Simulating Humans: Computer Graphics, Animation, and Control*. Oxford University Press, 1993.

[4] A. Baumberg and D. Hogg. An Efficient Method for Contour Tracking using Active Shape Models. In *Proc. IEEE Workshop on Motion of Non-rigid and Articulated Objects*, pages 194–199, November 1994.

[5] A. Baumberg and D. Hogg. Learning Flexible Models from Image Sequences. In *Proc. 3rd European Conference on Computer Vision*, volume 1, pages 299–308, May 1994.

[6] A. Baumberg and D. Hogg. Generating Spatiotemporal Models from Examples. *Image and Vision Computing*, 14(8):525–532, August 1996.

[7] A. M. Baumberg. *Learning Deformable Models for Tracking Human Motion*. PhD thesis, The University of Leeds, October 1995.

[8] E. L. Bienenstock, L. N. Cooper, and P. W. Munro. Theory for the Development of Neuron Selectivity; Orientation Specificity and Binocular Interaction in Visual Cortex. *The Journal of Neuroscience*, 2(1):32–48, 1982.

[9] M. J. Black, Y. Yacoob, and S. X. Ju. Recognizing Human Motion Using Parameterized Models of Optical Flow. In M. Shah and R. Jain, editors, *Motion-Based Recognition*, volume 9 of *Computational Imaging and Vision series*, pages 245–269. Kluwer Academic Publishers, 1997.

[10] A. Blake, M. A. Isard, and D. Reynard. Learning to Track the Visual Motion of Contours. *Artificial Intelligence*, 78:101–134, 1995.

[11] B. M. Blumberg and T. A. Galyean. Multi-Level Direction of Autonomous Creatures for Real-Time Virtual Environments. In *Proc. SIGGRAPH 95*, pages 47–54, August 1995.

[12] A. F. Bobick and J. W. Davis. Action Recognition Using Temporal Templates. In M. Shah and R. Jain, editors, *Motion-Based Recognition*, volume 9 of *Computational Imaging and Vision series*, pages 125–146. Kluwer Academic Publishers, 1997.

[13] A. F. Bobick and A. D. Wilson. A State-based Technique for the Summarization and Recognition of Gesture. In *Proc. 5th International Conference on Computer Vision*, pages 382–388, June 1995.

[14] M. Brand, N. Oliver, and A. Pentland. Coupled hidden Markov models for complex action recognition. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 994–999, June 1997.

[15] A. J. Bulpitt. *A Multiple Adaptive Resonance Theory Architecture Applied To Motion Recognition Tasks*. PhD thesis, University of York, June 1994.

[16] A. J. Bulpitt and N. M. Allinson. Human Motion Recognition Using Co-operative ART Networks. In *Proc. World Congress on Neural Networks*, volume 3, pages 708–711, July 1993.

[17] H. Buxton and S. Gong. Advanced Visual Surveillance using Bayesian Networks. In *Proc. IEEE Workshop on Context-based Vision*, pages 111–123, June 1995.

[18] G. A. Carpenter and S. Grossberg. ART 2: self-organization of stable category recognition codes for analog input patterns. *Applied Optics*, 26(23):4919–4930, 1987.

[19] C. Cédras and M. Shah. Motion-based recognition: A survey. *Image and Vision Computing*, 13(2):129–155, March 1995.

[20] T. F. Cootes and C. J. Taylor. Active Shape Models - ‘Smart Snakes’. In *Proc. British Machine Vision Conference*, pages 266–275, September 1992.

[21] T. F. Cootes and C. J. Taylor. A Mixture Model for Representing Shape Variation. In *Proc. British Machine Vision Conference*, pages 110–119, September 1997.

[22] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Training Models of Shape from Sets of Examples. In *Proc. British Machine Vision Conference*, pages 9–18, September 1992.

[23] D. R. Corrall and A. G. Hill. Visual Surveillance. *GEC Review*, 8(1):15–27, 1992.

[24] T. Darell and A. Pentland. Space-Time Gestures. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 335–340, June 1993.

[25] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B*, 39:1–38, 1977.

[26] D. DeSieno. Adding a Conscience to Competitive Learning. In *Proc. IEEE International Conference on Neural Networks*, volume 1, pages 117–124, 1988.

[27] J. L. Elman. Finding Structure in Time. *Cognitive Science*, 14:179–211, 1990.

[28] J. Fernyhough, A. G. Cohn, and D. C. Hogg. Building Qualitative Event Models Automatically from Visual Input. In *Proc. 6th International Conference on Computer Vision*, pages 350–355, 1998.

[29] J. H. Fernyhough, A. G. Cohn, and D. C. Hogg. Generation of Semantic Regions from Image Sequences. In *Proc. 4th European Conference on Computer Vision*, volume 2, pages 475–484, 1996.

[30] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, second edition, 1990.

[31] B. Fritzke. Growing Cell Structures - A Self-Organizing Network for Unsupervised and Supervised Learning. *Neural Networks*, 7(9):1441–1460, 1994.

[32] D. M. Gavrila and L. S. Davis. Towards 3-D model-based tracking and recognition of human movement: a multi-view approach. In *Proc. International Workshop on Automatic Face and Gesture Recognition*, pages 272–277, 1995.

[33] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1992.

[34] S. Gong and H. Buxton. On the Visual Expectations of Moving Objects. In Neumann B., editor, *Proc. 10th European Conference on Artificial Intelligence*, pages 781–784. John Wiley & Sons, 1992.

[35] R. M. Gray. Vector Quantization. *IEEE ASSP Magazine*, 1(2):4–29, April 1984.

[36] Grzeszczuk and D. Terzopoulos. Automated Learning of Muscle-Actuated Locomotion Through Control Abstraction. In *Proc. SIGGRAPH 95*, pages 63–70, August 1995.

[37] S. Haykin. *Neural Networks - A Comprehensive Foundation*. Macmillan College Publishing Company, 1994.

[38] T. Heap and D. Hogg. Improving Specificity in PDMs using a Hierarchical Approach. In *Proc. British Machine Vision Conference*, pages 80–89, September 1997.

[39] T. Heap and D. Hogg. Wormholes in Shape Space: Tracking through Discontinuous Changes in Shape. In *Proc. 6th International Conference on Computer Vision*, pages 344–349, 1998.

[40] G. Herzog and P. Wazinski. VIsual TRAnslator: Linking Perceptions and Natural Language Descriptions. *Artificial Intelligence Review*, 8:175–187, 1994.

[41] D. Hogg. Model-Based Vision: A Program to See a Walking Person. *Image and Vision Computing*, 1(1):5–20, 1983.

[42] D. C. Hogg. *Interpreting Images of a Known Moving Object*. PhD thesis, University of Sussex, January 1984.

[43] R. Howarth and H. Buxton. Selective attention in dynamic vision. In *Proc. 13th International Joint Conference on Artificial Intelligence*, pages 1579–1584, August 1993.

[44] R. J. Howarth and H. Buxton. Analogical representation of space and time. *Image and Vision Computing*, 10(7):467–478, September 1992.

[45] X. D. Huang, Y. Ariki, and M. A. Jack. *Hidden Markov Models for Speech Recognition*, volume 7 of *Edinburgh Information Technology series*. Edinburgh University Press, 1990.

[46] M. Isard and A. Blake. Contour Tracking by Stochastic Propagation of Conditional Density. In *Proc. 4th European Conference on Computer Vision*, volume 1, pages 343–356, April 1996.

[47] M. Isard and A. Blake. A mixed-state CONDENSATION tracker with automatic model-switching. In *Proc. 6th International Conference on Computer Vision*, pages 107–112, 1998.

[48] M. Isard and A. Blake. ICONDENSATION: Unifying Low-Level and High-Level Tracking in a Stochastic Framework. In *Proc. 5th European Conference on Computer Vision*, volume 1, pages 893–908, June 1998.

[49] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception & Psychophysics*, 14(2):201–211, 1973.

[50] K. Kakusho, N. Babaguchi, and T. Kitahashi. Recognition of Social Dancing from Auditory and Visual Information. In *Proc. 2nd International Conference on Automatic Face and Gesture Recognition*, pages 289–294, October 1996.

[51] T. Kohonen. The Self-Organizing Map. *Proceedings of the IEEE*, 78(9):1464–1480, September 1990.

[52] J. Laszlo, M. van de Panne, and E. Fiume. Limit Cycle Control And Its Application To The Animation Of Balancing And Walking. In *Proc. SIGGRAPH 96*, pages 155–162, August 1996.

[53] G. F. Lawler. *Introduction to Stochastic Processes*. Chapman & Hall, 1995.

[54] Y. Linde, A. Buzo, and R. M. Gray. An Algorithm for Vector Quantizer Design. *IEEE Transactions on Communications*, COM-28(1):84–95, January 1980.

[55] F. Liu and R. W. Picard. Finding Periodicity in Space and Time. In *Proc. 6th International Conference on Computer Vision*, pages 376–383, 1998.

[56] S. P. Luttrell. Code Vector Density in Topographic Mappings: Scalar Case. *IEEE Transactions on Neural Networks*, 2(4):427–436, July 1991.

[57] R. J. Morris and D. C. Hogg. Statistical Models of Object Interaction. In *Proc. IEEE Workshop on Visual Surveillance*, pages 81–85, January 1998.

[58] S. Nagaya, S. Seki, and R. Oka. A Theoretical Consideration of Pattern Space Trajectory for Gesture Spotting Recognition. In *Proc. 2nd International Conference on Automatic Face and Gesture Recognition*, pages 72–77, October 1996.

[59] H.-H. Nagel. From image sequences towards conceptual descriptions. *Image and Vision Computing*, 6(2):59–74, May 1988.

[60] S. A. Niyogi and E. H. Adelson. Analyzing gait with spatiotemporal surfaces. In *Proc. IEEE Workshop on Motion of Non-rigid and Articulated Objects*, pages 64–69, November 1994.

[61] B. North and A. Blake. Using Expectation-Maximisation to Learn Dynamical Models from Visual Data. In *Proc. British Machine Vision Conference*, pages 669–678, September 1997.

[62] N. Oliver, B. Rosario, and A. Pentland. Statistical modeling of human interactions. In *Proc. IEEE Workshop on the Interpretation of Visual Motion*, June 1998.

[63] A. Pentland. Machine Understanding of Human Motion. Technical Report 350, MIT Media Laboratory Perceptual Computing Section, September 1995.

[64] K. Perlin. Real Time Responsive Animation with Personality. *IEEE Transactions on Visualization and Computer Graphics*, 1(1):5–15, March 1995.

[65] R. Polana and R. Nelson. Low Level Recognition of Human Motion. In *Proc. IEEE Workshop on Motion of Non-rigid and Articulated Objects*, pages 77–82, November 1994.

[66] R. Polana and R. Nelson. Temporal Texture and Activity Recognition. In M. Shah and R. Jain, editors, *Motion-Based Recognition*, volume 9 of *Computational Imaging and Vision series*, pages 87–124. Kluwer Academic Publishers, 1997.

[67] A. Psarrou, S. Gong, and H. Buxton. Modelling Spatio-Temporal Trajectories and Face Signatures on Partially Recurrent Neural Networks. In *Proc. IEEE International Conference on Neural Networks*, volume 5, pages 2226–2231, November 1995.

[68] L. R. Rabiner and B. H. Juang. An Introduction to Hidden Markov Models. *IEEE ASSP Magazine*, 3:4–16, January 1986.

[69] M. Reiss and J. G. Taylor. Storing Temporal Sequences. *Neural Networks*, 4:773–787, 1991.

[70] G. Retz-Schmidt. A REPLAI of SOCCER: Recognizing Intentions in the Domain of Soccer Games. In *Proc. 8th European Conference on Artificial Intelligence*, pages 455–457, August 1988.

[71] D. Reynard, A. Wildenberg, A. Blake, and J. Marchant. Learning Dynamics of Complex Motions from Image Sequences. In *Proc. 4th European Conference on Computer Vision*, volume 1, pages 357–368, 1996.

[72] C. W. Reynolds. Flocks, Herds, and Schools: A Distributed Behavioural Model. *Computer Graphics*, 21(4):25–34, July 1987.

[73] R. D. Rimey and C. M. Brown. Controlling Eye Movements with Hidden Markov Models. *International Journal of Computer Vision*, 7(1):47–66, November 1991.

[74] B. D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

[75] H. Ritter. Asymptotic Level Density for a Class of Vector Quantization Processes. *IEEE Transactions on Neural Networks*, 2(1):173–175, January 1991.

[76] K. Rohr. Towards Model-Based Recognition of Human Movements in Image Sequences. *CVGIP: Image Understanding*, 59(1):94–115, January 1994.

[77] P. L. Rosin and T. Ellis. Detecting and Classifying Intruders in Image Sequences. In *Proc. British Machine Vision Conference*, pages 293–300, September 1991.

[78] D. E. Rumelhart and D. Zipser. Feature Discovery by Competitive Learning. *Cognitive Science*, 9:75–112, 1985.

[79] R. Schalkoff. *Pattern Recognition: Statistical, Structural and Neural Approaches*. John Wiley & Sons, 1992.

[80] S. M. Seitz and C. R. Dyer. Cyclic Motion Analysis Using the Period Trace. In M. Shah and R. Jain, editors, *Motion-Based Recognition*, volume 9 of *Computational Imaging and Vision series*, pages 61–85. Kluwer Academic Publishers, 1997.

[81] M. Shah and R. Jain, editors. *Motion-Based Recognition*, volume 9 of *Computational Imaging and Vision series*. Kluwer Academic Publishers, 1997.

[82] M. Shah and R. Jain. Visual Recognition of Activities, Gestures, Facial Expressions and Speech: An Introduction and a Perspective. In M. Shah and R. Jain, editors, *Motion-Based Recognition*, volume 9 of *Computational Imaging and Vision series*, pages 1–14. Kluwer Academic Publishers, 1997.

[83] K. Sims. Evolving Virtual Creatures. In *Proc. SIGGRAPH 94*, pages 15–22, July 1994.

[84] T. Starner and A. Pentland. Visual Recognition of American Sign Language Using Hidden Markov Models. In *Proc. International Workshop on Automatic Face and Gesture Recognition*, pages 189–194, 1995.

[85] N. Sumpter, R. D. Boyle, and R. D. Tillett. Modelling Collective Animal Behaviour using Extended Point Distribution Models. In *Proc. British Machine Vision Conference*, pages 242–251, September 1997.

[86] X. Tu and D. Terzopoulos. Artificial Fishes: Physics, Locomotion, Perception, Behaviour. In *Proc. SIGGRAPH 94*, pages 43–49, July 1994.

[87] M. Turk and A. Pentland. Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, 3(1):71–96, 1991.

[88] D. Wang and M. A. Arbib. Complex Temporal Sequence Learning Based on Short-term Memory. *Proceedings of the IEEE*, 78(9):1536–1542, September 1990.

[89] A. Wilson and A. Bobick. Learning Visual Behaviour for Gesture Analysis. In *Proc. IEEE International Symposium on Computer Vision*, pages 229–234, 1995.

[90] L.-Q. Xu and D. Hogg. Neural Networks in Human Motion Tracking - An Experimental Study. In *Proc. British Machine Vision Conference*, pages 405–414, September 1996.

[91] Y. Yacoob and M. J. Black. Parameterized Modeling and Recognition of Activities. In *Proc. 6th International Conference on Computer Vision*, pages 120–127, 1998.

[92] Y. Yacoob and L. Davis. Learned Temporal Models of Image Motion. In *Proc. 6th International Conference on Computer Vision*, pages 446–453, 1998.

[93] J. Yamato, J. Ohya, and K. Ishii. Recognizing Human Action in Time-Sequential Images using Hidden Markov Model. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 379–385, June 1992.

[94] Y. Zheng and J. F. Greenleaf. The Effect of Concave and Convex Weight Adjustments on Self-Organizing Maps. *IEEE Transactions on Neural Networks*, 7(1):87–96, January 1996.