Learning Deformable Models for Tracking Human Motion

by

Adam Michael Baumberg

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

The University of Leeds
School of Computer Studies
October 1995

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.

Adam Baumberg

School of Computer Studies
University of Leeds

Doctor of Philosophy
October 1995

Learning Deformable Models for Tracking Human Motion

Abstract

The analysis and automatic interpretation of images containing moving non-rigid objects, such as walking people, has been the subject of considerable research in the field of computer vision and pattern recognition. In order to build fast and reliable systems some kind of prior model is generally required. A model enables the system to cope with situations where there is considerable background clutter or where information is missing from the image data. This may be due to imaging errors (e.g. bluring due to motion) or due to part of an object becoming hidden from view.

Conventional approaches to the problem of tracking non-rigid objects require complex handcrafted models which are not easily adapted to different problems. A more recent approach uses training information to build models for image analysis. This thesis extends this approach by building flexible 2D models, *automatically*, from sequences of training images. Efficient methods are described for using the resulting models for real time contour tracking using optimal linear filtering techniques. The method is further extended by incorporating a feedback scheme to generate a more compact linear model which is shown to be more robust and accurate for tracking.

Models of the shape of an object do not utilise the temporal information contained within the training sequences. A novel method is described for automatically learning a spatiotemporal, physically-based model that allows the system to accurately predict the expected change in object shape over time. This approach is shown to increase the reliability of the system, requiring only a modest increase in computational processing.

The system can be automatically trained on video sequences to learn constraints on the apparent shape and motion of a particular non-rigid object in a particular environment. Results show the system is capable of tracking several walking pedestrians in real time without the use of expensive dedicated hardware. The output from this system has potential uses in the areas of surveillance, animation and gait analysis.

Declarations

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

Baumberg A. and Hogg D.C., "Learning Flexible Models from Image Sequences", Proceedings European Conference on Computer Vision, Stockholm, May 1994.

Baumberg A. and Hogg D.C., "An Efficient Method for Contour Tracking using Active Shape Models", Proceedings IEEE Workshop on Motion of Non-rigid and Articulated Objects, Texas, Nov 1994.

Baumberg A. and Hogg D.C., "An Adaptive Eigenshape Model", Proceedings British Machine Vision Conference, Birmingham, Sept 1995.

Baumberg A. and Hogg D.C., "Learning Spatiotemporal Models from Examples", Proceedings British Machine Vision Conference, Birmingham, Sept 1995.

Acknowledgements

I would like to thank David Hogg for introducing me to computer vision and for his helpful and enthusiastic supervision over the last three and a bit years. Also thanks to him for agreeing to fund several trips to conferences and workshops.

I am grateful to Xinquan Shen for some helpful and stimulating discussions which were particularly useful when I was starting off. I'd also like to thank Shaun, Greg and Rob (and the rest of the lab) for all that coffee and entertaining chat. Particular thanks to Stuart for organising many memorable social events and for use of his extensive library of textbooks. Thanks also to Li-Qun and Andy for proof-reading my work.

Most of all, I would like to thank my wife, Louise, for her support and encouragement, without whom this thesis would never have been written.

I am also grateful to EPSRC for providing my funding.

Contents

List of Figures

Chapter 1

Introduction

The work described in this thesis was motivated by addressing a seemingly simple problem – to track the positions of a number of pedestrians in an outdoor scene. The aim of this work is to automatically process sequences of images taken from a camera viewing outdoor pedestrian scenes. Some typical images are shown in figure 1.1. The system should be able to extract the position of the moving pedestrians in the scene and to follow each person throughout the sequence. Ideally additional information such as silhouette shape should be available for higher level event recognition routines such as deciding whether the person is walking or running. The system should be suitable for a range of applications such as automated surveillance, animation and human motion analysis.

This problem is an example of an inherently difficult class of problems in machine vision – the analysis of the motion of non-rigid objects. In order to tackle these problems some kind of simplifying assumptions are generally required to constrain the allowable object shape and motion. These constraints allow the system to cope with missing data (where the object becomes hidden from view), noise in the image data and background clutter. These *a priori* constraints embody a model of the object.

1.1 A hierarchy of object models

The model-based approach to image understanding allows the incorporation of prior knowledge into the system. This approach has some biological foundations. Human beings interpret visual

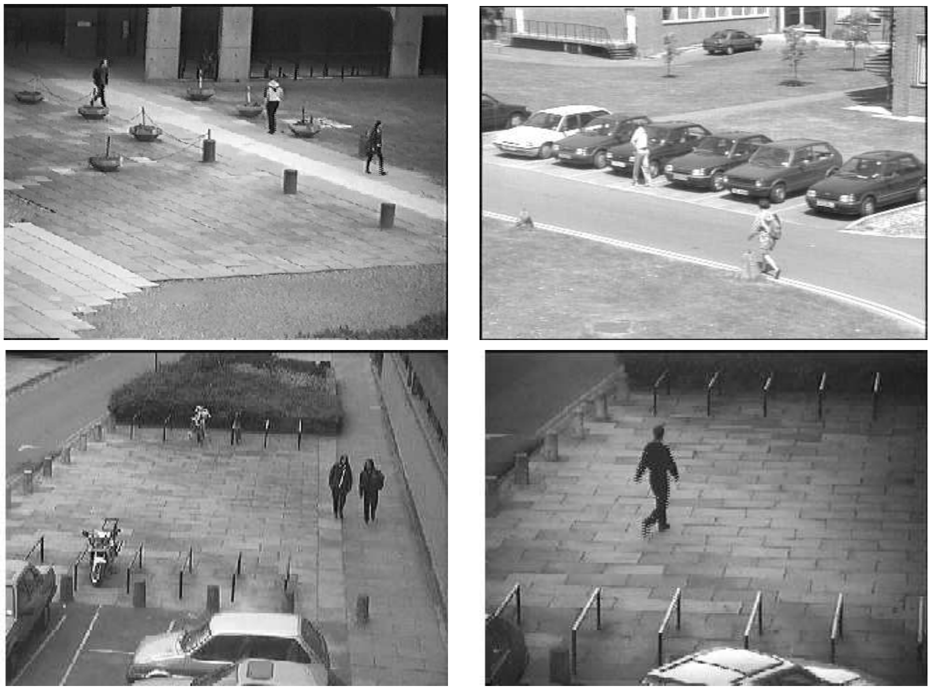


Figure 1.1: Example camera images

information by comparison to knowledge in our memories. One important consideration in object modeling is the specificity of the model. For instance, in studying the motion of a walking person a full 3D model describing the precise position of limbs and joints over time could be used. However, such a prescriptive model may only describe one particular walk by a particular pedestrian. At the other end of the spectrum, a very general model would be a deformable 3D parametric surface represented by a mesh of 3D points which incorporates little knowledge into the system. Between these two extremes lie a range of possible models.

A further practical consideration is the dimensionality of the model space. In general any model has an associated set of parameters such as joint angles, height, width, orientation, etc. Given the model and a set of parameters the object features can be projected into an image. The problem of image search becomes one of identifying the model parameters that when projected most closely resembles features in a given image. The computational expense of this process is related to the dimensionality of the model space (the higher the dimensionality the more costly the search).

1.2 Approach taken

To some extent the choice of model is dependent on the application. In order to track moving pedestrians in cluttered noisy images some *a priori* model was found to be necessary (especially when the video camera is not fixed). The approach taken in this work is to build a “non-representational” model (i.e. with no notion of limbs etc) derived from real training data. The model is acquired automatically (requiring no operator input). The advantage of this approach is that the method can be applied to a wide range of problems without re-engineering the whole system. This contrasts with more conventional hand-crafted models. The training data allows the model to be tuned to the particular constraints of a given object in a given scenario. The method is “data driven” which allows the system to cope with shapes that are not usually represented in conventional models (e.g. variability in shape due to clothing).

A feature of this work is that the 2D outline (or silhouette) of the object is modeled. An advantage of this approach is that in the majority of cases the object silhouette is observable in the image (assuming the object is not occluded) whereas a complete set of internal features of an object are rarely apparent in all images (due to self-occlusion). For example, in the case of a walking pedestrian the arm is often hidden behind the rest of the person’s body. Furthermore the model parameters encapsulate variability in the outline due to orientation as well as change in shape due to articulation.

The use of a 2D model to describe a 3D object is a unique feature of this work. The pose of the 3D object is not completely unconstrained but represented by the typical poses in the training set. Hence the position of the imaging device (e.g. video camera) with respect to the ground plane is implicitly incorporated into the model. This may appear to be a significant drawback, but in real applications the camera location is rarely completely unconstrained. For instance, the camera is unlikely to be looking directly up at a person’s feet (although if this *were* the case, the system could still be trained up on these images). In fact unusual viewing angles are often confusing to a human observer.

A deforming silhouette seems to incorporate considerable information in much the same way as the moving light displays of Johansson [1] and it is not difficult for the human visual system to interpret silhouette sequences such as that shown in figure 1.2. This suggests that a silhouette model may be applicable to high level recognition tasks as well as to the original tracking application.

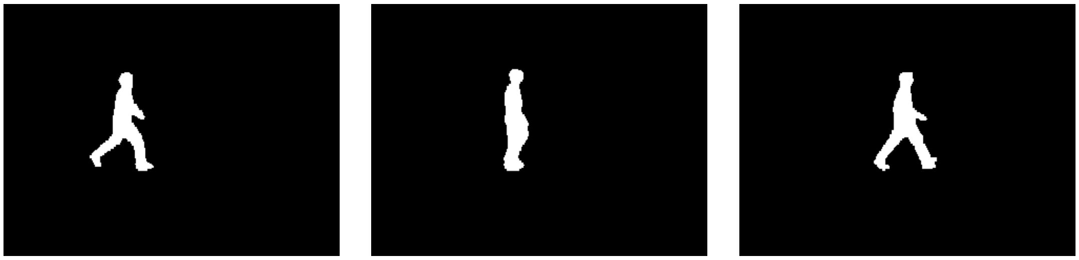


Figure 1.2: Three images from a sequence of silhouettes

1.3 Overview of the thesis

In this introduction some of the broader issues relating to the area have been discussed. Chapter 2 gives a review of techniques relevant to the problem of tracking non-rigid objects and related problems. The remaining chapters describe the original work of the thesis and include results on real image sequences. The work is organised as follows :-

- **Chapter 3**

  A novel method for automatically building a linear shape model of a moving object is described using training image sequences taken with a fixed camera. A novel method for extending conventional point based statistical methods to parametrised curves is given.

- **Chapter 4**

  A new efficient method for contour tracking based on a linear shape model is outlined. Current methods in optimal linear filtering are incorporated into the mechanism, resulting in a fast and robust, variable scale tracking scheme.

- **Chapter 5**

  A simple method for improving the linear shape model is detailed, based on an iterative feedback mechanism. A compact linear model is automatically generated.

- **Chapter 6**

  The linear spatial shape model is extended to a physically-based spatiotemporal model learnt from training sequences of typical object motion. The new spatiotemporal model is shown to be more robust than the previous spatial models.

Finally, conclusions and future work are discussed in Chapter 7.

Chapter 2

Background review

2.1 Introduction

Automatically tracking the motion of a non-rigid object, such as a walking person, from sequences of images is a challenging problem which in general requires some kind of prior information to be solvable. In this chapter, current techniques in non-rigid motion analysis are discussed as well as some more specific techniques applicable to human motion analysis.

Prior information can be derived statistically from training information using "Principal Component Analysis" (PCA) as in the Point Distribution Model (PDM) outlined in section 2.2. Alternatively, physically motivated constraints can be utilised which limit object shape to elastic deformations of a template as in the Finite Element approach outlined in section 2.3. These approaches are usually regarded as "model-based" as the prior information contains the approximate shape of the object. These two key approaches have many similarities and can be combined (see [2]). They are both to some extent linear models in that the model features are related by a linear transformation to model shape parameters. (This is only true if the pose parameters of the PDM are fixed). In both cases linear shape "modes" are derived using an eigenanalysis method and both methods produce highly compact models with a small set of parameters.

Other approaches such as "snakes", "Kalman snakes" and "Active Splines" (reviewed in sections 2.4 and 2.5) make fewer shape assumptions. These methods are 2D, contour based approaches where object shape is constrained to be continuous and smooth and to deform smoothly.

These more general approaches are not conventionally described as model-based approaches (although any set of constraints can be regarded as a low-level model). In the interpretation of real images – that is images that have been captured from a camera in an outdoor environment – more detailed prior knowledge is generally required. Apart from the problems of self-occlusion previously noted, real images are often of poor quality due to poor lighting and low resolution. Other problems include shadows, reflections (e.g. due to wet road surfaces) and poor weather conditions (rain, cloud, etc). Such problems can only be overcome by incorporating more prior information into the model (i.e. using a higher level model).

High-level models can incorporate a great deal of information about object shape and even expected motion over time. Examples of 3D representational models include the cylinder-based model, WALKER [3], described in section 2.7 and a similar model used by Rohr [4]. These complex models are "hand-crafted" and consist of an explicit 3D representation of the object generated by a human expert (e.g. a programmer). The model has few parameters – In the case of Rohr's model there is one "pose" parameter. This results in fast and robust tracking but will fail when the input walk does not fit the typical walking motion described in the model (e.g. atypical behaviour such as running or suddenly stopping) or when the imaging device is non-stationary. Similar 2D "stick" models have also been used with some success in controlled environments, e.g. Leung and Yang [5]. These approaches utilise models based on a theoretical conceptualisation. Consequently such an approach suffers when the reality differs from this preconceived model (e.g. variability in shape due to clothing, atypical walks, etc) although some degree of error-tolerance can be allowed. The alternative data-driven approach builds a model from a representative set of real training data.

Other approaches assume the joints of the human body have been marked (e.g. Chen and Lee [6], Bulpitt [7]). In section 2.6, the non-representational eigenimage model of Murphy *et al* [8] is summarised. This approach is related to the "eigenface" approach of Turk and Pentland [9] and the grey-level extensions to the PDM [10, 11] and has many similarities with the approach taken in this thesis. However, one of the drawbacks of "image" based representations is the computational cost involved in operating on relatively large windows of image pixels (e.g. in calculating optical flow). Furthermore the dimensionality of the resulting model is still high (typically 30 model parameters are used) and the method usually relies on a fixed camera.

Commercial surveillance systems use a simple background subtraction image processing technique to recover moving objects within a scene. This technique (described in section 2.8) requires a fixed camera (i.e. stationary, with fixed zoom and aperture) and is the first step in many of the tracking systems described above. In fact this technique proves useful in the model acquisition method described in this thesis. Background subtraction has many limitations, not least of which is the requirement that the camera is fixed. Rowe and Blake [12] have extended this approach by using a stationary steerable camera which can pan and tilt and mapping the image onto a fixed "virtual camera" image plane. Even with a fixed camera, subtraction methods are sensitive to changes in light, poor contrast, reflections as well as occlusion and imaging noise.

2.2 The Linear Point Distribution Model

2.2.1 Description of the model

Statistical analysis of 2D landmark data has become a well established tool in computer vision (e.g. morphological methods [13]). A recent advance in this area is the Point Distribution Model (PDM) introduced by Cootes *et al* [14, 15, 16]. In general, a PDM is a statistical model of a set of (2D or 3D) points. The statistical model described by Cootes *et al* is a linear model (ignoring the rotational component of the model) and will be referred to as the "Linear Point Distribution Model" or LPDM in this thesis. The LPDM has been used successfully for image interpretation (e.g. with medical images [17, 16] and for automatic face identification [18]) and image sequence analysis (e.g. using a stochastic deformable model [19]).

In a PDM, shape is represented by a set of n labeled "landmark" points (see figure 2.1 for an example). Each point corresponds to a particular (often biological) feature on the object such as the tip of the index finger in the case of modeling a hand.

The LPDM is based on a statistical analysis of the coordinates of these points over a training set. Each training shape can be represented by a shape-vector $\mathbf{x}$, consisting of the landmark coordinates. Modeling in 2D,

$$\mathbf{x} = (x_0, y_0, x_1, y_1, ..., x_{n-1}, y_{n-1})^T$$

where (x_i, y_i) is the position of the i'th landmark point on the training shape.

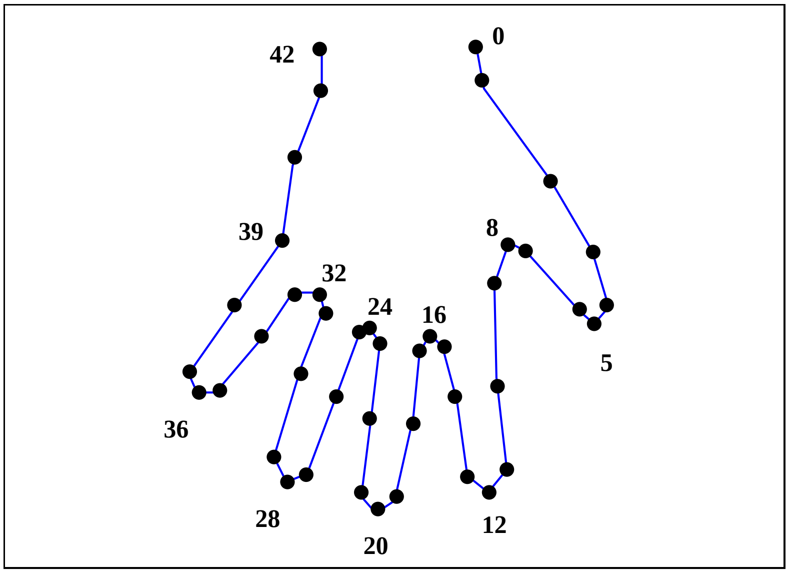


Figure 2.1: A PDM representation of a hand shape

The training shapes are aligned using a Generalised Procrustes Analysis technique (as derived by Gower [20]). A weighted least squares method is used to align each shape to the mean shape. The weights are chosen so that more significance is given to the more "stable" landmark points. This process results in a mean shape-vector $\overline{\mathbf{x}}$ and a set of aligned training shape-vectors $\mathbf{x_i}$. The next stage in the analysis is to subtract the mean shape-vector from each training shape-vector, i.e. let

$$\mathbf{dx_i} = \mathbf{x_i} - \overline{\mathbf{x}} \tag{2.1}$$

The $2n \times 2n$ covariance matrix S is then calculated using

$$S = E\left(\mathbf{dx}\,\mathbf{dx}^T\right) \tag{2.2}$$

where $E(...)$ is the expectation (or averaging) operator over the training set.

Modes of variation of the landmark points are represented by the $2n$ unit-length eigenvectors of S that solve

$$S\mathbf{e_i} = \lambda_i \mathbf{e_i}$$

where $\lambda_0 \geq \lambda_1 \geq ... \geq \lambda_{2n-1} \geq 0$. The eigenvectors form an orthonormal basis for the shape space. Hence the shape-vectors $\mathbf{dx}$ can be rewritten in the form

$$\mathbf{dx} = \sum_{i=0}^{2n-1} b_i \mathbf{e_i}$$

where $b_i = \mathbf{dx} \cdot \mathbf{e_i}$

It can be shown that over the training set the parameters b_i are linearly independent and the total variance explained by each eigenvector is equal to the associated eigenvalue. i.e.

$$E(b_i b_j) = \begin{cases} 0 & : \quad i \neq j \\ \lambda_i & : \quad i = j \end{cases}$$

Thus the eigenvectors corresponding to the largest eigenvalues represent the most significant modes of variation. A subset containing the m most significant eigenvectors is retained as a basis for the model shape space. A shape in the model space x can be written as a sum of the mean shape and a weighted sum of eigenvectors using

$$\mathrm{x} = \overline{\mathrm{x}} + P\mathbf{b} \tag{2.3}$$

where P is a $2n \times m$ matrix whose columns are the m most significant eigenvectors and $\mathbf{b} = (b_0, ..., b_{m-1})^T$ is a vector of m coefficients. Given an aligned shape vector x', the minimum least squares approximation to the shape in the model space is given by a linear projection,

$$\mathbf{b} = P^T \left(\mathrm{x}' - \overline{\mathrm{x}}\right) \tag{2.4}$$

This eigenvector analysis is an application of “Principal Component Analysis” or the Karhunen-Loeve Transform (see for example, Gonzalez and Woods [21]).

New “feasible” shapes can be generated by varying the shape parameters b_i within suitable limits. As the variance of the i’th shape parameter within the training set is simply λ_i, suitable limits might be $\pm 2\sqrt{\lambda_i}$.

2.2.2 Active Shape Models

Cootes *et al* describe a method (the “Active Shape Model”) for locally optimising the shape parameters of the LPDM to fit features in an image [15]. The “Active Shape Model” (ASM) can be regarded as a 2D application of Lowe’s refinement technique [22]. The LPDM is particularly suited to this kind of iterative approach due to the simplicity in deriving an appropriate Jacobian matrix for updating the shape parameters.

The ASM assumes a rough initial estimate for the orientation, scale and position of the model as well as the linear shape parameters. Given these parameters the model shape can be projected into the image frame using

$$\mathbf{X} = Q(s,\theta)\left[\overline{\mathbf{x}} + P\mathbf{b}\right] + \mathbf{X_c} \tag{2.5}$$

where $Q(s,\theta)$ is a rotation by θ and a scaling by s and $\mathbf{X_c}$ is a translation by (X_c, Y_c). The shape-vector $\mathbf{X}$ represents the position of the n landmark points in image coordinates.

At each iteration of the refinement process, suggested movements for each landmark point $d\mathbf{X_i}$ are calculated from image features. The usual approach is to search for the strongest edge, along the normal to the model boundary at each landmark point. The vector $d\mathbf{X_i}$ is set to the displacement of the estimated landmark position to the edge feature and scaled proportionally to the edge strength (to reflect the confidence in this measurement). The method is illustrated in figure 2.2.

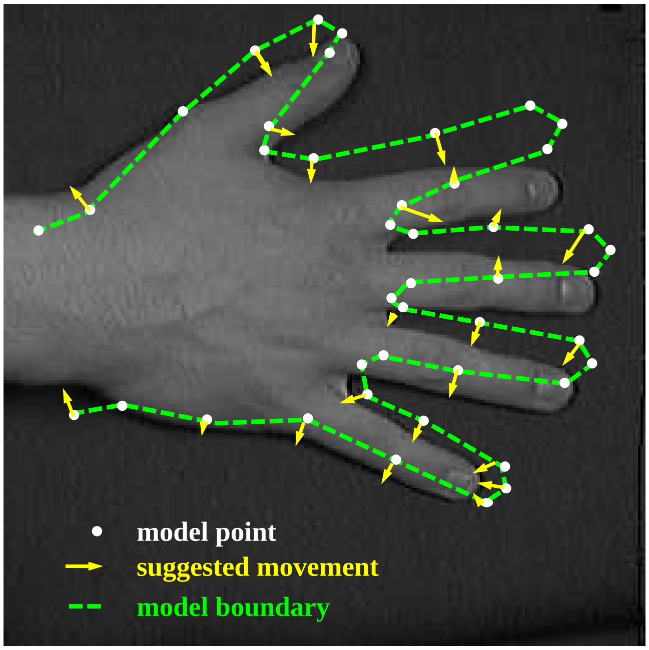


Figure 2.2: Updating an Active Shape Model

Given this set of displacements represented by the shape-vector displacement $d\mathbf{X}$, estimates for changes in the pose parameters dX_c, dY_c, $d\theta$ and the relative change in scale ds are calculated.

Cootes *et al* project the point-displacements in the image frame to displacements in the model coordinate frame, $d\mathrm{x}$ using the equation

$$d\mathrm{x} = Q((s(1+ds))^{-1}, -(\theta + d\theta))[Q(s,\theta)\mathrm{x} + d\mathbf{X} - d\mathbf{X}_{\mathrm{c}}] - \mathrm{x}$$

which can be rewritten in the form

$$Q(s',\theta')d\mathrm{x} = d\mathbf{X} - \underbrace{(Q(s',\theta') - Q(s,\theta))\,[\mathrm{x}]}_{\text{term 1}} - \underbrace{d\mathbf{X}_{\mathrm{c}}}_{\text{term 2}} \tag{2.6}$$

where $s' = s(1+ds)$ and $\theta' = \theta + d\theta$. Equation 2.6 can be interpreted as correcting the displacements $d\mathbf{X}$, taking into account the updated pose: “term 1” removes the changes in scale and rotation and “term 2” removes the change in origin.

The model point displacements $d\mathrm{x}$ are projected into adjustments to the vector of shape parameters $\mathbf{b}$, using

$$d\mathbf{b} = P^T d\mathrm{x}$$

which is simply the least squares solution to the problem

$$Jd\mathbf{b} = d\mathbf{X}'$$

where $d\mathbf{X}'$ is the vector of corrected point displacements in the image frame and J is the Jacobian matrix with respect to the m shape parameters. i.e.

$$J_{ij} = \frac{\delta(d\mathbf{X}')_i}{\delta b_j}$$

From equation 2.5, $J = QP$.

The shape and pose parameters are updated using a weighted update scheme as follows

$$\begin{aligned} X_c &\rightarrow X_c + w_t dX_c \\ Y_c &\rightarrow Y_c + w_t dY_c \\ \theta &\rightarrow \theta + w_\theta d\theta \\ s &\rightarrow s(1 + w_s ds) \\ \mathbf{b} &\rightarrow \mathbf{b} + W_b d\mathbf{b} \end{aligned} \tag{2.7}$$

where w_s, w_θ, w_t are scalar weights and W_b is a diagonal matrix of weights for each shape parameter. In the conventional ASM, W_b is set to the identity or preferably, each weight is set proportional to the standard deviation of the corresponding shape parameter over the training set. This allows the more significant shape parameters to vary more freely.

Each iterative step refines the shape and pose parameters to reduce the error between image edge features and the projected model. After each iteration the shape parameters are further constrained to ensure the shape is close enough to the mean shape in terms of a Mahalanobis distance metric. Explicitly

$$\begin{aligned} s^2 &= \sum \frac{b_i^2}{\lambda_i} \\ b_i' &= \begin{cases} \left(\frac{s_{\max}}{s}\right) b_i & s > s_{\max} \\ b_i & \text{otherwise} \end{cases} \end{aligned} \tag{2.8}$$

where s_{max} is the maximum allowed distance from the mean. The constraint ensures the vector $\mathbf{b}$ lies within a hyper-ellipsoid centered about the origin. Points within this hyper-ellipsoid have a reasonably high *a priori* probability density, assuming the training shapes were sampled from a Gaussian distribution about the mean shape (see Haslam *et al* [23]).

In order to improve the speed and robustness of the ASM, a multi-scale search mechanism can be used, described by Cootes *et al* [24].

2.2.3 Lowe refinement

Lowe describes an iterative scheme for fitting parametrised 3D models to images [22]. The scheme is based on Newton's method and is stabilised using *a priori* constraints. Given a vector of non-linear parameters $\mathbf{p}$ a sequence of estimates are calculated using

$$\mathbf{p}^{(i+1)} = \mathbf{p}^{(i)} - \mathbf{q}$$

At each iteration $\mathbf{q}$ is calculated by minimising,

$$||J\mathbf{q} - \mathbf{e}||^2 + \alpha^2 ||W(\mathbf{q} - \mathbf{d})||^2 \tag{2.9}$$

where

- $\mathbf{e}$ is the error between estimated and observed positions of model features in the image
- $\mathbf{d}$ is a vector of *a priori* parameter constants (the "prior model")
- W is a diagonal matrix in which each weight is inversely proportional to the standard deviation σ_i for parameter i
- J is the Jacobian matrix.
- α is a "trade-off" weight that is dynamically adjusted to affect the stability and rate of convergence.

The 1st term in equation 2.9 pulls the solution towards the image data and the 2nd term stabilises the solution by pulling towards the starting position $\mathbf{d}$. In order to ensure the final solution closely fits the image data, the starting point of the prior model $\mathbf{d}$ is reset to the results of the previous iteration.

Applying this scheme to the shape parameters of an ASM would minimise the following error at each iteration

$$||(\mathbf{b}' - \mathbf{b}) - d\mathbf{b}||^2 + \alpha^2||W(\mathbf{b}' - \mathbf{b})||^2$$

where $\mathbf{b}'$ is the vector of updated shape parameters and W is a diagonal matrix of weights with

$$W_{ii} = \frac{1}{\sqrt{\lambda_i}}$$

This leads to the update equation

$$b_i' = b_i + \left(\frac{\lambda_i}{\alpha^2 + \lambda_i}\right) db_i \tag{2.10}$$

which is similar (but not identical) to the ASM update of equation 2.7 in that the more significant modes with larger eigenvalues vary more freely than the less significant modes.

2.3 The Finite Element Method

The Finite Element Method (FEM) is an engineering technique for efficient computational simulation of physical systems (see, for example, Bathe [25]). Pentland and Sclaroff describe the application of these techniques to problems in computer vision [26, 27, 28, 29]. The approach taken is to build an elastic physical model of a deformable object and use finite element analysis to produce a compact, orthogonal set of shape parameters suitable for tracking and recognition tasks. Nastar and Ayache have successfully applied these techniques in the analysis of time sequences of 3D medical data sets [30, 31].

In contrast to the training based approach of the PDM, the FEM utilises a physical model generated from a single example of the object's shape along with certain assumptions about the physical material properties of the object. Using "Modal Analysis" it is possible to reduce the dimensionality of the FEM shape representation without a significant loss in accuracy. This allows the (theoretical) physical system to become over-constrained where insufficient measurements are available as well as reducing the computational load of the simulation.

2.3.1 Shape representation in Finite Element Analysis

The basic concept of the FEM is to represent a body in terms of a set of regions or "elements" described by a set of labeled nodes. The quantity of interest (in this case, displacement) is approximated by a set of piecewise continuous functions over the body, defined over a finite number of sub-domains called elements. The interpolation function used is continuous and usually a low order polynomial. Some typical finite elements are illustrated in figure 2.3.

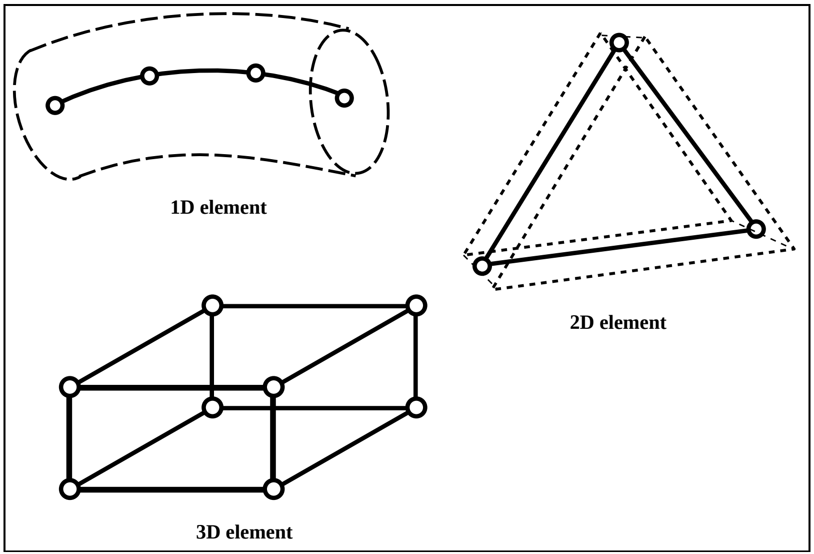


Figure 2.3: Some finite elements

Hence object shape is represented by a set of nodal displacements $\mathbf{U}$ from an initial shape with nodal representation $\mathbf{X}$. A shape is regarded as the result of pushing, pinching and pulling an initial lump of elastic material. Unlike the PDM, the FEM provides an analytic characterisation of the object surface between nodes.

2.3.2 Modal Analysis

Utilising known or assumed physical properties of the object (such as stress and strain matrices, uniform density), global mass, damping and stiffness matrices are derived by formulating appropriate integrals over each element and summing over the whole domain.

The resulting governing equation describes the evolution of the system over time under the influence of external loads acting on the nodes and for a system of n nodes in d dimensions is given by

$$M\ddot{\mathbf{U}} + C\dot{\mathbf{U}} + K\mathbf{U} = \mathbf{R}(t) \tag{2.11}$$

where $\mathbf{U}$ is the $dn \times 1$ vector of nodal displacements, M, C and K are $dn \times dn$ symmetric matrices describing the mass, damping and material stiffness between each point within the object and $\mathbf{R}$ is a $dn \times 1$ vector of external forces acting on the nodes.

The modal analysis approach decouples the above system by transforming to a basis of "M-orthogonal" free vibration modes derived by solving the eigenvalue problem

$$K\phi_i = {\omega_i}^2 M\phi_i \tag{2.12}$$

Assuming Rayleigh damping ($C = a_0 M + a_1 K$), the system of equations is decoupled into dn independent 2nd order differential equations. This is achieved by defining a transformation matrix Φ whose columns are the eigenvectors ϕ_i.

$$\Phi = [\phi_1, \phi_2, ..., \phi_{dn}]$$

Then, letting $\mathbf{U} = \Phi\tilde{\mathbf{U}}$, the governing equation (equation. 2.11) becomes

$$\ddot{\tilde{\mathbf{U}}} + \tilde{C}\dot{\tilde{\mathbf{U}}} + \Omega^2\tilde{\mathbf{U}} = \Phi^T\mathbf{R}(t) \tag{2.13}$$

where Ω^2 is a diagonal matrix of eigenvalues,

$$\Omega^2 = \begin{pmatrix} {\omega_1}^2 & & & \\ & {\omega_2}^2 & & \\ & & \ddots & \\ & & & {\omega_{dn}}^2 \end{pmatrix}$$

and $\tilde{C} = a_0 I + a_1 \Omega^2$ is also diagonal. Each vibration mode has an associated frequency ω_i. The higher frequency vibration modes can be ignored as theoretically they will have little amplitude and are generally difficult to measure with any degree of accuracy. The lower frequency modes tend to correspond to intuitive deformations such as “bending” and “shearing”.

The modal amplitudes and modal velocities can be dynamically estimated by time-integration of the transformed governing equation (2.13) or equivalently using a Kalman filter mechanism (see, for example, Gelb [32]). The modal analysis approach has several benefits. The 2nd order governing equation is decoupled and there is a reduction in dimensionality achieved by ignoring high frequency modes. This results in faster and more efficient tracking and shape recovery methods.

2.4 Snakes and Kalman Snakes

The snake (or active contour model) of Kass *et al* [33] provides a powerful mechanism for low-level image interpretation (e.g. for tracking deformable objects in the plane [34]). A snake is an energy-minimising spline that is attracted to image features such as edges. An internal energy function regularises the problem, modeling the spline as an elastic membrane (with constraints on smoothness). A local energy-minimisation technique (such as an Euler method) is employed so that the discretised contour “slithers” down the nearest well in the energy surface. The dynamic system can be viewed in terms of image forces pulling the contour towards edge features and internal “elastic” forces maintaining smoothness.

A simple snake minimises the energy

$$E_{\text{snake}} = \int_0^1 E_{\text{int}}\left[\mathbf{v}(s)\right] + E_{\text{image}}\left[\mathbf{v}(s)\right] ds$$

where the contour’s coordinate functions are denoted by $\mathbf{v}(s) = (x(s), y(s))$.

The internal deformation energy is given by

$$E_{\text{int}} = \alpha(s)|\mathbf{v}_{\mathrm{s}}(s)|^2 + \beta(s)|\mathbf{v}_{\mathrm{ss}}(s)|^2$$

The two "physical" parameters $\alpha(s)$ and $\beta(s)$ control the "tension" and "rigidity" of the contour at a given point.

The external image forces are derived from the energy potential E_{image} which can be set as follows

$$E_{\text{image}}(x, y) = -c|\nabla(G_\sigma \circ I(x, y))|$$

where $G_\sigma \circ I$ denotes the convolution of a Gaussian filter with the image and σ controls the spatial scale. The Gaussian blurring effectively increases the size of the energy well around a local minimum. As the snake reaches equilibrium, the spatial scale of the Gaussian filter is reduced to recover finer detail.

In order to perform the minimisation, the snake is discretised at regular sample points $\mathbf{v}_{\mathbf{i}} = (x_i, y_i)$ and an iterative local optimisation procedure applied.

Terzopoulos and Szeliski have shown that the elastic snake system is equivalent to a steady state Kalman filter with constant unit covariance matrix [35]. They describe a true Kalman filter approach, the "Kalman Snake" which provides a mechanism for tracking an elastic snake contour over successive image frames. One advantage of this approach is that the model parameters (such as the weighting attributed to new measurements) can be derived from a statistical sensor model and can be allowed to vary over time.

Terzopoulos *et al* have extended the 2D snake model to elastically deformable 3D models [36, 37].

2.5 Active Splines

Blake *et al* describe a statistical framework for efficiently tracking B-spline contours using a Kalman filter mechanism [38]. These "Active Splines" are evolved from the principles of the snake. For computational efficiency a contour is represented by a parametric curve such as a cubic B-spline. The implicit continuity and elasticity of the B-spline allows a simple stochastic model to be used for contour tracking without the need for an explicit "regularising" internal energy function. Prior knowledge can be incorporated into the tracker by an elastic coupling with a template B-spline ("Coupled Contours" [39]). This persistent template mechanism improves stability by incorporating shape memory, restricting the prior distribution of the contour shape. An extended affine invariant shape template is described [38] which allows the contour to more readily undergo affine transformations. In this section, the method will be examined in more detail.

2.5.1 State Space

A (closed) B-spline curve $\mathbf{v}(s) = (X(s), Y(s))$ is defined parametrically for $0 \leq s \leq N$ in terms of N time varying control points $\mathbf{Q_k} = (X_k(t), Y_k(t))$ by

$$\begin{aligned} \mathbf{v}(s) &= \sum_{k=1}^{N} B_k(s)\mathbf{Q_k} \\ &= H(s)\mathbf{Q_k} \end{aligned}$$

where B_k is a piecewise cubic interpolation function for the i'th control point and

$$H(s) = (B_1(s), B_2(s), ...B_N(s))$$

The state space is represented by the state vectors $\mathbf{X} = (X_1, ...X_N)$ and $\mathbf{Y} = (Y_1, ...Y_N)$. Blake *et al* introduce a distance metric associated with this state space given by

$$d(\mathbf{X}, \mathbf{X}') = |\mathbf{X} - \mathbf{X}'|$$

where the norm $|...|$ is defined by

$$\begin{aligned} |\mathbf{X}|^2 &= \int_0^N X(s)^2 ds \\ &= \mathbf{X}^T \mathcal{H} \mathbf{X} \end{aligned}$$

and the matrix $\mathcal{H}$ is given by

$$\mathcal{H}_{i,j} = \int_0^N B_i(s)B_j(s)ds \qquad (2.14)$$

2.5.2 Feature Search

An observed contour $(X_f(s), Y_f(s))$ is defined by searching along normals (or parallel lines) from the current estimate $(\widehat{X}, \widehat{Y})$ within a search window. An elliptical search window is derived analytically from the covariance of the current estimate. In the interests of speed the contrast is examined at three points: on the estimated curve and at the two extremes of the search window. The point with the highest contrast (i.e. intensity gradient) is retained as the observed value of $(X_f(s), Y_f(s))$. The contrast is measured at the given search scale. If there is no significant measurement at the 3 points, the search window is halved and the process repeated. When there is no significant feature found within the window (i.e "lock" is lost), no observation is made. An diagram illustrating feature search is shown in figure 2.4.

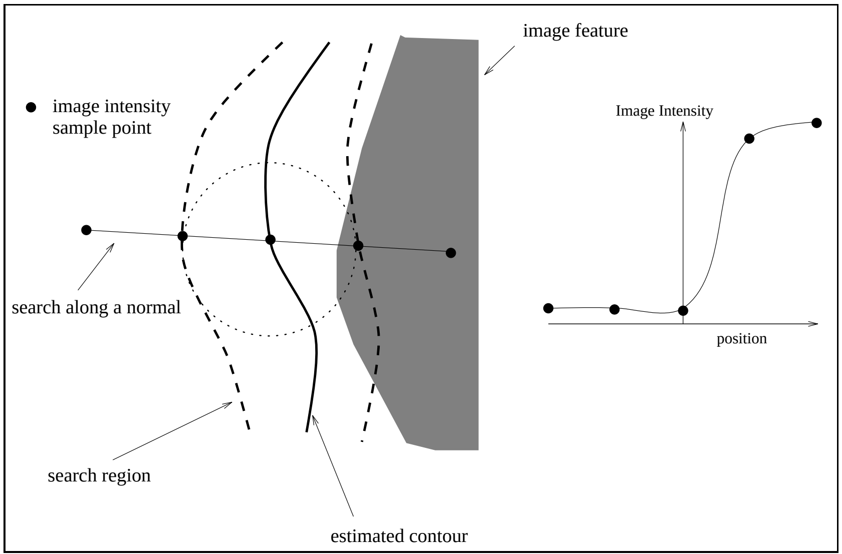


Figure 2.4: Feature search along a normal – Image intensities are sampled at discrete points along the normal

In reality the measurements $(X_f(s), Y_f(s))$ are made at discrete curve points. However, a theoretical, continuous sensor model can be shown to be equivalent to state space observations $(\mathbf{X_f}, \mathbf{Y_f})$, the least-squares approximation to the continuous observed curve points, with an associated covariance matrix R, for each of $\mathbf{X_f}$ and $\mathbf{Y_f}$ given by

$$R = r\mathcal{H}^{-1}$$

where r is the measurement variance constant.

The variance of a point measurement is set proportional to the size of the search window to reflect the fact that measurement errors will be larger when the search scale is large. As each point measurement is made by searching along a straight line, the X and Y measurements will be coupled (i.e. the measurement is not isotropic). In this summary, the isotropic case will be assumed although Blake *et al* describe the appropriate modifications.

2.5.3 Stochastic Dynamic Model

The control point positions are modeled using a constant velocity model with random accelerations expressed by the equation

$$\frac{d}{dt}\begin{pmatrix}\mathrm{X}\\ \dot{\mathrm{X}}\end{pmatrix} = \begin{pmatrix}\dot{\mathrm{X}}\\ \mathbf{0}\end{pmatrix} + \begin{pmatrix}\mathbf{0}\\ \mathrm{w}\end{pmatrix}$$

where $\mathrm{w}(t)$ is a zero-mean, temporally uncorrelated Gaussian noise process. A similar independent equation applies for $\mathbf{Y}$. Assuming an isotropic, homogeneous Gaussian noise distribution, the covariance matrix for w is proportional to $\mathcal{H}^{-1}$.

2.5.4 Kalman filter mechanism

Between successive image frames no observations are made and the covariance matrix P, associated with the augmented state estimate $(\hat{\mathrm{X}}, \hat{\dot{\mathrm{X}}})$ is updated appropriately. Observations of the point-feature $(X_f(s), Y_f(s))$ at time $t = t_k$ are applied sequentially using the Kalman filter update equation

$$\begin{pmatrix}\hat{\mathrm{X}}\\ \hat{\dot{\mathrm{X}}}\end{pmatrix} \rightarrow \begin{pmatrix}\hat{\mathrm{X}}\\ \hat{\dot{\mathrm{X}}}\end{pmatrix} + K(s)\left(X_f(s, t_k) - H(s)\hat{\mathrm{X}}\right)$$

where the Kalman gain is given by

$$K(s) = P\begin{pmatrix}H(s)^T\\ \mathbf{0}\end{pmatrix}\left[(H(s)|\mathbf{0})P\begin{pmatrix}H(s)^T\\ \mathbf{0}\end{pmatrix} + \sigma^2\right]^{-1}$$

and σ is the standard deviation of the individual point measurement.

A persistent template mechanism can be applied using a virtual input of $\mathbf{0}$ applied to the filter but coupled outside the subspace $\mathcal{V}$ of affine transformations of the template. The template stabilises the system preventing the contour from becoming tangled and increasing robustness.

2.5.5 Spatio-temporal scale

One advantage of using a statistical Kalman filter framework is that the covariance of the current estimate models the positional variance of each point on the contour. Assuming isotropy, a circular search window is constructed about each contour point with radius $2\rho(s,t)$ where ρ^2 is the positional variance at s given by

$$\rho(s)^2 = (H(s)|\mathbf{0})P\,(H(s)|\mathbf{0})^T$$

In the absence of image measurements when "lock" is lost over the whole contour, the search scale increases as the uncertainty of the state estimates increase with time. Similarly, the Kalman gain will increase so that when new measurements are eventually applied, the contour will react quickly and lock onto the image feature.

Once the contour has "locked on" (i.e. the estimated contour is reasonably close to the underlying object contour and this contour lies within the uncertainty bounds of the estimated contour) the search window and Kalman gain decrease allowing motion coherence to be exploited and the contour to be recovered more accurately.

2.6 Eigenimage decomposition

Murphy *et al* [8] describe a novel approach to analysis of human motion based on eigenimage decomposition. Their approach is "task-based" as opposed to the conventional "representational" computer vision paradigm. The basis of the method is to use the Karhunen-Loeve Transform (KLT) on a statistically representative set of training images.

A modified KLT procedure is used for computational efficiency. Images of size $n \times m$ are considered as nm element vectors. Typically n and m are large (> 64) resulting in image vectors with over 8000 elements. In a similar manner to the LPDM the mean image vector is removed and a linearly independent eigenbasis calculated (these are called "eigenimages"). Given N training images, where N is typically equal to 100, eigenimages can be calculated from the eigenvectors of an $N \times N$ "pseudo-covariance" matrix.

An image which is "similar" to the images contained in the training set can be represented by a linear combination of a subset of the eigenimages (added to the mean image). Typically 30 coefficients are sufficient to represent images for recognition of pose.

In the experiments of Murphy *et al* raw images are not used. Instead, the magnitude of the optical flow at each pixel is used as input to the KLT. Image sequences are represented by sequences of the 30 most significant KL coefficients. The resulting information is fed into a neural net classifier. The method has been applied to side view images of humans on a treadmill and to outdoor images of subjects walking in front of a stationary camera. In order to extract suitable image windows, a simple correlation process was required to track the person across the image. Using this method the pose of the subject can be identified and it is possible to identify each of a small class of subjects on the basis of gait.

Although well suited to high level recognition tasks this approach is still computationally expensive requiring many pixel-based operations. The method does not appear to solve the general pedestrian tracking problem in a noisy environment (e.g. for a crowded scene) that has motivated the work in this thesis. A similar approach is taken by Turk and Pentland for face representation [9]. Cootes *et al* have combined an eigenimage approach with the shape model of the LPDM [10, 11].

2.7 The WALKER model

Articulated, primitive based 3D models have been used successfully in a variety of applications (e.g. DigitEyes [40], Lowe refinement [22]). Much of this work is based on the work of Hogg [3] in which a representational model of a walking person (based on the Marr and Nishihara body model [41]) is used. The WALKER model of Hogg represents object shape in terms of elliptical cylinders representing rigid parts of the body and connected appropriately at the joints (see figure 2.5). A pedestrian's posture is parametrised by a set of joint angles, for example the angle between the torso and the left thigh (the "Left Hip" joint angle).

The WALKER model represents a class of walking motions in terms of an idealised walk cycle. Each joint angle is modeled as a periodic function of a parameter PSTR representing the position in the walk cycle. The joint angle functions were precomputed by analysis of a particular walk sequence and are represented by 10 point cubic B-splines. The allowable postures are constrained by allowing each joint angle to be slightly out of step with the idealised posture cycle.

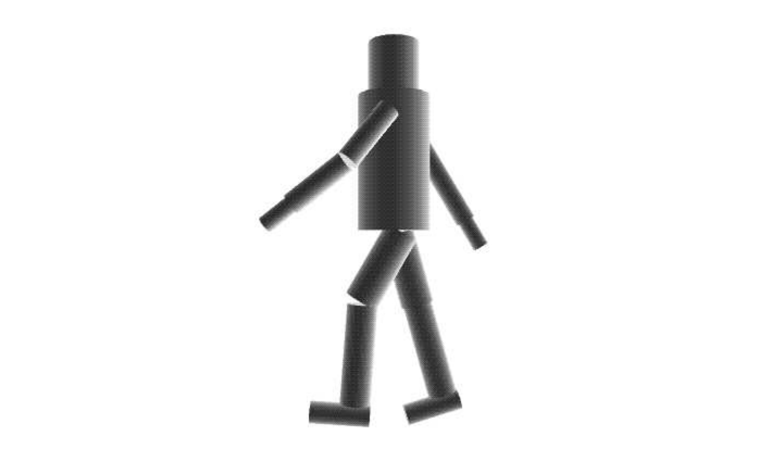


Figure 2.5: A pedestrian shape in the WALKER model

For instance,

$$\begin{aligned} \mathrm{LEFT\_HIP} &= \mathrm{hip\_curve}(\mathrm{PSTR} + \mathrm{DPSTR}) \\ -0.04 &< \mathrm{DPSTR} < 0.04 \end{aligned}$$

where LEFT_HIP is the angle between the torso and the left thigh and the function $\mathrm{hip\_curve}$ is a smooth periodic function describing this angle for the idealised walk cycle.

The posture parameter PSTR is constrained to vary slowly over time. The walker is constrained to move in the direction he or she is facing and constraints on the speed of motion are also explicitly incorporated into the model.

2.7.1 Tracking with WALKER

At each image frame, Hogg propagates the WALKER model constraints to obtain a set of box constraints on the joint angles and position parameters. An evaluation or plausibility function $\mathrm{EVAL}(s)$ of an instantaneous model instance s (representing the joint angles and position parameters) is defined using a weighted sum of independent evaluations for the different body parts. The search space is sampled and the most plausible model instance obtained for the current image using a “generate and test” strategy. The model constraints are then propagated to the next image frame and the process repeated. By evaluating the plausibility of each part independently, a more efficient search procedure is employed.

The plausibility functions are based on projecting the cylinder model onto the image to obtain a set of “ribbons”. Each ribbon consists of a pair of parallel line segments which correspond to the “side” edges of a projected cylinder. The plausibility of a ribbon is calculated using a “fuzzy” matching function, by searching a rectangular strip about each line segment for suitable edge features in the image.

The tracking procedure works well when strong constraints exist and the resulting search space is not too large. For the first image frame a change detection method (see section 2.8) or a global hierarchical search mechanism is required.

A similar approach, based on the work of Hogg is described by Rohr [4]. In this work, Rohr reduces the parameter search space by only tracking one posture parameter based on the position within a generic walk cycle (the generic model is based on a set of 60 male walks). The motion is constrained to be parallel to the image plane. The significant extensions in this work include the removal of hidden model contours and the use of a Kalman filter.

Both these methods have proven successful in recovering full 3D descriptions of a walking pedestrian from real image data in a constrained environment. The models used contain a large amount of prior information which has been hand-generated, requiring considerable time and resources. These approaches are domain dependent and require new (hand-generated) models to be applied to new situations (e.g. other types of human motion). Murphy *et al* have shown that a full 3D representation is not always necessary (such as for recognition on the basis of gait).

2.8 Background subtraction and change detection

2.8.1 Change detection

Change detection is a method for detecting moving objects in an image sequence taken with a fixed camera. Given two successive (grey-scale) image frames $I_{k+1}(x, y)$ and $I_k(x, y)$ a differenced image is calculated by subtracting the image intensities at each pixel. i.e.

$$D_k(x, y) = |I_{k+1}(x, y) - I_k(x, y)|$$

The differenced image D_k is usually thresholded to obtain a binary image with pixels flagged where there is a significant change in intensity.

Under the assumptions of a fixed camera, with fixed aperture and constant lighting conditions the flagged pixels correspond to parts of a moving object. If the moving object is “flat filled” then there will be flagged pixels corresponding to the leading and trailing edges of the object. If the object is textured some of the internal pixels will also be flagged.

The flagged pixels can be grouped by clustering to obtain a set of regions. Processing a scene with one or more moving objects which are well separated in the image will result in regions corresponding to each moving object. A more robust approach, differencing image features (such as edges) is described by Jain *et al* [42]. These techniques have been employed in a variety of applications (e.g. by Hogg [3], Rohr [4], Li-Qun [43], among others).

2.8.2 Background subtraction

Another powerful technique, background subtraction, relies on the availability of a “background” reference image $I_{\text{ref}}(x, y)$. This image may be obtained by acquiring an image from a fixed camera when there are no moving objects in the scene. Alternatively, a background image can be obtained from a sequence of images $I_k(x, y)$ by median filtering over time. Explicitly

$$I_{\text{ref}}(x, y) = \text{Median}(I_0(x, y), I_1(x, y)...I_n(x, y))$$

The median filter may be replaced by an appropriate robust running average, updated periodically to account for changing lighting conditions.

Image subtraction (and thresholding) is performed as for change detection and the resulting flagged pixels correspond to objects of interest (such as moving objects). Assuming the camera is stationary with fixed lighting conditions and good contrast, the method can be used to segment moving objects in a scene. Connected components of flagged pixels usually correspond to separate objects and small regions can be ignored. However, when several moving objects overlap in the image (or are too close together) only one amalgamated region is obtained.

This technique has been used as a first step in many vision applications (e.g. by Niyogi and Adelson [44], Murphy *et al* [8]). Both of these image subtraction techniques are sensitive to shadows, changes in lighting (e.g. due to the sun passing behind a cloud), camera vibrations, poor contrast and occlusion.

An extension of this method to deal with a steerable camera which is allowed to pan and tilt is described by Rowe and Blake [12]. The camera image is back-projected onto a “virtual camera” image plane which remains fixed. A background image is generated for the virtual camera image by sweeping the camera across the scene. A statistical model for each pixel is required to cope with errors in the projection process (due to unmodeled depth variation within the scene). The method is computationally expensive, typically taking several hours to build a model of the background. Once the background has been extracted, contour tracking can be performed in real-time.

Chapter 3

Building a Contour Model

3.1 Introduction

The “Point Distribution Model” outlined in section 2.2 has proven a useful mechanism for building a compact shape model from training examples of a class of shapes. In this thesis, the class of shapes of interest are the 2D silhouettes of walking pedestrians viewed from a variety of angles. The conventional PDM requires a human operator to hand generate a set of labeled points (corresponding to particular features) from training images of the object of interest. This data set is then processed automatically to generate a mean shape and a set of modes of variation with associated shape parameters.

A natural extension of this work is to automate the whole process, extracting a training set and building the model automatically. The problem is to extract a reasonably consistent shape-vector[1] from real training images containing examples of the object. A simple approach to this problem is described in this chapter. By processing large amounts of data, the effects of noise due to occlusion and mis-segmentation are reduced and a relatively simple segmentation scheme can be employed. In order to extract a large training set of shape-vectors, the processing of image data needs to be sufficiently fast. The system described has been implemented to run in near “real-time” (processing over 4 image frames per second). This allows the use of live video input to improve image quality.

[1] i.e. an ‘n’ dimensional vector that represents shape

The control points of a B-spline are used as a shape-vector, since a spline is convenient for data approximation and fast to render. Moreover, B-splines have successfully been used for tracking image contours (e.g. by Blake *et al* [38]). One of the advantages of this approach over the conventional PDM method is that there is no need to estimate positions of features that do not appear in a particular training image. For example, consider the training image in figure 3.1. A conventional PDM might label the boundary points at the elbow, hand, hip, knee, feet, etc with appropriate extra boundary points evenly spaced between these feature points. However, in the example image the left arm is not visible (due to self-occlusion) and estimating the appropriate feature points becomes difficult and prone to error. By regarding the silhouette as an abstract closed continuous shape (with no landmark features) an automatic procedure can be applied.

The model described here is essentially 2D but is trained on a selection of arbitrary views. The variation in shape due to different viewpoints is treated as flexibility in 2D shape, allowing the model to be used for tracking over the range of viewpoints for which it was trained.

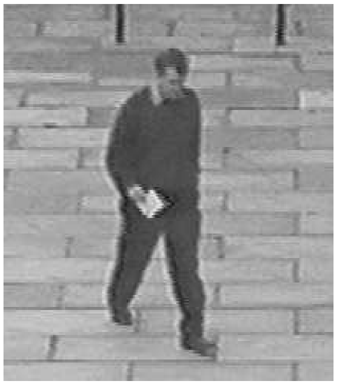


Figure 3.1: Example training image

3.2 Outline of the method

A system has been implemented to build a shape model automatically from real training images. The system takes live video images from a static camera, processes them and extracts fixed length shape-vectors representing the moving objects in the scene. The data is then analysed off-line to generate a model. A diagram illustrating this system is shown in figure 3.2.

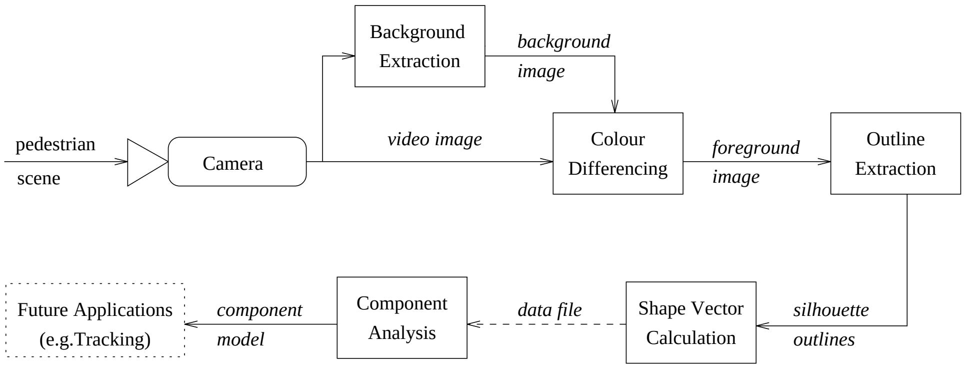


Figure 3.2: Overview of the system

There are four main stages:-

- *Image preprocessing* to obtain a binary background-foreground image.
- *Outline extraction* to obtain the boundary of each foreground shape.
- *Shape vector calculation* to obtain an item of training data.
- *Off-line analysis* to build the shape model.

3.3 Image Preprocessing

In order to segment the moving objects from a sequence of images, a background subtraction scheme similar to the method described in section 2.8 is used. The background image is continually updated (median filtering over time) to account for changing lighting conditions. An approximation to the median filter is used (kindly provided by Hyde and Worrall [45]). Two methods of image subtraction have been employed using grey-scale and colour images.

3.3.1 Grey-scale subtraction

Given a sequence of grey-scale images the moving objects are segmented using standard background subtraction. For a given image frame $I_k(x, y)$, a differenced image $\Delta I_k(x, y)$ is calculated by pixel-wise absolute subtraction from the reference background image $I_{\text{ref}}(x,y)$. i.e.

$$\Delta I_k(x,y) = |I_k(x,y) - I_{\text{ref}}(x,y)|$$

To reduce the effects of noise in the images, the differenced image is blurred using a standard Gaussian blur filter (see for example Gonzalez and Woods [21]) and the resulting blurred difference image $\Delta I'_k$ thresholded to produce a binary image, $B_k(x,y)$, where

$$B_k(x,y) = \begin{cases} \texttt{BACKGROUND} & \Delta I'_k(x,y) < \lambda_y \\ \texttt{FOREGROUND} & \Delta I'_k(x,y) \geq \lambda_y \end{cases}$$

The threshold value λ_y is chosen to be fairly low to ensure the foreground objects are well defined connected regions in the binary image, although this increases the effects of noise. These regions correspond to moving objects in the scene (in this case, walking pedestrians).

Results of this processing are shown in figure 3.3.

3.3.2 Colour subtraction

The additional information contained in colour images can be combined to improve the segmentation of moving objects. Image sequences were obtained in YUV format, which consists of one luminance field, Y and two chrominance fields U and V [2]. A background image was generated by treating each field independently, median filtering over time. For each pixel of an image frame the quantities Y_{img}, U_{img}, V_{img} and Y_{ref}, U_{ref}, V_{ref} were available i.e. the Y, U and V components of the current image and the reference (background) image. The differenced YUV values, ΔY, ΔU, ΔV are considered where

$$\begin{aligned} \Delta Y &= Y_{\text{img}} - Y_{\text{ref}} \\ \Delta U &= U_{\text{img}} - U_{\text{ref}} \\ \Delta V &= V_{\text{img}} - V_{\text{ref}} \end{aligned}$$

Under the null hypothesis that the current pixel is a "background pixel" the quantities ΔY, ΔU, ΔV are assumed to be sampled from independent zero-meaned, Gaussian distributions with

[2] This is the European equivalent to the YIQ standard as described in Foley and Van Dam [46]

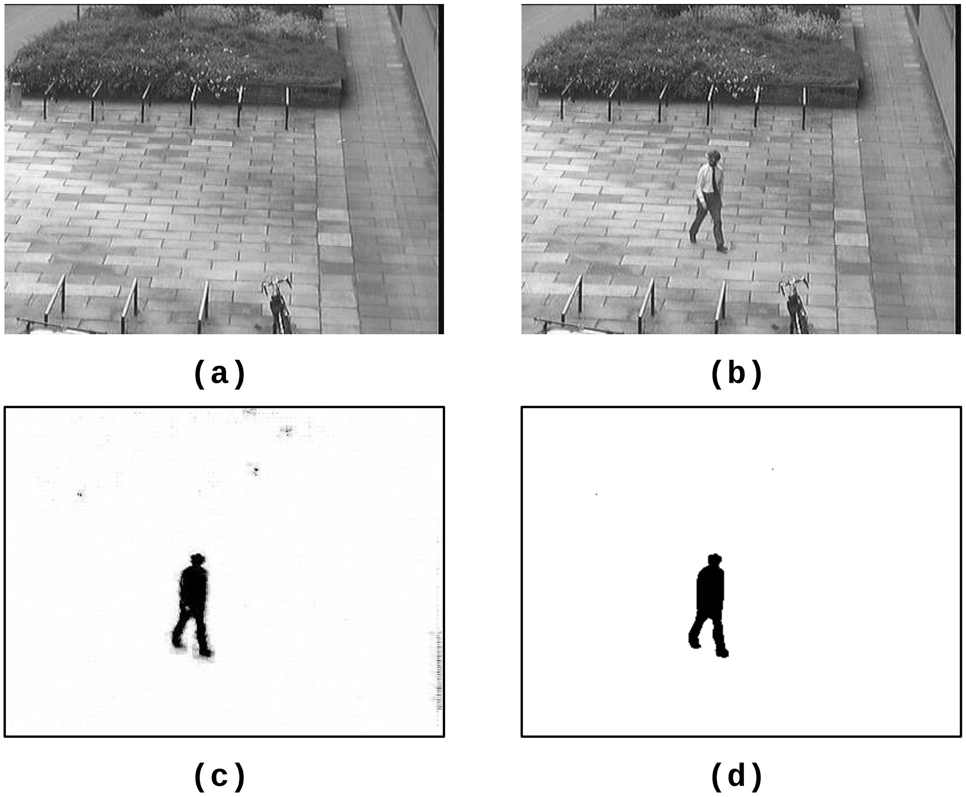


Figure 3.3: Image Preprocessing: (a) background image, (b) video input image, (c) differenced image, (d) blurred and thresholded image

variances σ_Y^2, σ_U^2 and σ_V^2. (A background pixel is assumed to have a fixed value with some normal random noise present due to errors in the imaging process).

Hence the quantity ΔS^2 is calculated for each pixel where

$$\Delta S^2 = \frac{\Delta Y^2}{\sigma_Y^2} + \frac{\Delta U^2}{\sigma_U^2} + \frac{\Delta V^2}{\sigma_V^2}$$

and the null hypothesis is rejected if $\Delta S^2 > \lambda_{\mathrm{yuv}}$ (and hence the pixel is assigned the value `FOREGROUND`). Otherwise the null hypothesis is accepted (i.e. the pixel is assigned the value `BACKGROUND`[3]). As for grey-scale subtraction, a conservative threshold is chosen for λ_{yuv} ensuring the foreground regions corresponding to moving objects are well defined and connected.

In order to improve robustness the image $\Delta S^2(x, y)$ is blurred with a Gaussian filter before thresholding.

This method requires estimates for the parameters σ_Y, σ_U and σ_V. These parameters are estimated from an initial image sequence where there is little or no movement. Values for ΔY, ΔU, ΔV are calculated as above over the whole image and the sample variance of each field used as the estimate for the variance of the underlying noise distribution.

3.3.3 Further noise reduction

When there is poor contrast between the moving object and the background, fragmentation can occur, resulting in several foreground regions where there should only be one connected region. This effect can be reduced by further image processing operations (at the expense of speed and resulting image resolution).

Morphological filters were applied to fill these "gaps" (see for example Sonka, Hlavac and Boyle [47]). In order to join regions separated by k pixels along an extended boundary, the following operations were performed on the binary image

- k successive dilation operations (i.e. region growing the `FOREGROUND` regions)
- k successive erosion operations (i.e. region shrinking the `FOREGROUND` regions)

[3] in fact, `FOREGROUND` $= 255$ and `BACKGROUND` $= 0$

3.4 Extracting silhouettes

The above image processing scheme generates a binary image in which every pixel where there is evidence of movement is set to FOREGROUND. Each connected FOREGROUND region is potentially the silhouette of a single moving object within the scene. The following object specific constraints may be utilised to reject regions which are unlikely to be a single pedestrian.

$$\begin{aligned} \texttt{NO\_PIXELS} &> \texttt{MIN\_REGION\_SIZE} \quad (3.1) \\ \texttt{NO\_PIXELS} &< \texttt{MAX\_REGION\_SIZE} \quad (3.2) \\ \texttt{REGION\_HEIGHT/REGION\_WIDTH} &< \texttt{MAX\_HEIGHT\_TO\_WIDTH} \quad (3.3) \\ \texttt{REGION\_HEIGHT/REGION\_WIDTH} &> \texttt{MIN\_HEIGHT\_TO\_WIDTH} \quad (3.4) \end{aligned}$$

where NO_PIXELS is the number of pixels in the region, REGION_HEIGHT is the height of the region's bounding box[4] and REGION_WIDTH is the width of the region's bounding box.

The first constraint (equation 3.1) removes small regions which are often due to noise "spikes" in the image data. Constraint 3.2 removes regions that are too large which may be the result of a change in lighting conditions. Constraints 3.3 and 3.4 ensure the region has a tall rectangular bounding box and removes regions where several moving pedestrians are amalgamated into one region or when the object is not a human (e.g. a car). The last two constraints are specific to images of pedestrians where the normal to the ground plane is roughly vertical in the image. These constraints are only required for noisy, cluttered training images where the moving objects are not necessarily of interest or where several moving objects may overlap in the image.

The connected FOREGROUND pixels are segmented from the binary image using a standard "flood-fill" algorithm (see for example Foley and Van Dam [46]). Feasible regions that satisfy the above constraints are traced (clockwise) to produce a chain of boundary points which is used as the basis for the calculation of a training shape-vector.

[4] i.e. minimum vertically aligned enclosing rectangle

3.5 Shape vector calculation

3.5.1 Finding a point of reference on the boundary

In order to proceed, a fixed reference point on the closed boundary (which will have an associated parameter value $u = 0$) is required. A consistent method is required which is not highly susceptible to noise.

The method used is to find the principal axis (i.e. the axis through the centroid of the boundary points which minimises the sum of the perpendicular distances to that axis). The reference point is chosen to be the upper (in terms of image coordinates) of the two points where the axis crosses the boundary. It is assumed that this point will be fixed for humans in the scene. This is reasonable for scenes where people always appear in an upright position.

A more general method may select the intersection point that is nearest to the centroid, or some other suitable choice. In the case where the principal axis may be inappropriate (e.g. vehicles viewed from the side and head on), a very simple method may use the upper-most point (in image coordinates) over the complete contour. This method is more suitable for training a model, specific to a fixed viewing angle (e.g. images of cars taken from a fixed camera).

The boundary points are now reordered so that the first point is the reference point and approximated by a cubic B-spline (for an example see figure 3.4). Each shape can be reflected about its principal axis to double the volume of training data (as has been done by Hill, Thornham and Taylor [48]).

3.5.2 Approximating with a cubic B-spline

The control points of a length-wise uniformly spaced B-spline are used as a shape vector. Previous steps extract from each moving shape an ordered set of n boundary points $\mathbf{W}_i = (X_i, Y_i)$, with $0 \leq i < n$ which are approximated with a (closed) spline $\mathbf{P}(u) = (P_x(u), P_y(u))$ with N control points $\mathbf{Q}_k = (R_k, S_k)$ where $N \ll n$. The function $\mathbf{P}(u)$ is expressed as follows:

$$\mathbf{P}(u) = \sum_{k=0}^{N-1} \mathbf{Q}_k B_k(u)$$

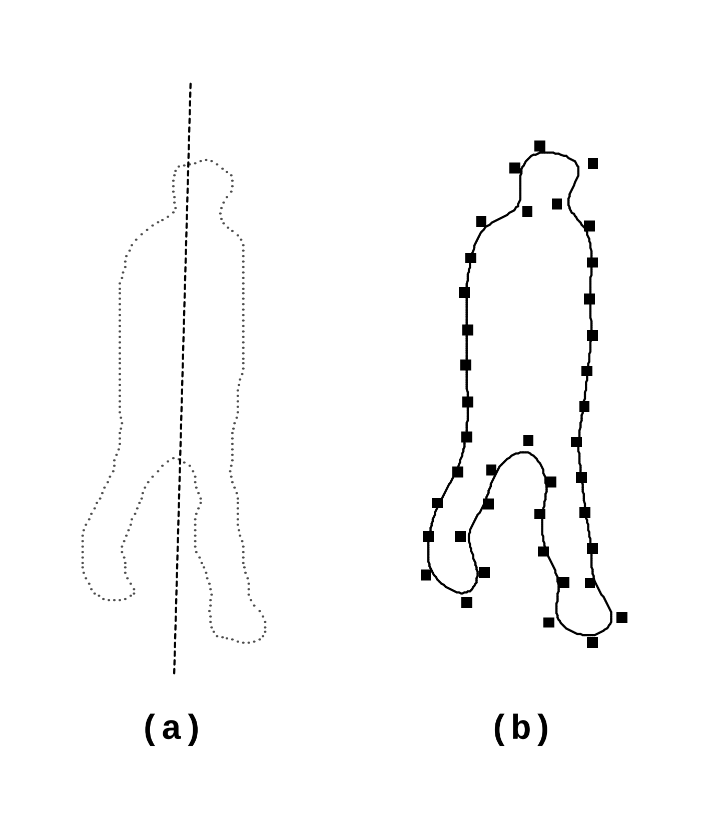


Figure 3.4: Extracting a spline:
(a) data points with principal axis,
(b) resulting spline.

where the B_k are modified B-spline basis functions. As the curves are closed the basis functions are defined such that $u = 0$ is equivalent to $u = N$ as follows:

$$B_k(u) = \begin{cases} \mathcal{B}(u-k) & (u-k) \geq 0 \\ \mathcal{B}(u+N-k) & (u-k) < 0 \end{cases}$$

where $\mathcal{B}(u)$ is the standard B-spline basis function which is non-zero in the interval $0 < u < 4$.

The required approximating spline minimises the error function, erf, given by

$$\mathrm{erf} = \sum_{i=0}^{n-1} (P_x(u_i) - X_i)^2 + (P_y(u_i) - Y_i)^2$$

where u_i is some parameter value associated with the i'th data point.

Using standard methods (see for example Bartels, Beatty and Barsky [49]) the following N equations are obtained:

$$\sum_{k=0}^{N-1} M_{i,k} R_k = \sum_{j=0}^{n-1} B_i(u_j) X_j \tag{3.5}$$

where $0 \leq i < N$ and $M_{r,s} = \sum_{k=0}^{n-1} B_r(u_k) B_s(u_k)$. An analogous set of equations are obtained for S_k. For a reasonably close approximation of the boundary, the parameter values can be set as follows:

$$u_k = \begin{cases} 0 & \text{for } k = 0 \\ \lambda \sum_{i=1}^{k} |\mathbf{W}_i - \mathbf{W}_{(i-1)}| & \text{for } k > 0 \end{cases} \tag{3.6}$$

where $\mathbf{W}_n \equiv \mathbf{W}_0$ and λ is chosen such that $u_n = N$.

To calculate the spline control points, $\mathbf{Q}_k$, the matrix $M_{i,j}$ must be inverted for each shape. In order to avoid this computationally expensive step, $n' = wN$ new data points are calculated (where w is a whole number, typically set to 8). These new data points correspond to the *fixed* uniformly spaced parameter values:

$$u_k \equiv k \left(\frac{N}{n'} \right)$$

For details, see section 3.5.3.

Using these new data points and their associated parameter values, $M_{i,j}$ is fixed and need only be inverted once. This efficiently produces a uniform B-spline with the control points placed at approximately uniformly spaced intervals along the contour. Moreover, the method is fast and robust.

The control points of the spline make up the shape vector $\mathbf{x}$, where

$$\mathbf{x} = (R_0, S_0, R_1, S_1, \ldots, R_N, S_N)^T$$

3.5.3 Selecting Data Points for Spline Approximation

Conventionally the parameter values associated with data points $\mathbf{W}_k$ are based on the Euclidean distances between points (as in equation 3.6). This leads to a set of values u_k corresponding to the data values X_k. The discrete mapping u_k to X_k can then be extended to a continuous mapping u to $X(u)$ by linear interpolation. Hence given $u_k \leq u \leq u_{k+1}$ it is possible to interpolate $X(u)$ using

$$X(u) = \left(\frac{u - u_k}{u_{k+1} - u_k}\right) X_{k+1} + \left(\frac{u_{k+1} - u}{u_{k+1} - u_k}\right) X_k$$

A similar interpolation scheme is used to find $Y(u)$. Hence given a chosen parametric value u, a corresponding new data point $(X(u), Y(u))$ is obtainable. Regularly spaced parametric values (between 0 and N) are chosen to find n' new data points. These new data points can now be efficiently approximated with a uniform cubic B-spline.

3.6 Component Analysis of the data

A straight forward method for analysing the training data has been implemented where the B-spline control points are treated in exactly the same way as the landmark points of the LPDM of Cootes *et al* , described in section 2.2. Hence the training shapes are aligned and a mean shape-vector calculated. A covariance matrix is calculated (using equations 2.1 and 2.2) and the eigenvectors calculated. The resulting model consists of the mean shape $\overline{\mathbf{x}}$ and a subset of m eigenvectors $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_m$ (of unit length) corresponding to the m most significant modes of variation in the training data.

A slightly modified method is described in section 3.9 which takes into account the nature of the training data (i.e. the fact that the shape-vectors are spline control points as opposed to landmark points). The required modification, although providing a sound theoretical basis for subsequent methods, does not in fact produce a dramatic change in the "modes of variation" visualised in section 3.7.

3.7 Results

3.7.1 Shape extraction from live video

Images were taken from 15 minutes of live video of a quiet pedestrian scene containing some moving vehicles. Training shapes were automatically segmented and approximated by a cubic B-spline with 40 control points. Each shape was reflected about the principal axis resulting in over 700 training shapes (corresponding to approximately 50 people). Some of these shapes are shown in figure 3.5. There are evident errors in the training shapes due to mis-segmentation. However, the majority of shapes are reasonably accurate and this large body of shapes dominates the subsequent statistical analysis.

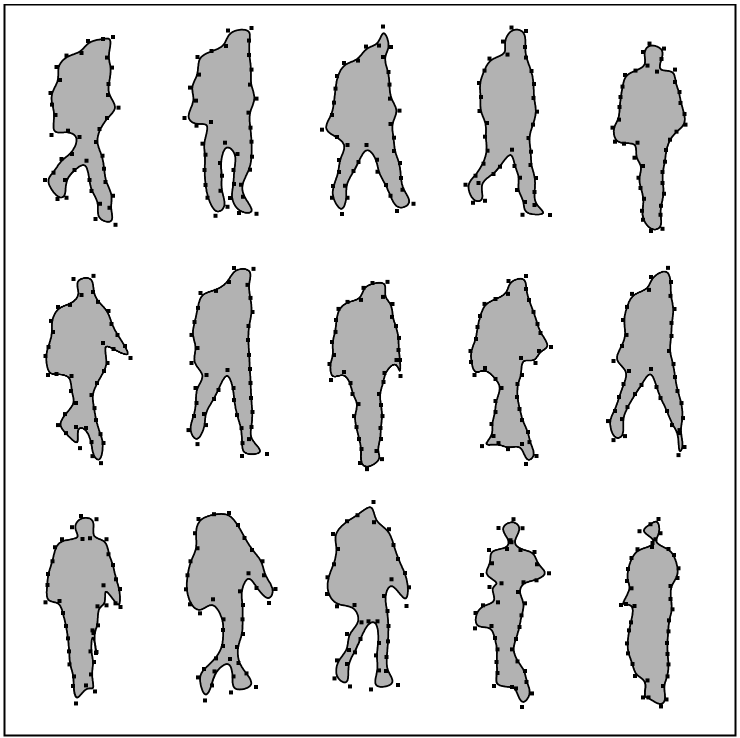


Figure 3.5: Some training shape-vectors

3.7.2 Modes of variation

Each training shape-vector had 80 parameters (40 control points in 2D). The first 18 modes accounted for 90% of the variance of the training data. The largest 19 eigenvalues are displayed in figure 3.6. The graph shows that there is a small set of significant eigenvalues and a larger set of relatively small eigenvalues. The small eigenvalues correspond to insignificant modes of variation that can be subsequently ignored.

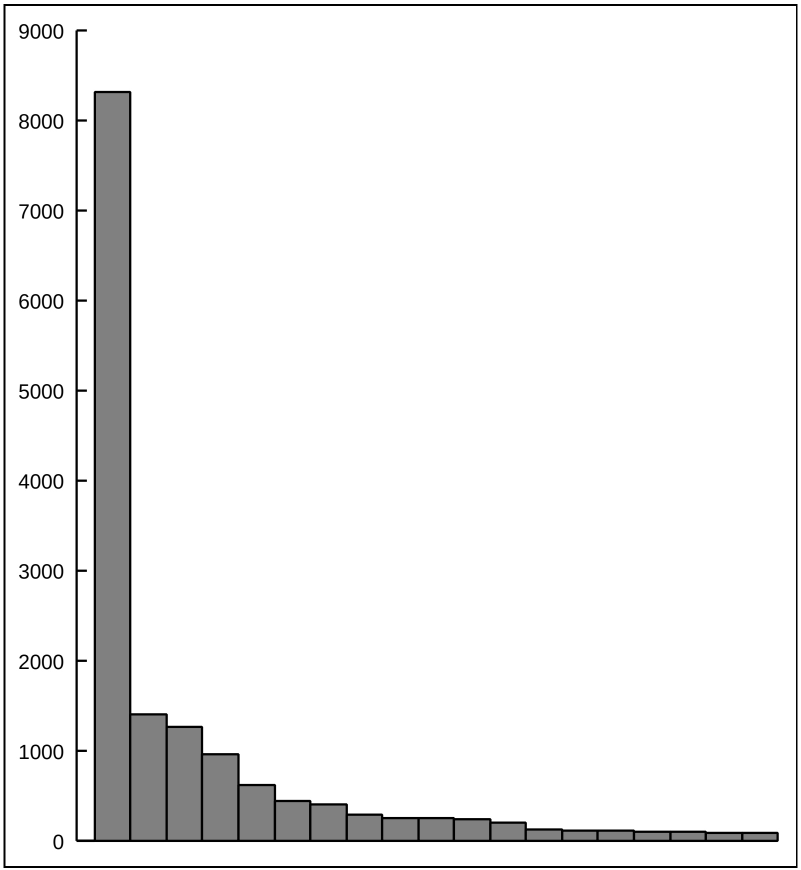


Figure 3.6: Plot of the first 19 eigenvalues

The first $m = 18$ eigenvectors can thus be used as an orthonormal basis for the model space of allowable shapes. Some of the significant modes of variation of the shape-vectors are shown in figures 3.7, 3.8 and 3.9.

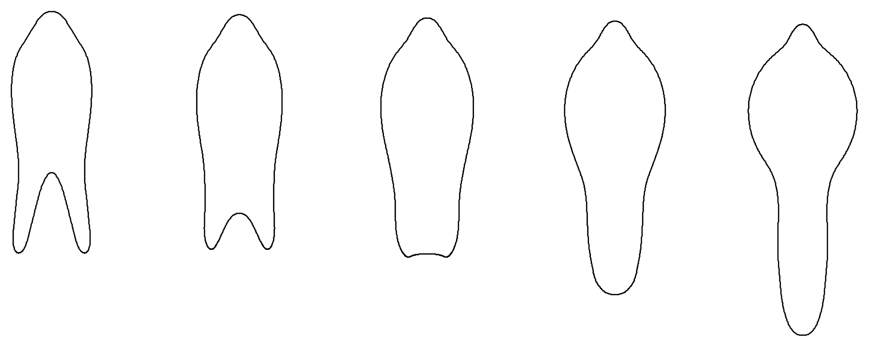


Figure 3.7: The effect of varying the component of the first mode by ±1.5 standard deviations

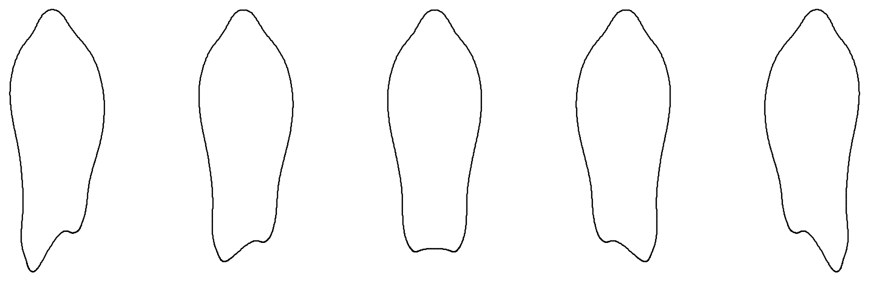


Figure 3.8: The effect of varying the component of the second mode by ±1.5 standard deviations

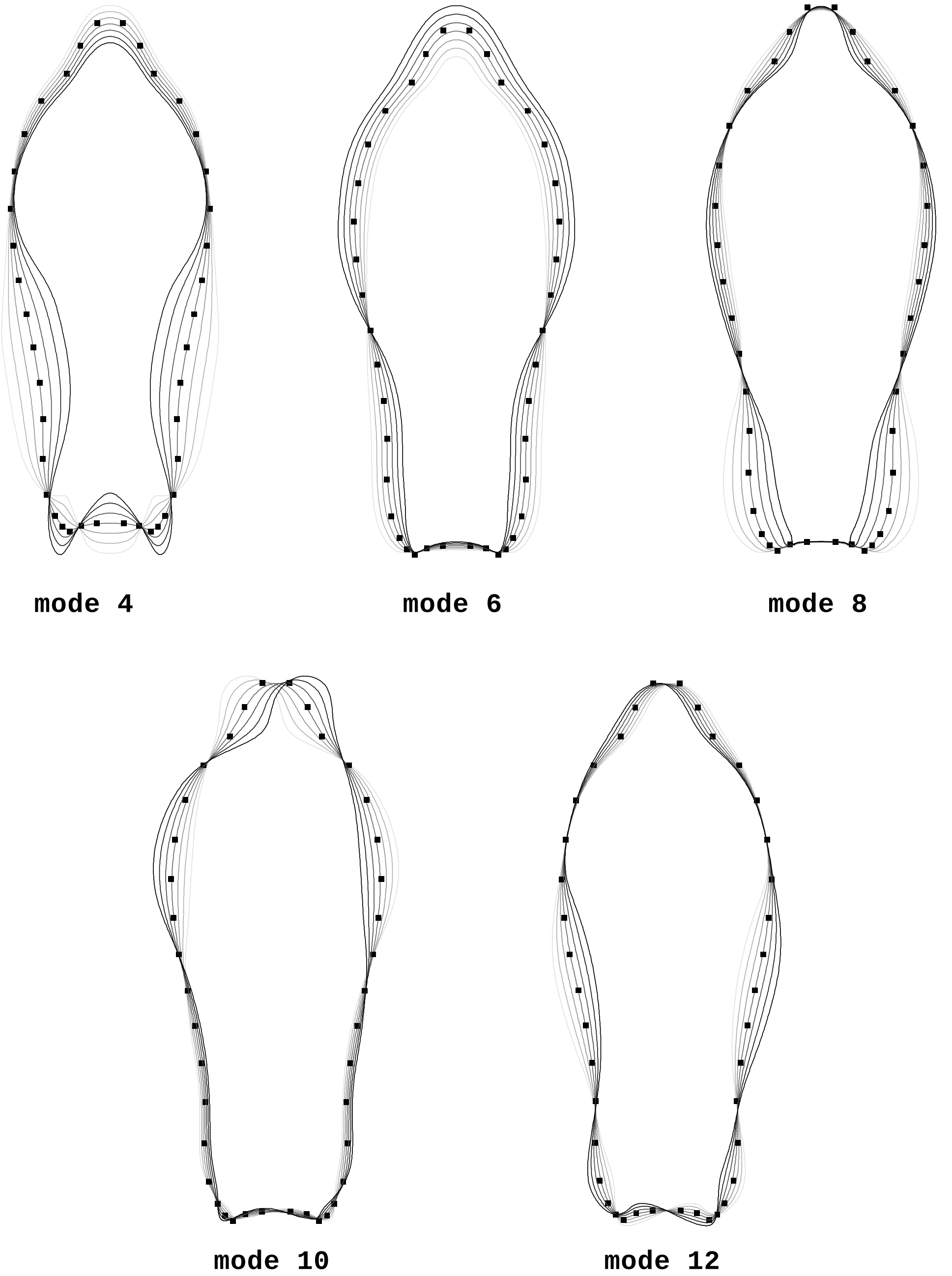


Figure 3.9: Diagrams illustrating some of the modes of variation

Visualisation of Modes

Each "mode of variation" represented by an eigenvalue and eigenvector corresponds to a line in shape-space through the mean shape. In order to visualise a particular mode, a small set of shape-vectors on this line are calculated by varying the associated shape parameter between suitable limits. Explicitly, for the i'th mode shape-vectors $\mathbf{x}^{(\mathrm{j})}$ are calculated using

$$\mathbf{x}^{(\mathrm{j})} = \overline{\mathbf{x}} + \mathrm{step}\left(\frac{j}{\sqrt{\lambda_i}}\right)\mathbf{e}_{\mathrm{i}}$$

where j varies between $-k$ and k (e.g. $j = -2, -1, 0, 1, 2$) and step is a suitable step size in standard deviations (typically around 0.5).

Each shape-vector represents a cubic B-spline and the splines are drawn either next to each other as in figure 3.7 or superimposed together (distinguished by rendering style) as in figure 3.9.

In figure 3.7 the mean shape is drawn in the centre of the diagram and in figure 3.9 the B-spline control points of the mean shape have been drawn in each diagram. The spline control points do not generally lie on the curve.

3.8 A simple application of the model

One very simple application of the linear model that has been generated is removing the effects of noise from a segmented shape. This can be done by projecting a shape-vector (obtained using the method described previously in this chapter) to the closest point in the *a priori* model space derived from the training set.

Hence given a noisy shape-vector $\mathbf{x}$, the m shape parameters $\mathbf{b} = (b_0, ..., b_{m-1})^T$ were calculated using equation 2.4. The shape parameters were further constrained so that $\mathbf{b}$ lies within a hyper-ellipsoid centered about the origin using equation 2.8. (The constant $s_{\max}$ was set to 16.0). The shape parameters were then projected back into the spline representation using equation 2.3 to get a "component-filtered" spline.

This process finds the closest point (with respect to the standard Euclidean distance metric) within the constrained model space to the noisy input shape. Results are shown on some real data in figure 3.10.

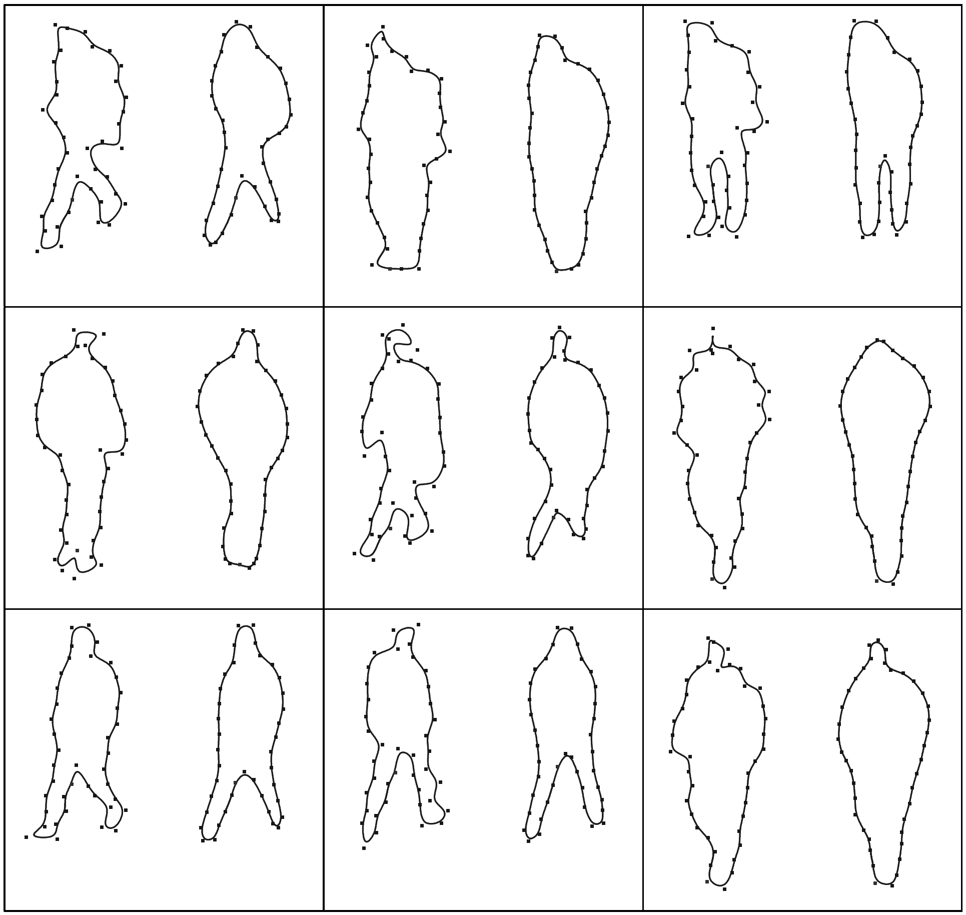


Figure 3.10: Projecting into the model space: In each case, the component filtered spline is shown to the right of the initial noisy input spline.

3.8.1 Limitations

The above method can be regarded as combining two noise reducing effects:-

1. Setting the components of the less significant modes of variation to zero. This is achieved by the first step of mapping into the space spanned by the significant modes.
2. Pulling the shape parameters towards the mean (when the shape is too far from the mean shape). This is achieved by constraining the vector b to lie within the hyper-ellipsoid. This takes a shape with low prior probability density to the closest point with a reasonably high prior probability density.

If the segmentation of the input shape is poor (e.g. a leg is missing) then all the control point positions will have significant errors resulting in large errors in all shape parameters. Thus there is insufficient information to reconstruct the original shape. Two examples of this problem are shown in figure 3.11.

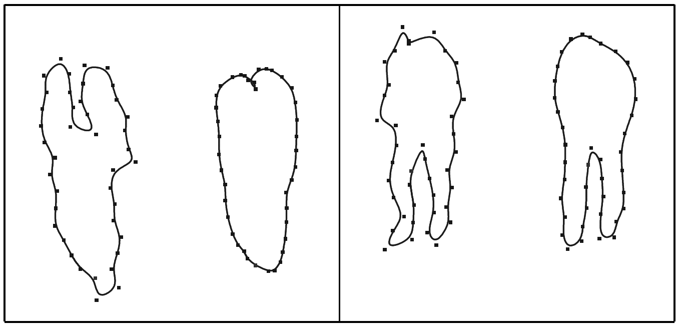


Figure 3.11: Projecting mis-segmented shapes

3.9 A modified component model

3.9.1 Principal Component Analysis

PCA aims to transform a correlated set of observed shape-vectors to a basis of linearly uncorrelated parameters. This is equivalent to diagonalising the shape-vector covariance matrix using a similarity transformation. The vector $\mathbf{dx} = (\mathbf{x} - \overline{\mathbf{x}})$ is transformed to a new basis using

$$\begin{aligned} \mathbf{dx} &= \sum_{i=0}^{2N-1} b_i \mathbf{e_i} \\ &= P\mathbf{b} \end{aligned} \tag{3.7}$$

where $\mathbf{b} = (b_0, ..., b_{2N-1})^T$ and $P_{jk} = [\mathbf{e_k}]_j$.

Assuming P is invertible the covariance matrix for $\mathbf{b}$ is simply

$$E(\mathbf{bb}^T) = P^{-1}E(\mathbf{dx}\,\mathbf{dx}^T)P^{-T}$$

In order to enforce linear independence, the above covariance matrix for $\mathbf{b}$ is diagonalised by appropriate choice of P^{-1}. This does not uniquely define P. A further orthogonality condition is required, namely

$$\mathbf{e_i} \cdot \mathbf{e_j} = \delta_{ij} \tag{3.8}$$

which is equivalent to $P^{-1} = P^T$.

3.9.2 Distance metric for splines

Equation 3.8 represents only one possible orthogonality condition. The scalar product corresponds to a choice of a standard Euclidean distance metric $f(...,...)$ to measure the error between two sets of landmarks (x_i, y_i) and (x'_i, y'_i) where

$$\begin{aligned} f(\mathbf{x}, \mathbf{x}') &= |\mathbf{x} - \mathbf{x}'| \\ &= \left(\sum_{i=0}^{N-1} (x_i - x'_i)^2 + (y_i - y'_i)^2 \right)^{\frac{1}{2}} \end{aligned}$$

Given two cubic B-splines $\mathbf{P}(u)$ and $\mathbf{P}'(u)$ defined by their N control points (x_i, y_i) and (x'_i, y'_i), a more accurate error metric d, measures the distances between corresponding points on each spline, sampled densely and uniformly over the parametric curves. i.e.

$$\begin{aligned} d(\mathbf{x}, \mathbf{x}') &= \left(\int_0^N |\mathbf{P}(u) - \mathbf{P}'(u)|^2 du \right)^{\frac{1}{2}} \\ &= \left(\int_0^N \sum_{i=0}^{N-1} \left((x_i - x'_i) B_i(u) \right)^2 du + \int_0^N \sum_{i=0}^{N-1} \left((y_i - y'_i) B_i(u) \right)^2 du \right)^{\frac{1}{2}} \end{aligned} \tag{3.9}$$

Equation 3.9 simplifies to the form

$$d(\mathbf{x}, \mathbf{x}') = [(\mathbf{x} - \mathbf{x}')^T \mathcal{M} (\mathbf{x} - \mathbf{x}')]^{\frac{1}{2}}$$

where the $2N \times 2N$ symmetric matrix $\mathcal{M}$ is defined by

$$\begin{pmatrix} \mathcal{M}_{2i,2j} & \mathcal{M}_{2i,2j+1} \\ \mathcal{M}_{2i+1,2j} & \mathcal{M}_{2i+1,2j+1} \end{pmatrix} = \begin{pmatrix} \mathcal{H}_{i,j} & 0 \\ 0 & \mathcal{H}_{i,j} \end{pmatrix} \tag{3.10}$$

and the $N \times N$ symmetric matrix $\mathcal{H}$ is given by

$$\mathcal{H}_{i,j} = \int_0^N B_i(u) B_j(u) du$$

There is a unique inner product associated with this metric given by

$$\langle \mathrm{x}, \mathrm{x}' \rangle = \mathrm{x}^T \mathcal{M} \mathrm{x}'$$

such that

$$d(\mathrm{x}, \mathrm{x}') = \langle \mathrm{x} - \mathrm{x}', \mathrm{x} - \mathrm{x}' \rangle^{\frac{1}{2}}$$

(see for example Cohn [50] for details on inner products). The inner product is used in place of the scalar product in equation 3.8 to give a more suitable orthogonality condition.

3.9.3 Eigenshape analysis

The desired transformation to a set of linearly independent $\mathcal{M}$-orthogonal eigenvectors is found by solving the eigenproblem

$$S\mathcal{M}\mathbf{e_i} = \lambda_i \mathbf{e_i} \tag{3.11}$$

where S is the training set covariance matrix $E(\mathbf{dx}\,\mathbf{dx}^T)$.

Using the notation of equation 3.7 the following results can be easily verified

1. The vectors $\mathbf{e_i}$ are orthogonal with respect to the inner product $\langle ..., ... \rangle$.
2. Hence by suitable normalisation

   $$\langle \mathbf{e_i}, \mathbf{e_j} \rangle = \delta_{ij}$$

   or equivalently $P^T \mathcal{M} P = I$.
3. Each shape coefficient b_i is given by projecting the shape-vector $\mathbf{dx}$ onto the line spanned by the i'th eigenvector (minimising the square distance d^2 to the line). i.e.

   $$b_i = \langle \mathbf{dx}, \mathbf{e_i} \rangle$$

4. The shape coefficients are linearly uncorrelated over the training set.

   $$\begin{aligned} E(b_i b_j) &= \mathbf{e_i}^T \mathcal{M} S \mathcal{M} \mathbf{e_i} = \langle \mathbf{e_i}, \lambda_j \mathbf{e_j} \rangle \\ &= \lambda_j \delta_{ij} \end{aligned}$$

5. Assuming an unbiased, homogeneous, isotropic Gaussian measurement noise model (with dense measurements uniformly spaced over the contour) as described by Blake *et al* [38], measurements for the shape parameters are uncorrelated (see section 4.2.4).

By analogy with equation 2.12 the eigenshape model can be regarded as a finite element system with mass matrix $\mathcal{M}$ and stiffness matrix S^{-1}.

3.10 Discussion

In this chapter a method for automatically generating a linear shape model from image sequences has been described. Results of an implementation have been shown for real image sequences of walking people. The system automatically extracts training shapes and labels these shapes using a B-spline representation.

By using a simple segmentation scheme to produce a large volume of noisy data a useful model of the human profile has been generated. By restricting the input domain to reasonable quality images from a fixed, colour video source, a model has been built which can be applied to less restricted problem domains.

An efficient method for extracting a shape vector based on a cubic B-spline has been demonstrated. The system can process large amounts of data in near real time to generate a compact data set. Statistical component analysis of the spline data gives a simple but effective model. A novel method for performing principal component analysis has been derived to provide a robust theoretical framework for statistical analysis of a training set of parametrised contours.

There are several advantages of using a B-spline contour to describe shape as opposed to a suitably dense set of “landmark” points (as in the LPDM). One advantage is that the representation provides an analytic characterisation of shape between nodes. This allows a relatively small number of nodes to be employed which reduces the computational expense of the eigen-analysis. The resulting eigenvectors are consequently of low dimensionality (e.g. 80 components) which reduces the amount of storage space required for the model (which is particularly important if the system requires many such models). Furthermore the spline representation will prove useful in tracking applications by allowing measurements to be made between nodes and providing an efficient method for calculating the normal to the curve at each point. Thus a large number of measurements can be taken (if desired), resulting in a robust over-determined system.

Chapter 4

Efficient Contour Tracking

4.1 Introduction

This chapter describes an efficient mechanism for tracking the model shape parameters described in chapter 3 (representing the outline of a deforming object, such as the silhouette of a walking pedestrian) through a sequence of images. The aim of the system is to track robustly one or more non-rigid objects in an outdoor scene in "real time" (i.e. processing images at 30 Hz) on modest hardware. The changes in shape between successive image frames captured at video rate can be significant and hence the contour can not be assumed to vary slowly. The tracking method must react well to large shape deformations but be simple enough to work in real-time. Sudden discontinuous changes in shape can occur where previously self-occluded features become visible. Noise and background clutter add to the difficulty of the task. In order to overcome these problems, the trained *a priori* shape model is used.

Cootes *et al* describe the "Active Shape Model" [15] (outlined in section 2.2.2) for locally updating shape parameters to fit features in an image. The method described here extends this work by incorporating a statistical framework similar to the tracking framework of Blake *et al* [38] (outlined in section 2.5), allowing the automatic control of spatial (and temporal) scale. A stochastic shape model is described allowing the contour to deform more easily in modes of variation that vary significantly within the training set. The statistical framework can be used to automatically control the search scale for feature search on an individual frame (in a similar manner to the multi-scale extension to the ASM of Cootes *et al* [24]) as well as over successive frames (allowing motion coherence to be exploited when "lock" has not been lost over the contour). A simple method is described to cope with known occlusion (e.g. when two tracked objects overlap in the image) improving the robustness of the system.

A significant advantage of using an *a priori* linear shape model over the "Active Spline" approach of Blake *et al* is that only a few shape parameters are required for tracking, improving the speed of the system. Furthermore, it can be shown that assuming a theoretical isotropic continuous sensor model, the filtering process for the shape parameters can be decoupled allowing each shape parameter to be filtered independently. In practice the (decoupled) system performs well, even when these assumptions are violated and a discrete (ansiotropic) measurement process used.

Results are included in this chapter, showing several pedestrians being tracked using images taken from a fixed camera, as well as the more difficult problem of tracking pedestrians in images taken with a hand-held moving camera.

4.2 Theoretical framework

4.2.1 State Space

The eigenshape analysis described in section 3.9 allows the vector x representing the 2D positions of N control points to be defined in terms of a set of m shape parameters $\mathbf{b} = (b_0, ..., b_{m-1})^T$ as follows:

$$\mathbf{x} = P\mathbf{b} + \overline{\mathbf{x}}$$

where P is an $2N \times m$ matrix of eigenvectors and $\overline{\mathbf{x}}$ is the mean shape-vector.

A contour in the model frame is projected into the image frame by rotation, scaling and translation using

$$\begin{pmatrix} X_i \\ Y_i \end{pmatrix} = Q \begin{pmatrix} x_i \\ y_i \end{pmatrix} + \begin{pmatrix} o_x \\ o_y \end{pmatrix}$$

where the 2 x 2 matrix Q is given by

$$Q = \begin{pmatrix} a_x & -a_y \\ a_y & a_x \end{pmatrix} = \begin{pmatrix} s\cos\theta & -s\sin\theta \\ s\sin\theta & s\cos\theta \end{pmatrix}$$

and the shape-vector

$$\mathbf{X} = (X_0, Y_0, ..., X_{N-1}, Y_{N-1})^T$$

represents the 2D control points of the B-spline contour in the image frame. Hence the state space consists of m shape parameters b_i, the origin of the object (o_x, o_y), and the alignment[1] parameters a_x, a_y, incorporating rotation and scaling. The state parameters are related to the shape-vector X by

$$\mathbf{X} = Q(a_x, a_y)(P\mathbf{b} + \overline{\mathbf{x}}) + \mathbf{o} \tag{4.1}$$

where

$$\mathbf{o} = \underbrace{(o_x, o_y, \ldots, o_x, o_y)}_{N\text{times}}^T$$

and Q is a $2N \times 2N$ rotation and scaling matrix given by

$$Q = \begin{pmatrix} \mathcal{Q} & & 0 \\ & \ddots & \\ 0 & & \mathcal{Q} \end{pmatrix}$$

4.2.2 Stochastic Model

Shape parameters

The shape part of the state vector is modeled as a simple discrete stochastic process as follows:

$$b_i^{(k)} = b_i^{(k-1)} + w_i^{(k-1)} \qquad w_i^k \sim N(0, \mu_i)$$

where b_i^k models the i'th parameter value at frame k and the noise term w_i^k is a zero-meaned, normally distributed random variable with variance μ_i . A dynamic model (assuming constant rate of change) was considered but found to be less stable with no appreciable improvement in performance. The underlying assumption of the shape model is that the shape parameters vary independently (the noise process is isotropic). This is reasonable as over the training set:

$$E(b_i b_j) = 0 \qquad i \neq j$$

[1] In this thesis, the term alignment refers to rotation and scaling but not translation.

As the variance of b_i over the training set is equal to λ_i, it is natural to set the noise terms using

$$\mu_i = \mu\lambda_i$$

where μ is an undetermined shape parameter and is typically set to 0.05. This allows the more significant shape modes to vary more freely. A diagram illustrating the resulting uncertainty ellipsoid from this stochastic model (assuming initial values are known with absolute certainty) is given in figure 4.1. Note that μ determines how easily the shape can deform with a value of $\mu = 0$ corresponding to complete rigidity.

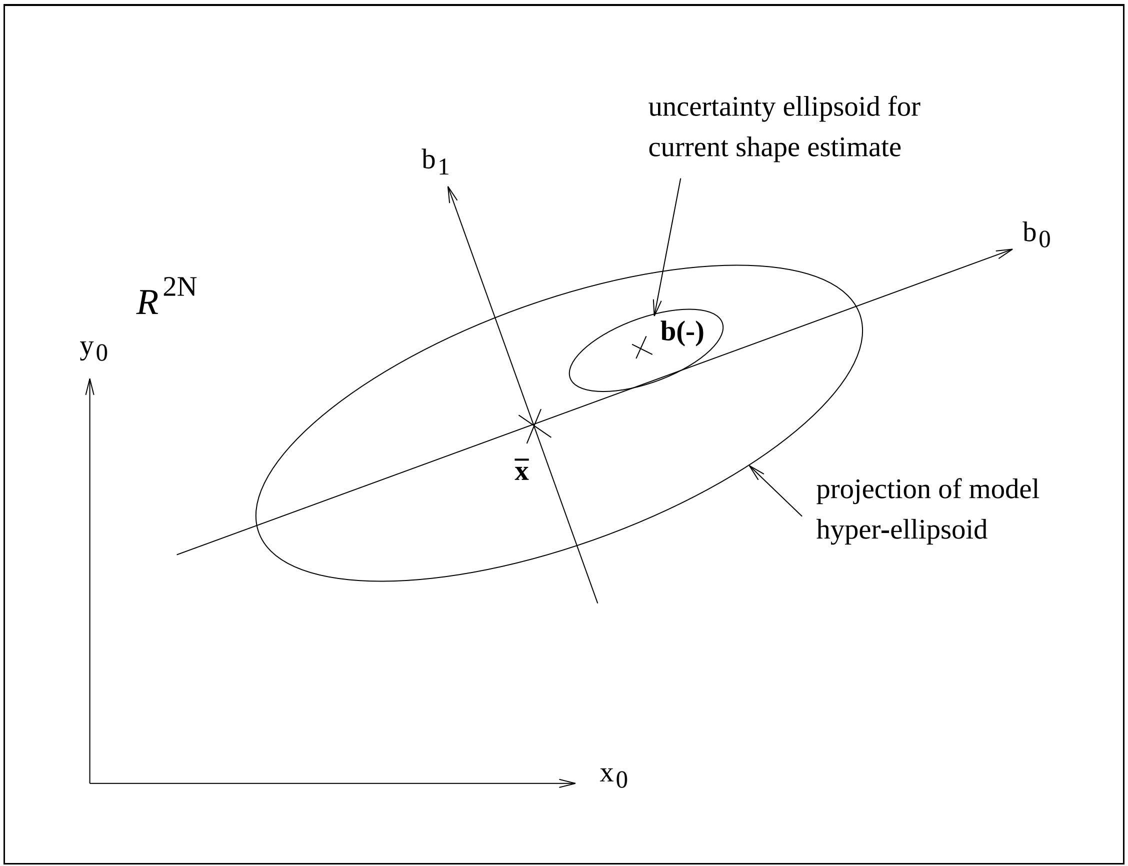


Figure 4.1: Diagram showing shape estimate uncertainty

Origin

The origin of the object is assumed to undergo uniform 2D motion with an additive random noise process (in both velocity and acceleration). This can be expressed by the differential equation:

$$\frac{d}{dt}\begin{pmatrix} o_x \\ \dot{o_x} \end{pmatrix} = \begin{pmatrix} \dot{o_x} \\ 0 \end{pmatrix} + \begin{pmatrix} v_x \\ w_x \end{pmatrix}$$

where $v_x \sim N(0, q_v)$ and $w_x \sim N(0, q_w)$. A corresponding model is used for o_y. Over a walk cycle, changes in shape affect the position of the origin. This can be accommodated by the random velocity term v_x, allowing the underlying "smooth" motion to be recovered. In the absence of sensor measurements this "smoothed" estimate of velocity determines the motion of the origin.

Alignment parameters

The alignment parameters a_x, a_y are assumed to be constant with added system noise as described by the equation:

$$\begin{pmatrix} a_x^{(k+1)} \\ a_y^{(k+1)} \end{pmatrix} = \begin{pmatrix} a_x^{(k)} \\ a_y^{(k)} \end{pmatrix} + \begin{pmatrix} w_{ax} \\ w_{ay} \end{pmatrix}$$

where $w_{ax}, w_{ay} \sim N(0, q_a)$.

4.2.3 The Discrete Kalman Filter

The standard discrete Kalman filter may be used to update state (and covariance) estimates of a system with discrete measurements at regularly spaced time intervals $t = k\Delta t$ (see, for example, Gelb [32]). For a standard measurement model,

$$\mathbf{z}_k = H_k\mathbf{x}_k + \mathbf{v}_k \quad \mathbf{v}_k \sim N(0, R_k)$$

(i.e. with measurement matrix H_k and measurement covariance matrix R_k), the state estimate update is given by

$$\hat{\mathbf{x}}_k(+) = \hat{\mathbf{x}}_k(-) + K_k(\mathbf{z}_k - H_k\hat{\mathbf{x}}_k(-))$$

The Kalman gain K_k is given by

$$K_k = P_k(+)H_k^T R_k^{-1}$$

and the covariance matrix P_k update is given by

$$P_k^{-1}(+) = P_k^{-1}(-) + H_k^T R_k^{-1} H_k \tag{4.2}$$

Note that the covariance matrix, denoted P_k is distinct from the matrix of eigenvectors denoted P.

4.2.4 Theoretical basis for decoupling shape filter

Blake *et al* [38] describe a theoretical continuous sensor model for measuring a B-spline contour ("feature") in an image. The sensor is assumed to be unbiased, homogeneous, isotropic and Gaussian. In Blake's notation the N control points of the B-spline are represented by the joint state-space vector $(\mathbf{X}(t), \mathbf{Y}(t))$ and the sensor measures the least squares approximation $(\mathbf{X_f}, \mathbf{Y_f})$ to the continuous curve. The $N \times N$ covariance matrix for the measurement process for each of the $\mathbf{X_f}, \mathbf{Y_f}$ measurements is given by

$$R_X = R_Y = r\mathcal{H}^{-1}$$

where $\mathcal{H}$ is the matrix defined in equation 2.14.

In terms of the shape-vector notation of this thesis, the sensor measures the observed shape-vector $\mathbf{X_{obs}}$ with $2N \times 2N$ covariance matrix R_k given by

$$R_k = r\mathcal{M}^{-1}$$

where $\mathcal{M}$ is the $2N \times 2N$ matrix defined in equation 3.10 and r is a scalar.

If the alignment and origin parameters are assumed to be fixed and the m shape parameters are filtered using a discrete Kalman filter (with measurements taken at each image frame), then from equation 4.1 and equation 4.2, the covariance matrix update equation is given by

$$P_k^{-1}(+) = P_k^{-1}(-) + [QP]^T [r\mathcal{M}^{-1}]^{-1} [QP]$$

The above update equation is simplified using the following easily obtainable results

- The alignment matrix, Q, commutes with the "metric" matrix $\mathcal{M}$. i.e. $Q\mathcal{M} = \mathcal{M}Q$

- The alignment matrix is a scaled rotation matrix. Hence $Q^T Q = s^2 I$
- The matrix of eigenvectors P was derived such that $P^T \mathcal{M} P = I$ (see section 3.9.3)

Using these results, the update equation simplifies to

$$\begin{aligned} P_k^{-1}(+) &= P_k^{-1}(-) + s^2 r^{-1} P^T M P \\ &= P_k^{-1}(-) + s^2 r^{-1} I \end{aligned}$$

Hence assuming $P_k(-)$ is diagonal, then after applying the measurement $\mathbf{X}_{\mathrm{obs}}$ the updated covariance matrix is still diagonal. Assuming P_0 is diagonal and noting the diagonal form of the stochastic shape model described in section 4.2.2, the covariance matrix is always diagonal. Thus the system can be decoupled into m independent 1D Kalman filters [2]. The covariance update equation for the i'th filter becomes

$$[\sigma_i(+)]^{-1} = [\sigma_i(-)]^{-1} + r_i^{-1} \tag{4.3}$$

where $r_i^{-1} = s^2 r^{-1}$ and $\sigma_i = [P_k]_{i,i}$ is simply the variance of the current estimate for b_i.

The corresponding shape parameter update equation is given by

$$\hat{b}_i(+) = \hat{b}_i(-) + \left(\frac{\sigma_i(-)}{r_i + \sigma_i(-)} \right) db_i$$

where

$$db_i = [P^T Q^T \mathbf{X}]_i - \hat{b}_i(-)$$

is the observed change in the i'th shape parameter. Note the similarity to Lowe refinement (equation 2.10). In the absence of previous measurements, the stochastic shape model will result in the variance $\sigma_i(-)$ being directly proportional to the eigenvalue λ_i, and the Lowe refinement shape update becomes almost identical to the Kalman filter update for the theoretical isotropic sensor model.

[2] i.e. a filter with a 1 dimensional state space

4.2.5 Discrete Measurement Model

Observed features

Although the object shape is represented by a continuous curve, it is convenient to sample the curve at L regular intervals between control points. Hence there are $n = NL$ points (p_i, q_i) given by

$$\mathbf{p} = G\mathbf{X}$$

where

$$\mathbf{p} = (p_0, q_0, \ldots, p_{n-1}, q_{n-1})^T$$

and G is a $2n$ x $2N$ sparse matrix mapping the control points to regularly spaced points on the curve, i.e.

$$\sum_{j=0}^{N} \begin{pmatrix} G_{2i,2j} & 0 \\ 0 & G_{2i+1,2j+1} \end{pmatrix} \begin{pmatrix} X_j \\ Y_j \end{pmatrix} = \begin{pmatrix} p_i \\ q_i \end{pmatrix}$$

Note that G commutes with the rotation matrix Q.

At each new frame, measurements are made by searching along the normal to the estimated contour at some or all of the sample points. The search is restricted to a specified search window obtained from the filtering process (see section 4.3.5). The point of maximum contrast is retained as the observed feature. The contrast is measured at the search scale and for reasons of speed only 3 points along the normal are examined: on the curve and at the extremes of the search window. This method is described by Blake *et al* [38] and was summarised in section 2.5.

Measurement covariance matrix

For each point measurement there is an associated measurement variance v_i which is set proportional to the square size of the search window at that point. Hence, if there are a total of $L'N$ measurements made within a unit frame-period, the pointwise measurement variance is given by

$$v_i = L'(c\rho_i)^2$$

where ρ_i is the size of the search window at the i'th sample point and c is a constant (typically set to 0.5). If there is no significant point of contrast found within the search window (the feature has been lost) then no measurement is made. This is achieved by setting v_i to infinity (i.e. $v_i^{-1} = 0$).

The "aperture problem", described by Horn [51], allows only the normal component of the displacement of the contour to be measured. Thus measurements are made by searching along the normals $\mathbf{n_i}$ to the estimated contour at each sample point. This results in coupling in the x and y components of the measurements. The inverse covariance matrix is given by the partitioning:

$$R_k^{-1} = \begin{pmatrix} A_0 & & 0 \\ & \ddots & \\ 0 & & A_n \end{pmatrix}$$

where A_i is the 2 x 2 pointwise inverse covariance matrix given by

$$A_i = v_i^{-1} \mathbf{n_i} \mathbf{n_i}^T$$

4.3 Tracking Filter

The point measurements are related to the state space parameters by the equation

$$\mathbf{p} = Q(a_x, a_y)G[P\mathbf{b} + \overline{\mathbf{x}}] + G\mathbf{o}$$

which is essentially non-linear (due to the dependence of Q on a_x and a_y). In the interests of speed, the shape, alignment and translation effects are filtered separately using the following scheme:

1. Assume the shape and alignment parameters are fixed
2. Estimate the change in origin using a dynamic Kalman filter
3. Remove the effects of this origin shift from the observations
4. Estimate the change in alignment parameters
5. Remove the effects of change in alignment
6. Update each shape parameter estimate independently using a 1D Kalman filter

If the effects of change in alignment are sufficiently small, the shape, alignment and translation effects can be filtered in parallel, ignoring changes in alignment and translation in the shape filter mechanism (i.e. omitting steps 3 and 5).

4.3.1 Updating the Origin

The x and y components of the origin are filtered independently. The measurement model for the x component of the origin, assuming the other parameters are fixed at their current estimates, is given by

$$p'_i = o_x + (\mathbf{v_k})_{2i}$$

where the noise term $\mathbf{v}_k \sim N(0, R_k)$ and similarly for the y component

$$q'_i = o_y + (\mathbf{v_k})_{2i+1}$$

The "measurements" $\mathbf{p}' = (p'_0, q'_0, ...)$ are calculated from the observed contour points $\mathbf{p}$ using

$$\mathbf{p}' = \mathbf{p} - Q(\hat{a}_x, \hat{a}_y)G(P\hat{\mathbf{b}} + \overline{\mathbf{x}})$$

For the x component filter, the 2×2 covariance matrix P_{o_x} for the state estimate $(\hat{o}_x, \hat{\dot{o}}_x)$ is updated using the standard Kalman filter equations. Explicitly

$$[P_{o_x}(+)]^{-1} = [P_{o_x}(-)]^{-1} + \begin{pmatrix} r^{-1} & 0 \\ 0 & 0 \end{pmatrix}$$

where $r^{-1} = \sum_{i=0}^{n-1} (R_k^{-1})_{2i,2i}$.

Between image frames the standard Kalman filter equations are used to obtain the estimated origin at the next frame.

4.3.2 Updating the Alignment

If the origin and shape parameters are fixed at their current estimates, the measurement model for the alignment parameters is given by

$$\mathbf{p} - G\hat{\mathbf{o}} = H \begin{pmatrix} a_x \\ a_y \end{pmatrix} + \mathbf{v_k}$$

where H is the $2n$ x 2 measurement matrix defined by

$$\begin{pmatrix} H_{2i,0} & H_{2i,1} \\ H_{2i+1,0} & H_{2i+1,1} \end{pmatrix} = \begin{pmatrix} \mathrm{s}_{2i} & -\mathrm{s}_{2i+1} \\ \mathrm{s}_{2i+1} & \mathrm{s}_{2i} \end{pmatrix}$$

where $\mathbf{s} = G(P\hat{\mathbf{b}} + \overline{\mathbf{x}})$.

The estimates $\hat{a}_x$, $\hat{a}_y$ and the 2 x 2 covariance matrix are updated with the corresponding Kalman filter equations. The alignment parameters are not assumed to be independent although for simplicity the system noise is assumed isotropic.

4.3.3 Updating the Shape parameters

The theoretical isotropic sensor model results in a decoupled Kalman filter. This provides a theoretical motivation for filtering each shape parameter independently, even when the ansiotropic discrete measurement process is used. In order to achieve this decoupling, the covariance matrix for the m shape parameters is restricted to be diagonal by ignoring off-diagonal elements in the covariance update equation.

Writing $\Delta\mathbf{p} = \mathbf{p} - \hat{\mathbf{p}}$, the measurement model for the i'th shape filter is given by

$$\Delta\mathbf{p} = \mathbf{h^{(i)}}(b_i - \hat{b_i}) + \mathbf{v_k}$$

where the vector $\mathbf{h^{(i)}}$ is an $2n \times 1$ measurement matrix given by

$$[\mathbf{h^{(i)}}]_j = [Q(\hat{a}_x, \hat{a}_y)GP]_{ji}$$

The covariance update equation for each filter is given by equation 4.3 where the "measurement variance" for the i'th shape parameter, r_i, is now defined by

$$r_i^{-1} = (\mathbf{h^{(i)}})^T R_k^{-1}\mathbf{h^{(i)}}$$

The state update equation for each filter is given by

$$\hat{b}_i(+) = \hat{b}_i(-) + \sigma_i(+)((\mathbf{h^{(i)}})^T R_k^{-1}(\Delta\mathbf{p}))$$

4.3.4 Enforcing the global shape constraint

Cootes *et al* constrain the model space of feasible shapes by ensuring the vector $\mathbf{b}$ lies within a hyper-ellipsoid (so that the Mahalanobis distance to the mean shape is constrained). This constraint can be applied using equation 2.8 after the shape parameters have been updated. This method has been implemented with some success. An alternative method, which produces a similar increase in stability, has also been implemented. A virtual input of 0 is applied to each shape filter at the start of each image frame with measurement variance for each shape parameter proportional to λ_i. This approach has several advantages.

- In the absence of image measurements (e.g. due to occlusion) the variance of each shape parameter estimate will rapidly increase (due to the stochastic shape model). The virtual input ensures each variance is bounded. This is valid, because the object shape is assumed to have come from the same (Gaussian) distribution as the training data. Hence the virtual input adds prior knowledge to the system.
- The virtual input will "pull" the solution towards the mean shape before image measurements are made. This discourages *a priori* unlikely solutions but does not prevent them if there is strongly supporting image evidence.
- These techniques can be combined by applying the virtual input at the start of each frame and the shape constraint after applying image measurements.

4.3.5 Automatic control of search scale

The Kalman filter provides a mechanism for automatically setting the search scale (as demonstrated by Blake *et al* [38]). The search window size at the i'th sample point is related to the positional variance $V(p_i)$ and $V(q_i)$ at the estimated contour point given by

$$\begin{aligned} V(p_i) &= [(QGP)P_k(QGP)^T]_{2i,2i} + V(o_x) \\ &= \sum_{j=0}^{l-1}((QGP)_{2i,j})^2\sigma_j + V(o_x) \end{aligned}$$

where $V(o_x)$ denotes the variance of the estimate $\hat{o}_x$ (and a similar equation is obtained for q_i). For simplicity, the alignment matrix Q is assumed constant in this calculation. An elliptical search window is used with semi-axes of length $2\sqrt{V(p_i)}$ and $2\sqrt{V(q_i)}$. Hence the search scale ρ_i along the normal $\mathbf{n_i}$ is given by

$$\rho_i = 2\sqrt{\frac{V(p_i)V(q_i)}{(\mathbf{n_i})_x^2 V(q_i) + (\mathbf{n_i})_y^2 V(p_i)}}$$

4.4 Implementation

4.4.1 Iterative Scheme

An iterative filter has been implemented so that the contour shape is refined several times for each frame. In order to improve the speed of the tracking mechanism, a subset of the NL sample points is used at each iteration. The method picks a random starting sample point and $(n_{\mathrm{sub}} - 1)$ additional evenly spaced points. The measurements are combined using the updating scheme described previously to find improved estimates for the state parameters. Subsequent iterations take sets of n_{sub} measurements using the current estimates to calculate the estimated point positions and search scale. A diagram illustrating this scheme is shown in figure 4.2.

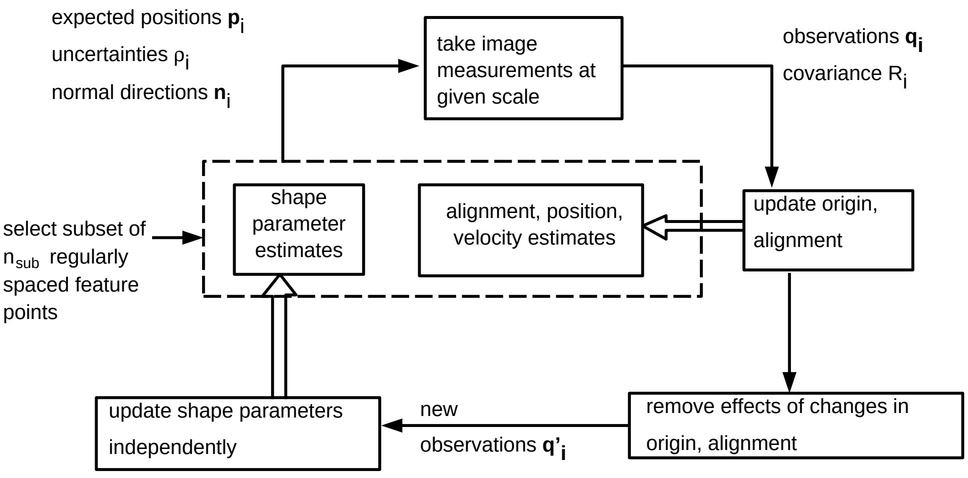

expected positions $\mathbf{p}_i$
uncertainties ρ_i
normal directions $\mathbf{n}_i$
take image measurements at given scale
observations $\mathbf{q}_i$
covariance R_i
select subset of n_{sub} regularly spaced feature points
shape parameter estimates
alignment, position, velocity estimates
update origin, alignment
update shape parameters independently
new observations $\mathbf{q}'_i$
remove effects of changes in origin, alignment

Figure 4.2: Diagram illustrating tracking filter mechanism

This mechanism is essentially a multi-scale search technique where the search scale is automatically controlled by the Kalman filter mechanism. The scheme allows the rough contour shape and position to be found quickly so that subsequent measurements of a particular contour point are more likely to lock on to the correct image feature. The choice of n_{sub} is a compromise between a minimum value corresponding to the total number of state parameters (as the measurements are coupled, each point measurement constrains only one free parameter) and a suitably large value corresponding to a "dense" set of measurements (ensuring the state update mechanism is overdetermined and hence robust).

4.4.2 Initialisation

The tracking mechanism requires initial estimates for the state parameters for each tracked object. In this implementation a crude motion detector is used using background subtraction on a subsampled image. The camera is assumed to be fixed (at this initialisation step) and in the interests of speed the noise reduction step described in chapter 3 is not carried out. The result of this processing is a binary "differenced image" where the foreground pixel regions correspond to moving objects in the scene. Objects that are already being tracked are removed from this image by clearing the bounding box of the tracked object in the binary image. The remaining significantly sized (in terms of numbers of pixels) connected components are assumed to correspond to new moving objects.

For each of these connected components the bounding box is calculated and the state parameters are initialised as follows

$$\begin{aligned}
\hat{a}_x(0) &= (y_r - y_l)/h_m \\
\hat{a}_y(0) &= 0 \\
\hat{o}_x(0) &= \frac{1}{2}(x_l + x_r) \\
\hat{o}_y(0) &= \frac{1}{2}(y_l + y_r) \\
\hat{\mathbf{b}}(0) &= \mathbf{0}
\end{aligned}$$

where the bounding box has a lower left-hand corner (x_l, y_l) and an upper-right corner (x_r, y_r). The constant h_m is the height of the mean shape. The initial variance of each shape parameter estimate is set to the associated eigenvalue λ_i.

Hence the estimated shape is initialised to the mean shape aligned vertically, centred at the origin of the bounding box and scaled to the height of the bounding box.

4.4.3 Measuring contrast

Two measures of contrast have been used to drive the tracking mechanism. The first assumes the camera is fixed and a reference background image has been calculated. The second uses a single image frame allowing the camera to be non-stationary.

Fixed camera method

A background (reference) image is calculated (at full resolution) in the usual manner. The contrast measure essentially looks at the intensity gradient of the differenced image (without thresholding) at the desired search scale. Rather than performing image differencing and edge operations on the whole image at multiple scales, the system only measures contrast where required by the tracking mechanism (allowing real time performance without image processing hardware).

The tracking mechanism requires a contrast measure at 3 points along the normal $\mathbf{n_i}$ through the estimated point $\mathbf{p_i}$ using the scale ρ_i. Defining the points $\mathbf{p}^{(k)}$ as follows

$$\mathbf{p}^{(k)} = \mathbf{p_i} + k\rho_i\mathbf{n_i} \tag{4.4}$$

contrast is measured at $\mathbf{p}^{(1)}$, $\mathbf{p}^{(0)}$ and $\mathbf{p}^{(-1)}$. The contrast c_k at $\mathbf{p}^{(k)}$ is measured by examining image intensities at $\mathbf{p}^{(k+1)}$ and $\mathbf{p}^{(k-1)}$ using

$$c_k = |I(\mathbf{p}^{(k-1)}) - I_{\text{ref}}(\mathbf{p}^{(k-1)})| - |I(\mathbf{p}^{(k+1)}) - I_{\text{ref}}(\mathbf{p}^{(k+1)})|$$

where $I(\mathbf{q})$ is the image intensity at the nearest pixel to the real-valued coordinates $([\mathbf{q}]_x, [\mathbf{q}]_y)$ and I_{ref} is similarly the intensity for the background image.

Note, the contrast measure is signed such that positive contrast corresponds to a larger absolute image difference at the inward facing sample point and a smaller absolute image difference at the outward facing sample point (see figure 4.3). The contour is attracted to points with high (positive) contrast so that the whole curve lies on the boundary of a moving region.

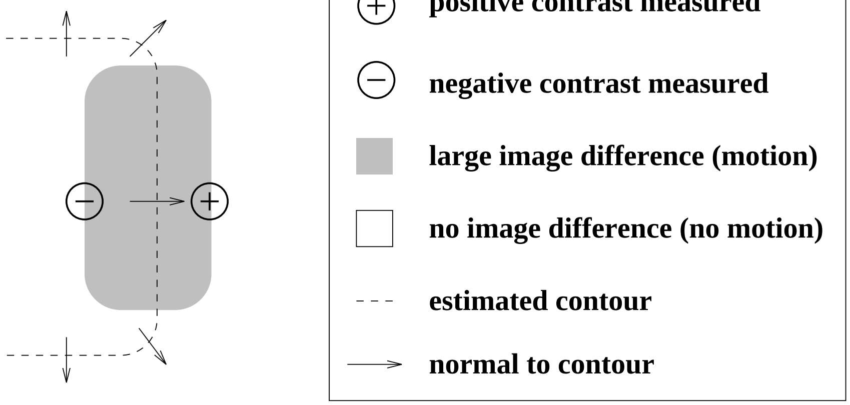

⊕ positive contrast measured
⊖ negative contrast measured
large image difference (motion)
no image difference (no motion)
estimated contour
normal to contour

Figure 4.3: Diagram showing signed contrast measure

Edge-based method

When the camera is non-stationary, an edge-based "contrast" measure is necessary. A 3×3 sobel edge filter on a subsampled grid is used. Horizontal and vertical sobel edge filters are applied and the resulting edge strength along these two directions projected along the normal direction. The absolute value of the edge strength along the normal direction is used. For each contrast measurement image intensities are sampled in a regularly spaced grid centred about the closest pixel to the (real-valued) position $\mathbf{p}^{(k)}$, defined in equation 4.4. The grid spacing is the closest integer to the search scale ρ_i.

This method is suitably fast and robust for the purposes of tracking in scenes with reasonable contrast. For colour input, contrast is measured on each colour field and the resulting edge strengths summed over all the fields.

4.4.4 Modeling Occlusion

In order to increase the robustness of the system in more difficult scenes where there are several objects being tracked, occlusion can be modeled. The method used is based on the work of

Koller *et al* [52]. In their work, it is assumed that nearer objects appear lower in the image plane and occlude farther away objects. Measurements that occur within known regions of occlusion are ignored, improving the robustness of object tracking.

A similar method is employed here, with the following simplifications:-

- Object regions that overlap are assumed to occlude one another (i.e. no depth assumption is used).
- An enlarged bounding box is used instead of an enlarged contour to model the object in the image plane (for the purposes of occlusion reasoning). This simplification reduces the computational burden of the occlusion reasoning.

Hence, at the start of each new image frame the currently tracked objects are drawn into an "occlusion image" using an enlarged bounding box centred about the estimated origin of each object. The height and width are set to 105% of the height and width of the contour's bounding box in the previous frame. Each rectangle is rendered with a new pixel value except where two or more rectangles overlap in which case a particular value is used to flag the pixel as being "occluded".

When measurements are made in the tracking mechanism at an estimated contour point $\mathbf{p_i}$, the associated pixel in the occlusion image is checked. If there is possible occlusion no measurement is made at that point (i.e. the measurement inverse variance is set to zero). This increases the overall measurement variance for each state parameter reducing the Kalman gain and increasing the state parameter uncertainties. The method is found to improve robustness where there is partial occlusion of tracked pedestrians.

4.5 Results

4.5.1 Quantitative Analysis

In order to measure the performance of the tracking system, eight reasonable quality image sequences were obtained in which the same person is seen walking in eight different directions relative to a fixed camera. A background image was also captured in which there are no moving objects. The images are of sufficiently good contrast to obtain an accurate segmentation of the pedestrian's silhouette using image subtraction as described previously. The resulting set of shapes is used to build a "generic" model for this camera view. The first and last frame from each sequence is shown in figure 4.4.

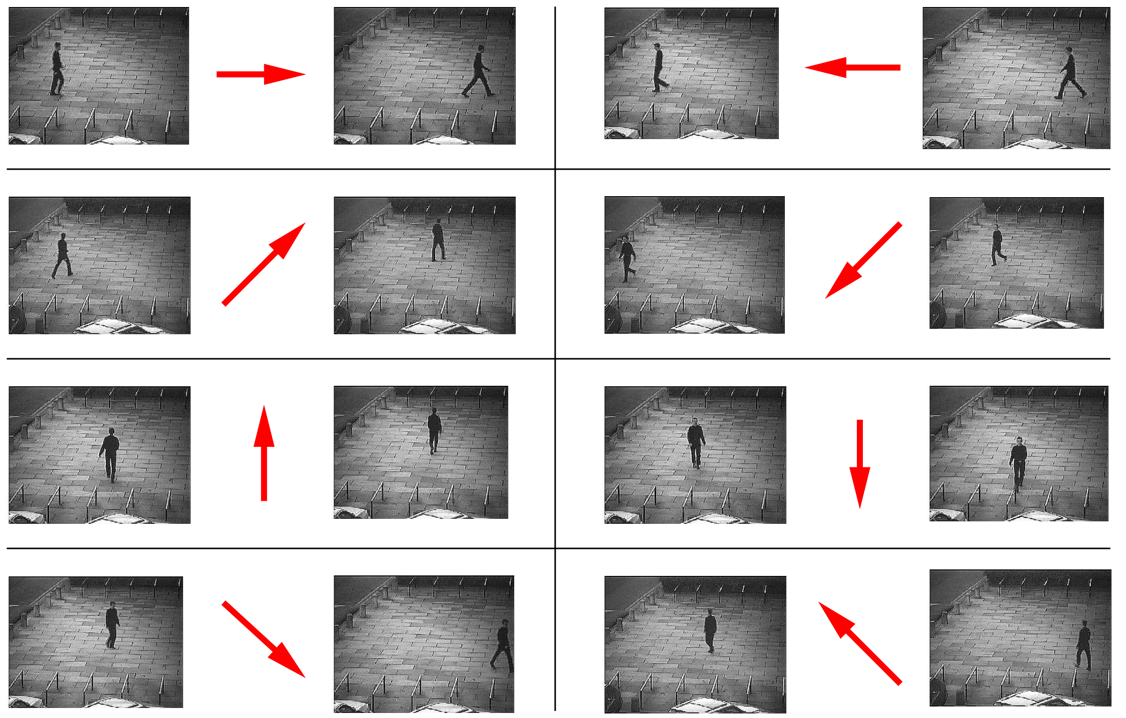


Figure 4.4: Training images for generic model

The system was tested on an additional "test" sequence showing the same person walking from left to right. The image subtraction segmentation for the test sequence appears to be fairly accurate and is used as the "ground truth" for subsequent analysis. The segmented (binary) image sequence was corrupted by adding randomly generated artifacts to the image. The first image in each sequence was left uncorrupted to ensure the initialisation phase was accurate (i.e. ensuring the initial position and size of the contour were close to the ground truth).

Three eigenshape models were used:-

1. A "rough model" generated from a noisy training set of shapes extracted from live video (unsupervised) from a fixed camera viewing a similar scene from a slightly different angle. This is the model from chapter 3. Each training shape was represented by a spline with $N = 40$ control points.

2. The "generic" model generated from the eight sequences described above. The model represents shapes of the silhouette of a person walking in a variety of directions. 40 control points were used.

3. A "specific" model generated from segmented shapes from the first training sequence only. In this sequence, the pedestrian walks from left to right across the image (i.e. in the same direction as in the test sequence). 32 control points were used.

Corrupting the images

Noisy binary images were generated by adding artefacts to the binary segmented test sequence. Randomly generated circles (with random position and radius) were drawn over the ground truth binary image in either the foreground or background colour. This type of noise was chosen to test the robustness of the system, for several reasons. Firstly the noise cannot be thresholded out (e.g. by ignoring observations where no "significant" contrast was measured). Secondly, the noise process will result in significant errors in contour measurements over whole sections of the curve. Hence these images are suitable for a rigorous test of the tracking system. Some corrupted images are shown in figure 4.5. It can be seen that the silhouette shape is disrupted and a conventional non model-based approach such as the "snake" of Kass *et al* would be unable to recover the object shape. Also note that the changes in shape are large so that the shapes can not be well represented by arbitrary small deformations of a mean shape or the shape in the previous frame.

Two types of noise were generated – using a temporally uncorrelated and a temporally correlated noise process. The initial temporally uncorrelated method adds the random artefacts to each image independently. The temporally correlated process adds *identical* artefacts to each image, thus generating partial occlusion of the whole scene.

The signal-to-noise-ratio (SNR) of the noisy images is calculated over the image sequence using

$$\mathrm{SNR_{in}(dB)} = 10log\frac{\mathrm{signal}}{\mathrm{noise}}$$

with

$$\begin{aligned}\mathrm{signal} &= \sum_{\mathrm{images}}\sum_{x,y}[I_{\mathrm{ref}}(x,y) - I_0]^2 \\ \mathrm{noise} &= \sum_{\mathrm{images}}\sum_{x,y}[I_{\mathrm{ref}}(x,y) - I'(x,y)]^2\end{aligned}$$

where $I_{\mathrm{ref}}(x,y)$ is the pixel value at (x,y) for the ground truth image and $I'(x,y)$ is the corresponding pixel in the corrupted image. The constant I_0 is set to halfway between the "background" and "foreground" pixel values, so that a patch of foreground and a patch of background both have the same signal strength, thus ensuring the SNR is independent of the relative image and object size.

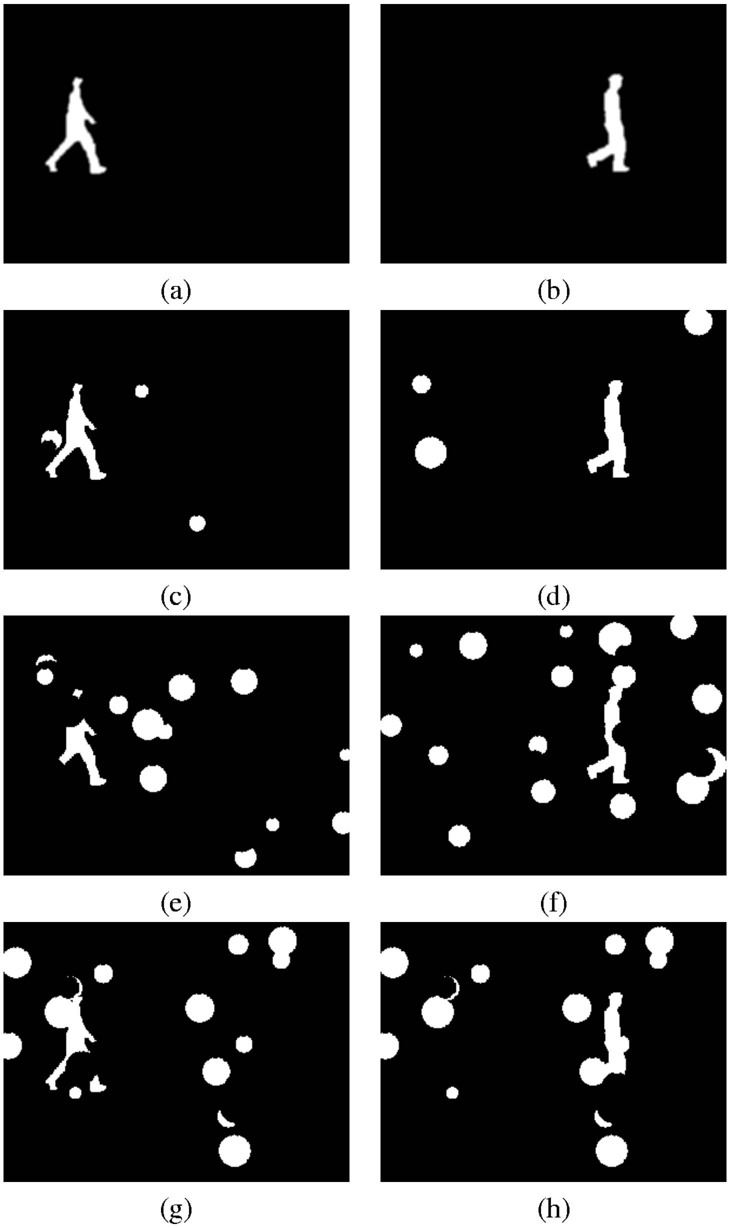
Measuring the accuracy of tracking

In order to measure the accuracy of the tracking process (i.e. the accuracy of shape, position and orientation of the tracked contour) an image based measure is used. Thus the error measure is independent of the parametrisation of the contour. The contour resulting from the tracking process is rendered flat filled in the "foreground" colour into the image I_{track}.

The tracking process is "local" so that the signal far from the object is never sampled. Hence, in this case, it is more appropriate to measure the signal in terms of the area of "foreground" pixels in the ground truth image. The signal and noise are calculated using

$$\begin{aligned}\mathrm{signal} &= 2\sum_{images}\sum_{x,y}[I_{\mathrm{ref}}(x,y)]^2 && (4.5) \\ \mathrm{noise} &= \sum_{images}\sum_{x,y}[I_{\mathrm{ref}}(x,y) - I_{\mathrm{track}}(x,y)]^2 && (4.6)\end{aligned}$$

where the pixel value for a "background" pixel is 0. The scale factor of 2 in equation 4.5 was chosen so that a SNR of 0 (i.e. $\mathrm{signal} = \mathrm{noise}$) would occur if the tracker silhouette consisted of a shape of the same area as the ground truth shape but inaccurately placed so that there is no overlap between the two. This is the "worst case" scenario where the tracker has completely failed



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 4.5: Some corrupted images:
(a), (b) original images
(c), (d) 12 dB (uncorrelated) noise added
(e), (f) 6 dB (uncorrelated) noise added
(g), (h) 6 dB correlated noise

to track the object. The output SNR (in dB), denoted $\mathrm{SNR}_{\mathrm{out}}$ is calculated in the usual manner, using the new values for the signal and noise.

Quantitative results

In order to ensure that the results were representative of the tracking performance, each experiment was repeated 20 times and the SNR (input and output) was calculated summing the signal and noise values over the whole set.

A plot showing the effect of temporally uncorrelated noise on the accuracy of the tracking system is shown in figure 4.6. For each eigenshape model, the number of modes of variation that encapsulated 95% of the appropriate training data were used. The main system parameters were fixed as follows

$$\begin{aligned} n_{\mathrm{sub}} &= 32 \\ L'N &= 320 \\ c &= 0.6 \\ \mu &= 0.2 \end{aligned}$$

An output SNR of 10 dB corresponds to 10% error in terms of the number of incorrect pixels over the number of pixels of interest. The performance of the tracker appears to be fairly robust even with significant input noise. The "specific" model incorporates more information appropriate to the input test sequence and hence produces a more accurate and more robust result whereas the "rough" model with a larger and more varied training set performs less well. As the input SNR increases the output SNR tends to values between 14.5 and 15.5 dB corresponding to errors of around 3%.

A further plot showing the effect of scene occlusion (temporally correlated noise) is given in figure 4.7. The effect of this type of noise is greater than that of temporally uncorrelated noise and an output SNR less than around 6 dB resulted when the system completely failed to track the pedestrian over part or all of the image sequence. For limited partial occlusion both the "specific" and "generic" models perform well.

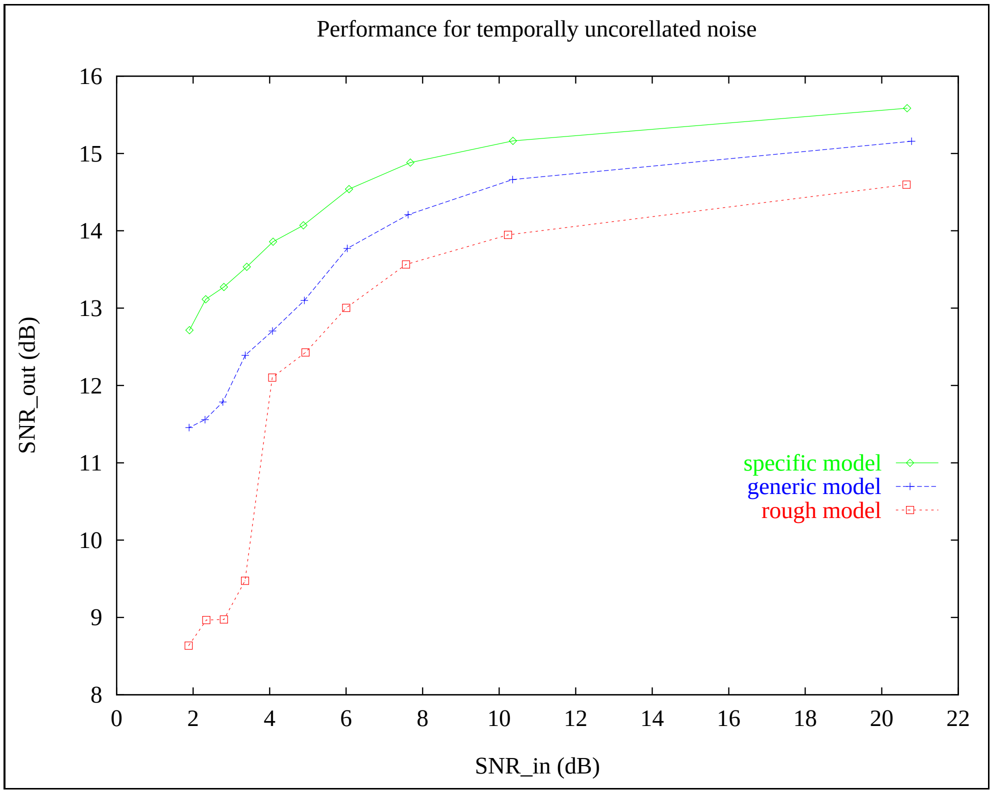


Figure 4.6: Effect of adding temporally uncorrelated noise on accuracy

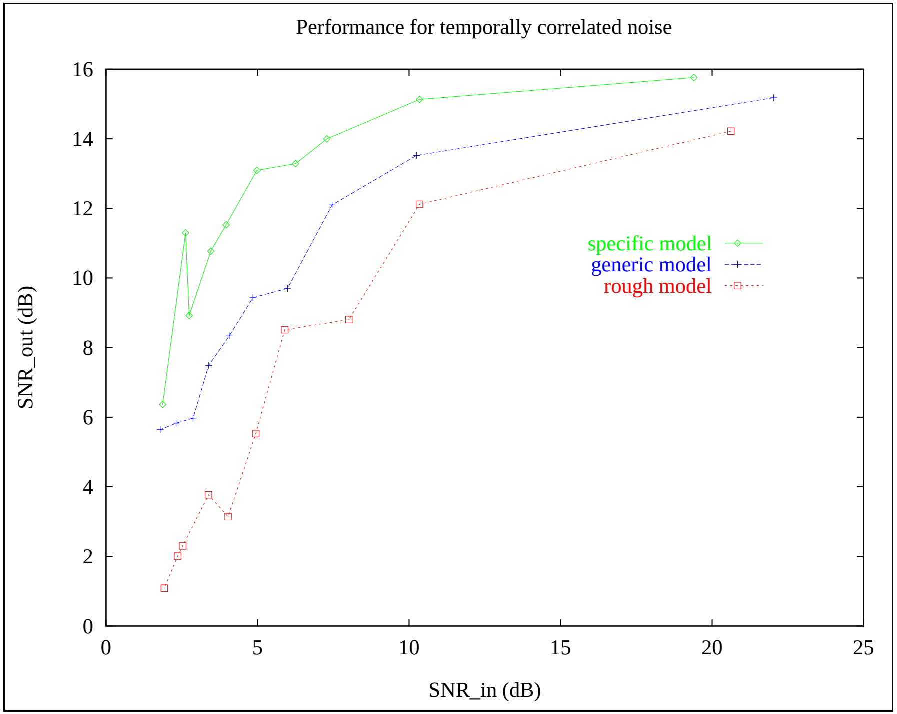


Figure 4.7: Effect of adding temporally fixed noise on accuracy

The effects of varying the total number of filter measurements (per frame) and the constant n_{sub} were investigated and the results shown in figures 4.8 and 4.9 respectively. Figure 4.8 shows the effect of varying these parameters on the accuracy of tracking. The input SNR for these experiments was fixed at 6.1 and the "generic" shape model was used. The surface plot shows that increasing these parameters generally results in an increase in accuracy. Figure 4.9 shows the effect of the number of measurements on the processing time taken. It can be seen that the time taken is linearly related to the number of measurements taken.

The number of shape parameters used, m, was also varied whilst keeping the remaining parameters fixed and using the "generic" shape model. The accuracy and speed of the results were measured as before and the results are shown in figures 4.10 and 4.11 respectively. The input SNR was fixed at approximately 4.6. The resulting accuracy tends to increase as the number of shape parameters is increased up to an optimal value of about 20. Subsequent modes contribute little to the shape representation and in fact can decrease the robustness of the system.

The processing time taken is linearly related to the number of shape parameters. This is

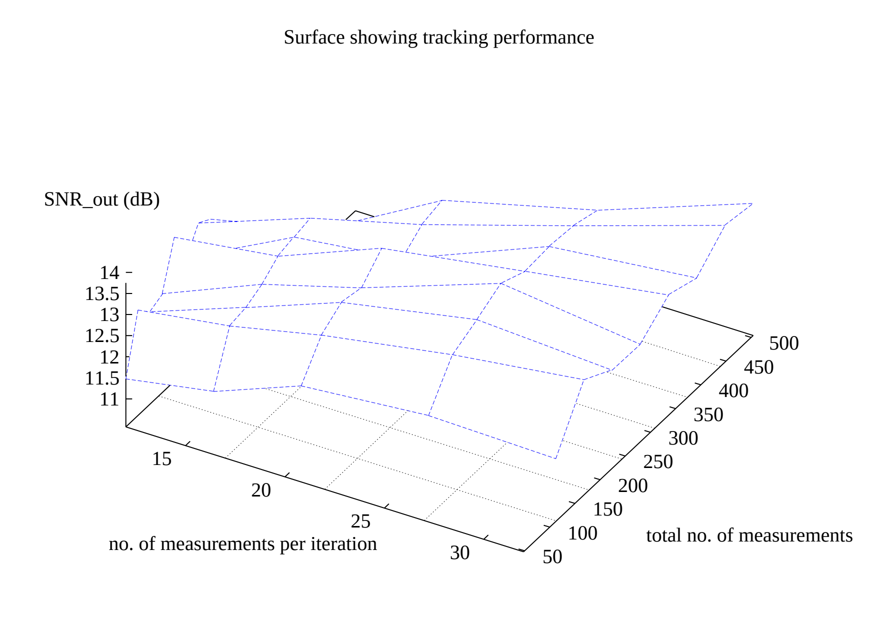


Figure 4.8: Effect of varying the measurement process on accuracy

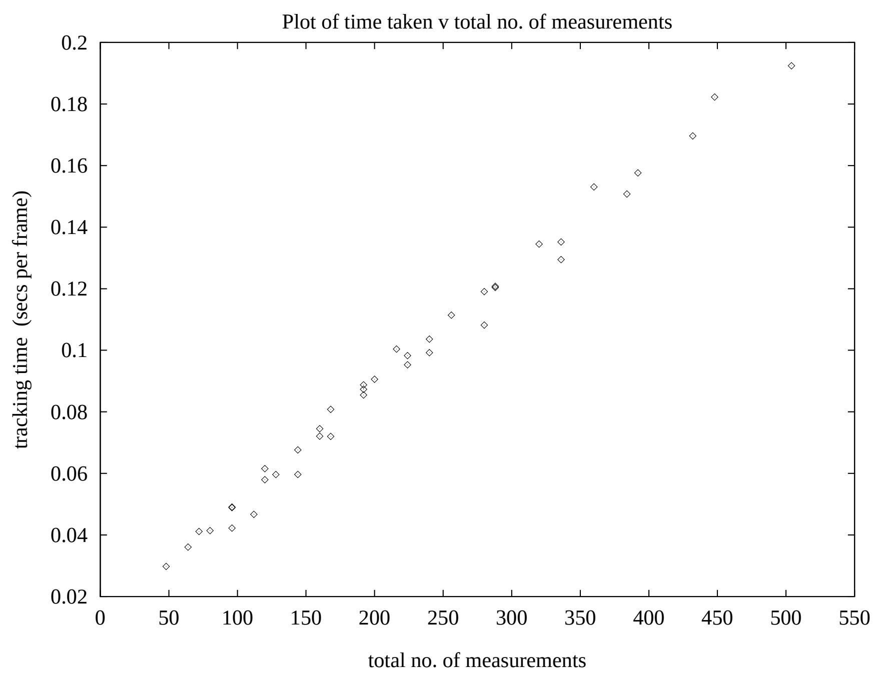


Figure 4.9: Effect of varying the total number of measurements on processing time taken

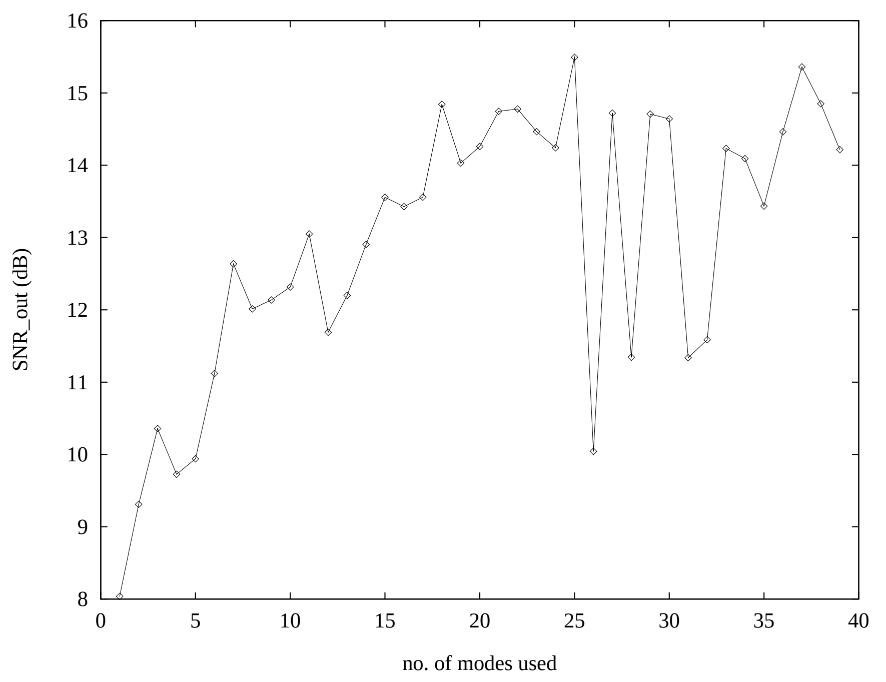


Figure 4.10: Graph of accuracy v no. shape parameters

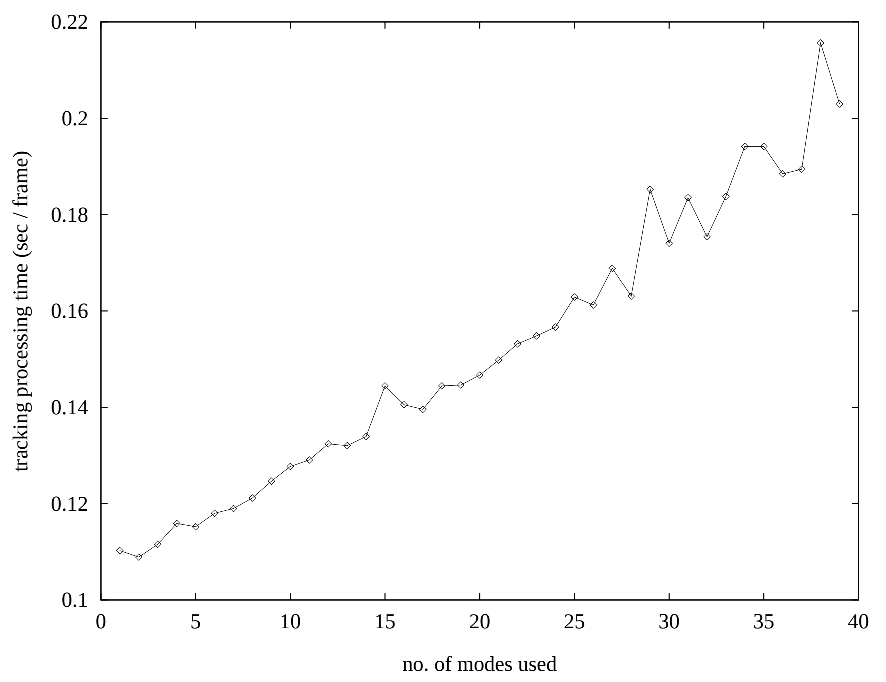


Figure 4.11: Graph of time taken v no. of shape parameters

due to the decoupling of the Kalman filter mechanism (the coupled filter is theoretically $O(m^2)$).

Further experiments investigated the choice of the "measurement parameter" c and the "noise model parameter" μ. For each pair of values 9 different noise-corrupted sequences of the same test sequence were processed and the results compared with the "ground truth" as before. Temporally uncorrelated noise and fixed temporally correlated noise was used with an input SNR of approximately 6 dB in both cases. The resulting surface plots are shown in figures 4.12 and 4.13 respectively. In order to aid the visualisation, the raw output signal to noise ratio is used (i.e. without using a logarithmic scale) and a smoothed surface approximating the data is drawn. It can be seen that optimal tracking performance is obtained for values of $c = 0.25$ and $\mu = 0.25$ approximately and that these values are not too critical. Small values of μ prevent the contour from deforming too quickly increasing robustness in the presence of noise. Too small a value however, freezes the contour preventing the tracking of a deforming shape. The optimal value for c is related to the choice of μ and in general will be larger for larger values of μ. In practice it is often desirable to use a more tolerant value of c of around 0.6 to allow the system to cope with shapes that are not well represented within the training set (e.g. when using a different camera view to that used in the training phase).

4.5.2 Qualitative results

Tracking with a fixed camera

Two test sequences of a walking pedestrian taken with a fixed camera were used. The generic model was used with 14 shape modes. The test sequences were not used in the generation of the shape model although the silhouettes are similar to those found in the model training set. The test sequence with the estimated contour superimposed is illustrated in figure 4.14. The frames are shown left to right top to bottom with every 4th frame displayed. The sequence was processed at 14.75 Hz (including the time taken accessing image files). The second processed sequence is shown in figure 4.15

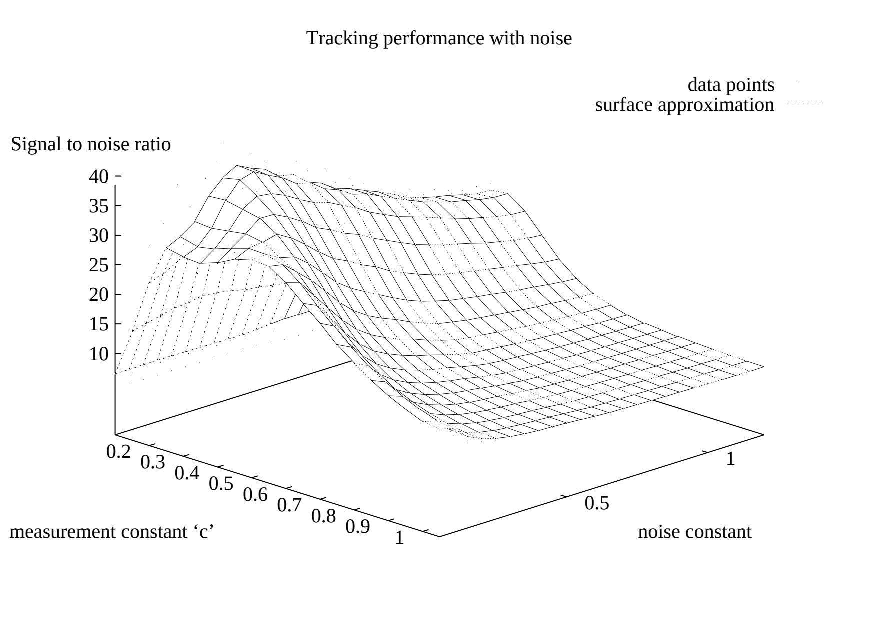


Figure 4.12: Plot showing the effect of measurement constant c and noise constant μ on accuracy in the presence temporally uncorrelated noise

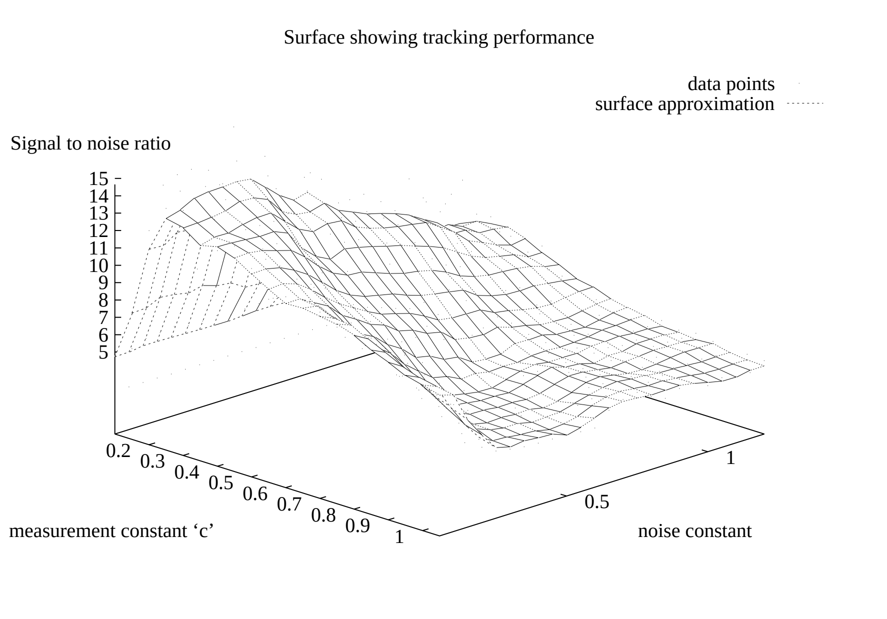


Figure 4.13: Plot showing effect of measurement constant c and noise constant μ on accuracy in the presence of temporally correlated noise

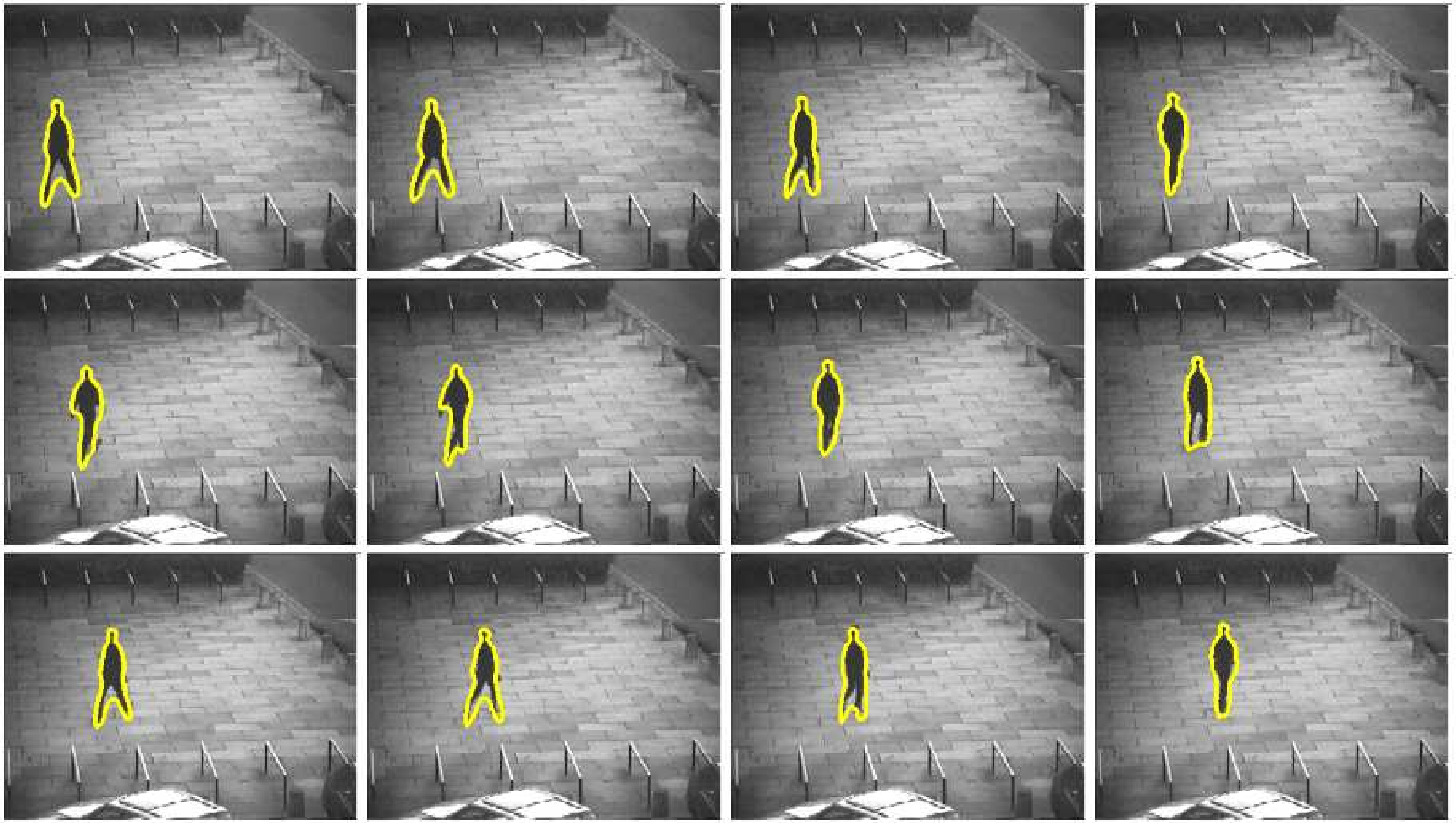


Figure 4.14: Results on 1st test sequence

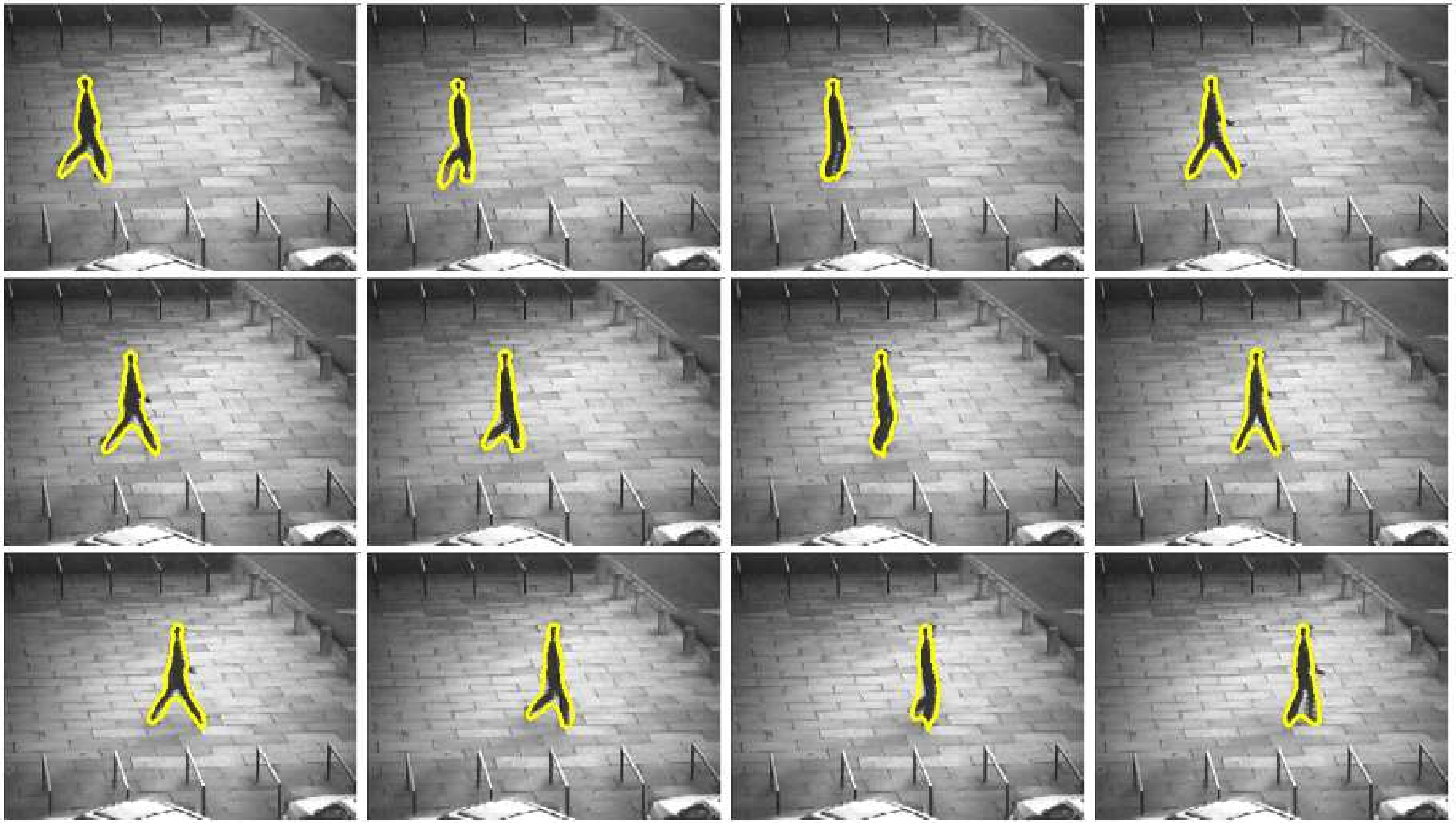


Figure 4.15: Results on 2nd test sequence

Search window

Figure 4.16 shows the initial search window obtained when tracking a walk sequence using the generic shape model with 14 shape parameters. As would be expected, the search window is largest near the walker's legs where the most significant shape deformation usually occurs. Figure 4.17 shows the search window for a single image frame over successive iterations, illustrating the multi-scale nature of the algorithm. For visualisation purposes n_{sub} was set fairly large for both these experiments.

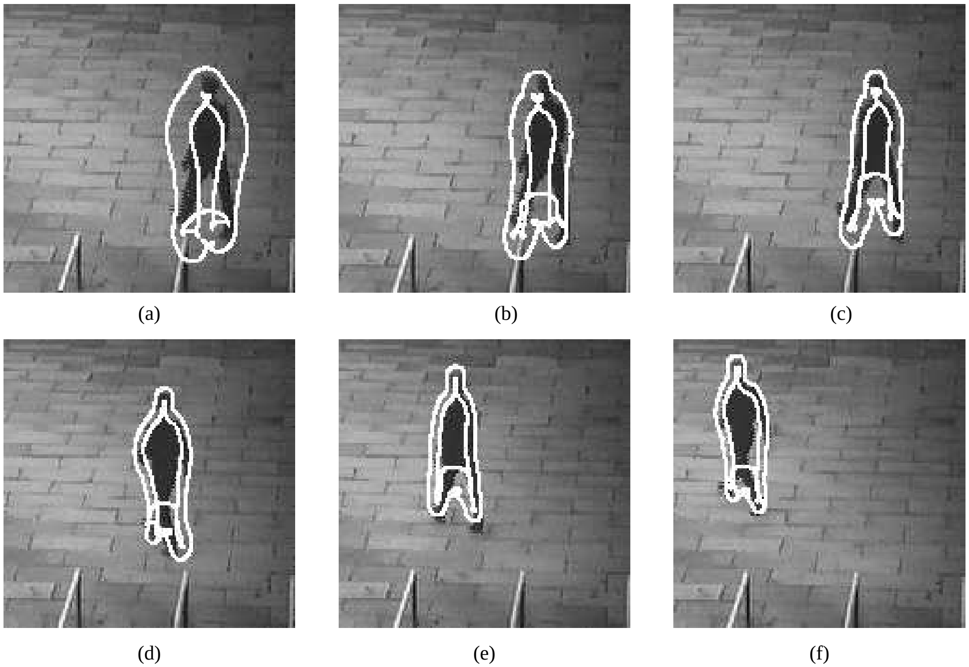


Figure 4.16: Search window: (a) to (f) frames 0, 1, 2, 16, 32 and 50

Tracking with modeled occlusion

When two or more tracked pedestrians overlap, the system copes by ignoring measurements where there is likely to be occlusion. A new test sequence, in which two pedestrians cross in front of each other, was used to demonstrate the occlusion reasoning. The results for this sequence are shown in figure 4.18. The estimated contour shape for each tracked pedestrian has been superimposed in separate colours. The "generic" model was used with 16 shape parameters. The three pedestrians are successfully tracked throughout the sequence with a qualitatively high degree of accuracy.

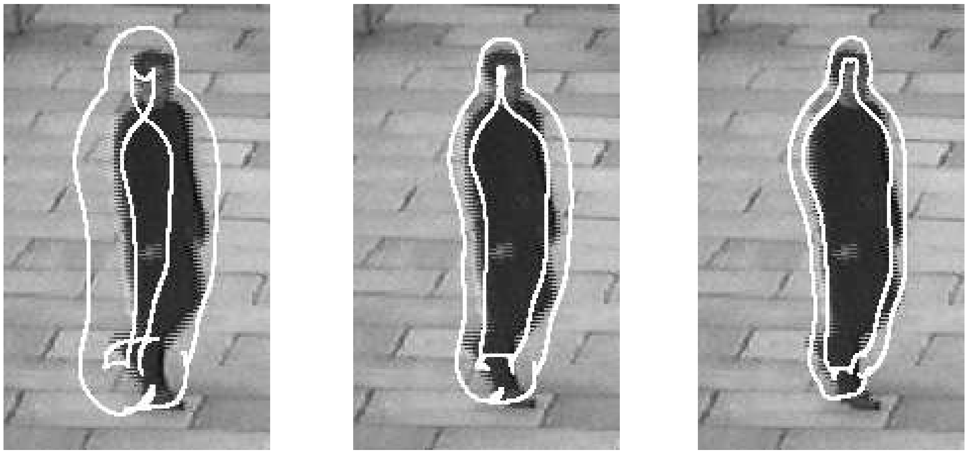


Figure 4.17: Search window for successive iterations on a single frame

Figure 4.19 shows a closeup of one of the image frames where partial occlusion occurs and the corresponding "occlusion" image for this frame. The white area in this image indicates pixels that are ignored in the measurement process. A further diagram (figure 4.20) shows the normals to the estimated contour where measurements were taken for one of the contours. It can be seen that the occlusion reasoning prevents potentially inaccurate measurements being taken.

Tracking with a moving camera

The system was tested on several sequences taken with a moving camera. In the first sequence the camera is initially fairly still, allowing image subtraction to be used for contour initialisation. The camera was hand held and a pedestrian was kept within the image by eye whilst zooming in on the walking pedestrian. This image sequence presents numerous difficulties as conventional subtraction based techniques can no longer be used. Furthermore, the camera is zooming and moving relative to the ground plane and there is also some camera shake.

The tracking system was applied to this difficult image sequence. The initialisation was done in the usual way (the camera was initially fairly still). Subsequent processing utilised an edge based contrast measure and the results are shown in figure 4.21. The images are ordered from left to right and from top to bottom with every 5'th image displayed. The estimated contour shape has been visualised over each input image frame.

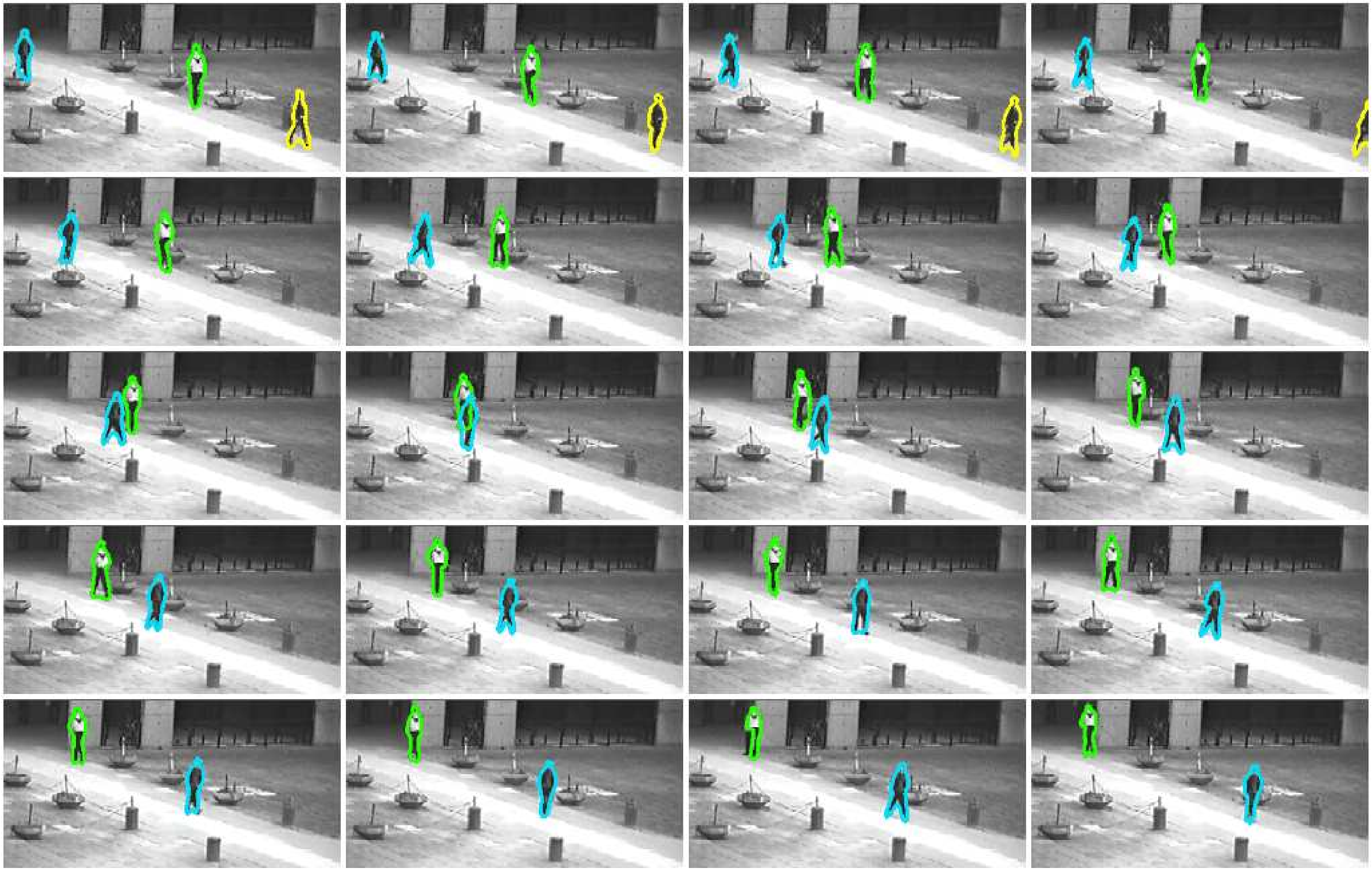


Figure 4.18: Results on test sequence with occlusion

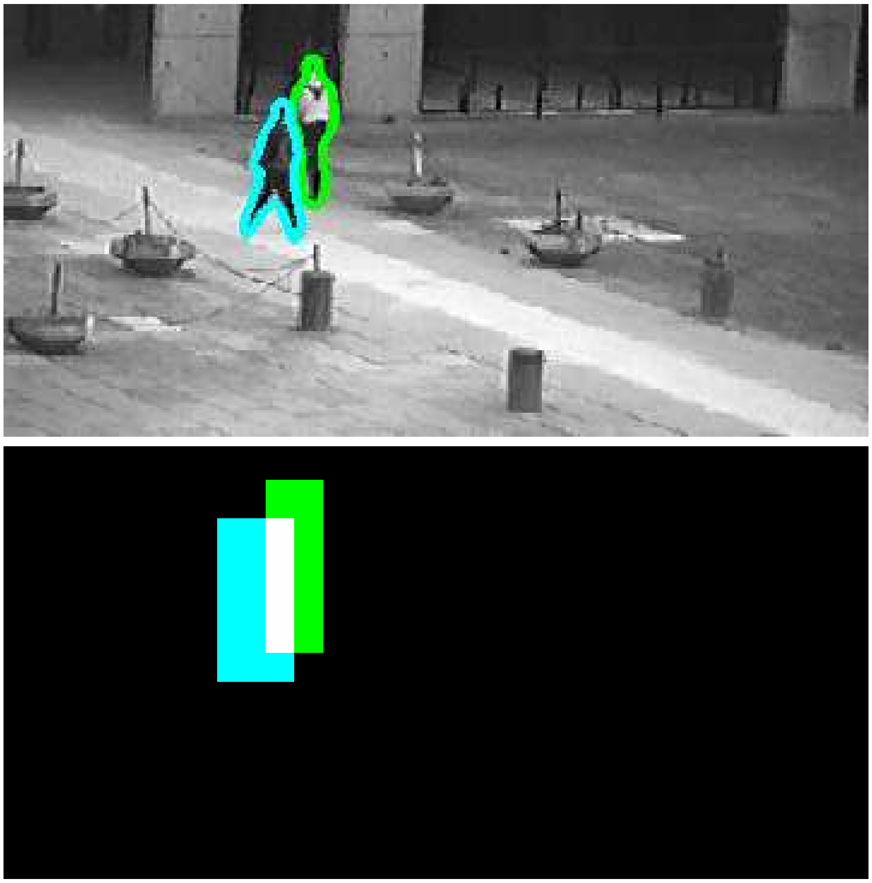


Figure 4.19: Occlusion Reasoning:
top: closeup of image with contours superimposed
bottom: corresponding occlusion image

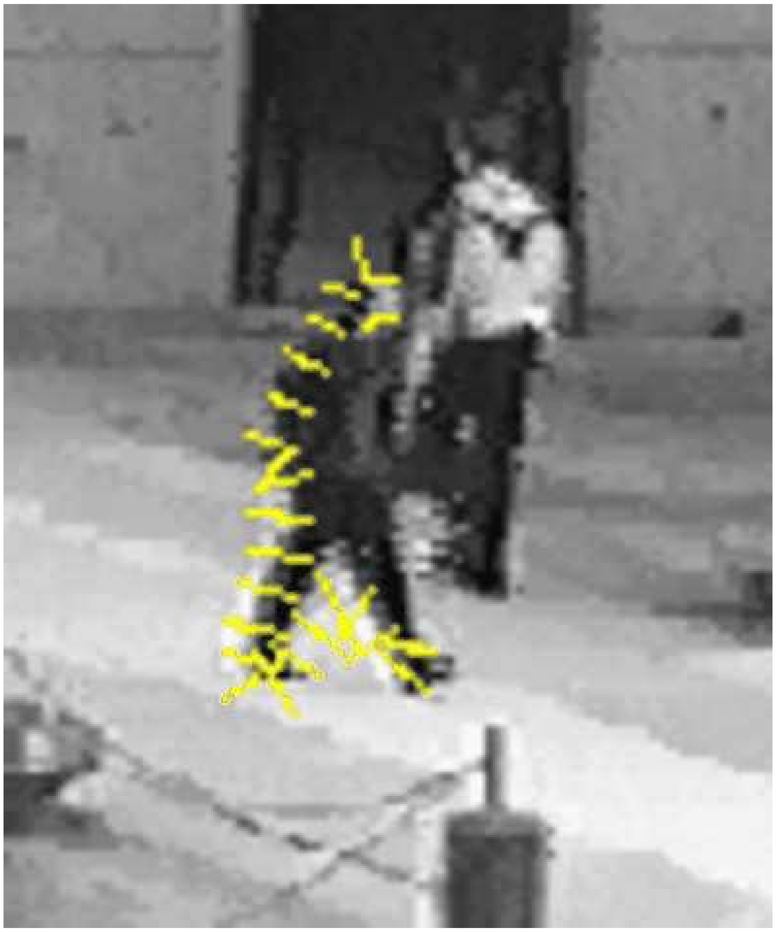


Figure 4.20: Measurements taken near occluded region

In order to improve the robustness of the system a highly constrained solution space was required and only 4 modes of variation were used (using the “generic” model) and the model space was constrained with a maximum Mahalanobis distance of 6. The results show the system copes reasonably well although the contour shape is only a “loose” fit to the underlying object shape.

A second similar (moving camera) image sequence was processed by the system (see figure 4.22). In this sequence two pedestrians are tracked as the hand-held camera pans and zooms. The two pedestrians are moving close together and in the same direction making the tracking task more difficult. The occlusion reasoning described previously, where an enlarged rectangle is used, helps prevent the two contours becoming tangled together.

Towards the end of this sequence an unmodeled third pedestrian walks into the image and occludes the tracked pedestrians. The system copes with this situation treating the occlusion simply as noise (there is no occlusion reasoning in this case as the 3rd object is not tracked). A closeup of one of these frames is shown in figure 4.23. As before 4 shape modes were used and an edge-based contrast measure drives the mechanism.

4.6 Discussion

The quantitative results show that the performance of the tracking system is affected by the system parameters and more importantly by the suitability of the linear shape model used. The errors measured in the quantitative analysis come from several sources.

- Smoothing error – due to the smoothing of the spline representation.
- Truncation error – caused by ignoring the less significant shape modes.
- Modeling error – due to an inaccurate *a priori* probability distribution (e.g. due to segmentation errors in the training shapes). Also due to inaccurate *a priori* assumptions in the stochastic model (e.g. unexpectedly large shape changes).
- Filtering error – due to ignoring of the off diagonal elements of the (shape) covariance matrix. These error are particularly noticeable if n_{sub} is too small and the shape parameter correlations are no longer insignificant.

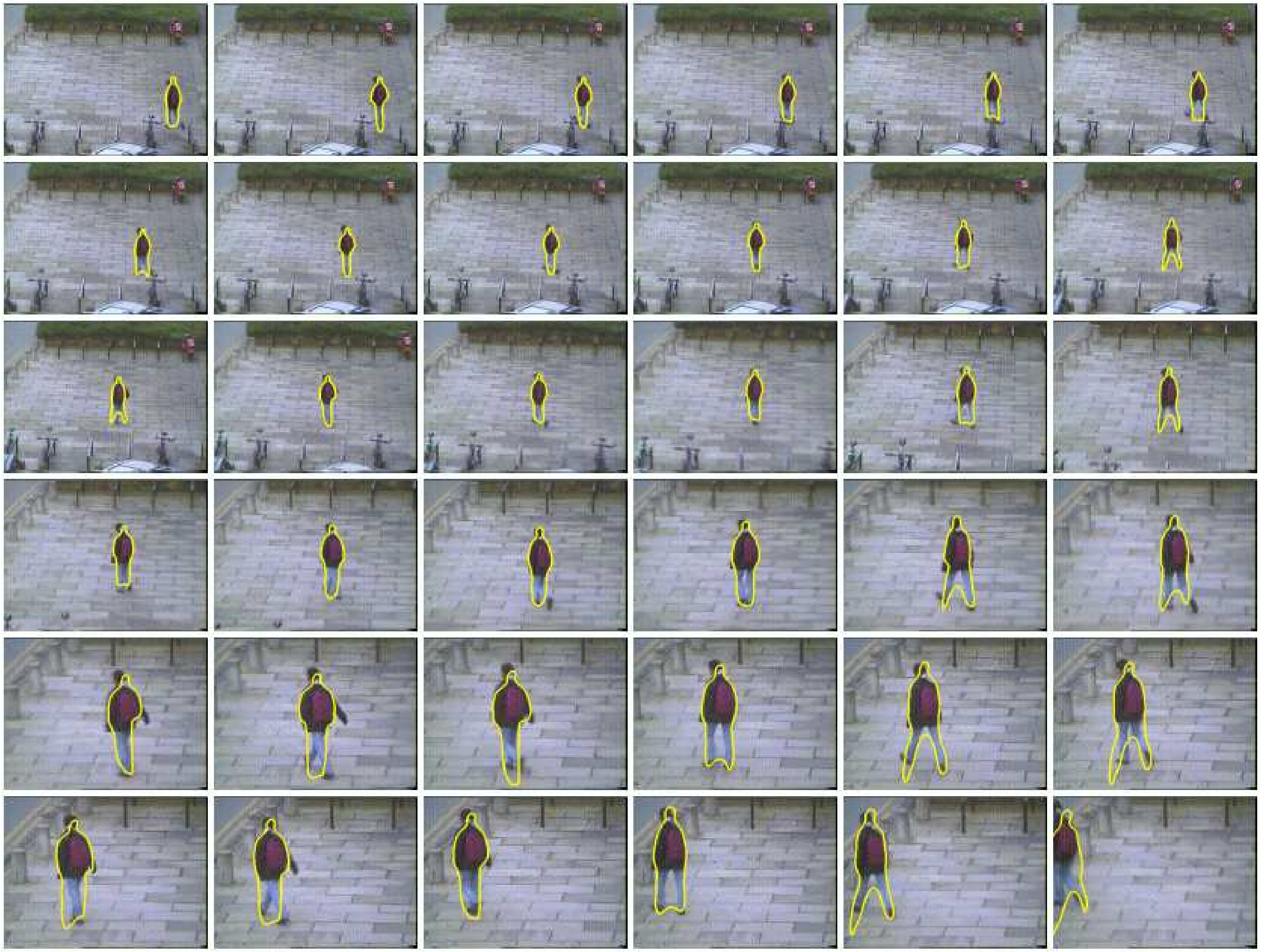


Figure 4.21: Tracking with a zooming camera

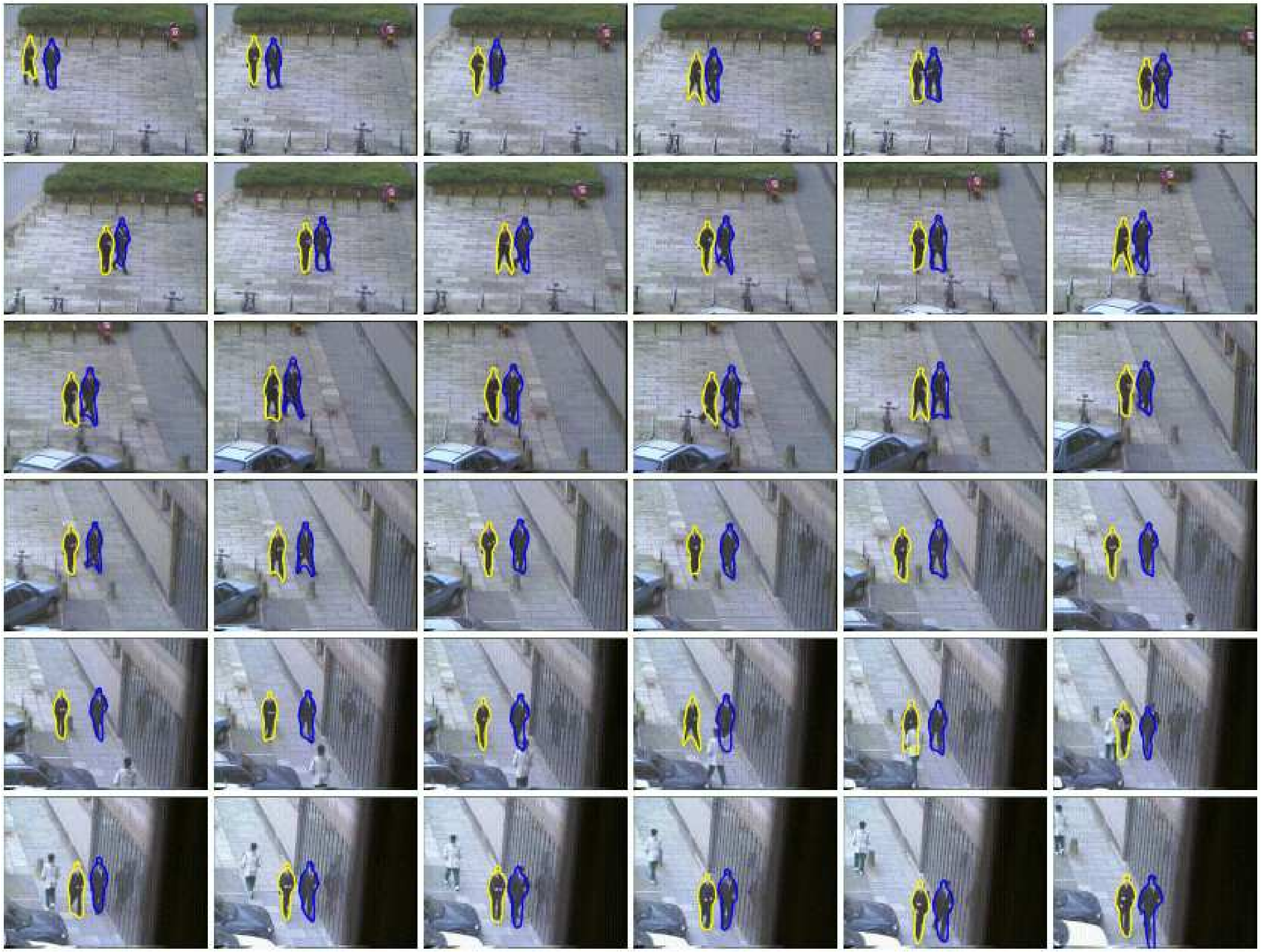


Figure 4.22: Tracking with a moving camera

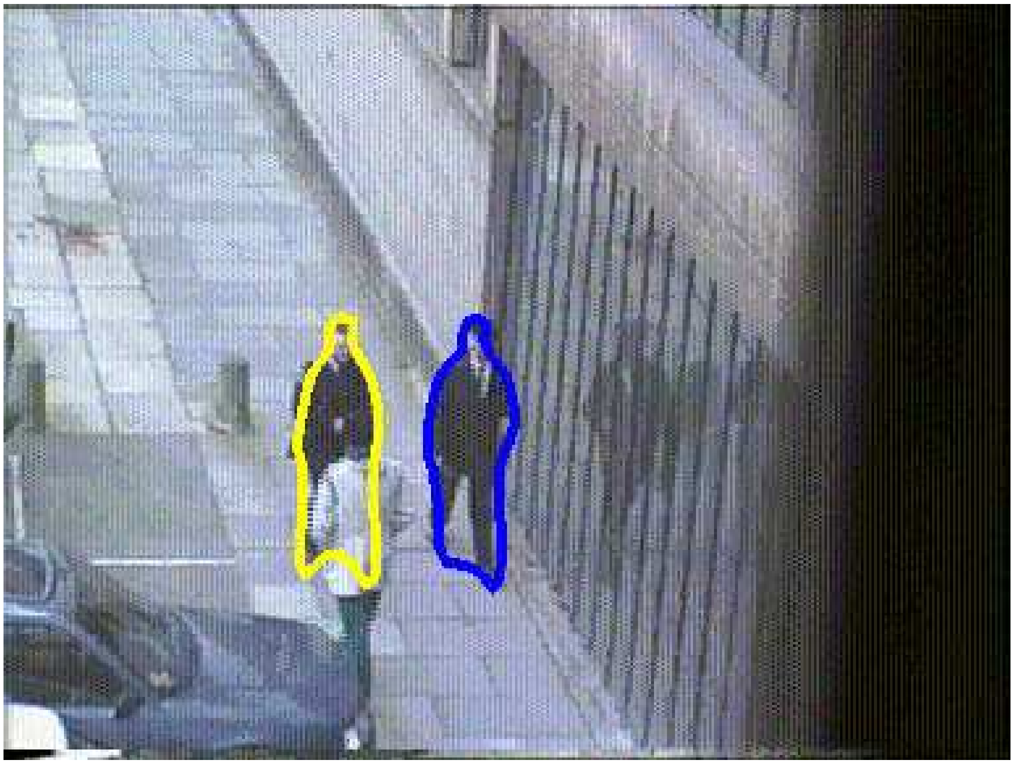


Figure 4.23: Closeup showing a third pedestrian occluding two tracked pedestrians

- Poor correspondence – even in the absence of image noise the measurement process is prone to errors as the contour can lock onto the wrong part of the image feature.

The modeling errors are typical of model-based vision. The model allows the system to perform robustly in the presence of noise but as a consequence will prevent 100% accurate tracking in the absence of noise when the object does not exactly conform to the modeling assumptions. Consequently, care must be taken to ensure that the desired output shapes are well represented in the training set.

By allowing some inaccuracies in the tracking system using only 10 shape modes from the "generic" model and a total of 160 measurements per image frame, real time performance of over 33 Hz was achieved with an output SNR of 13. In highly constrained situations (e.g. tracking people moving left to right across the image plane) the number of shape parameters and measurements can be further reduced allowing very high frame rates to be achieved. Note that the processing times in the above experiments include data accessing times (from movie files) and the experiments were run on a 100MHz R4000 Indy workstation.

The observed (and theoretical) complexity of the tracking system is $O(L'm)$, where m is the number of shape parameters and NL' is the total number of measurements per image frame. This contrasts with the conventional coupled Kalman filter which can be shown to be $O(m^2)$.

Chapter 5

Adaptive Improvements

5.1 Introduction

One of the problems associated with landmark free methods is that a large degree of variability in any shape descriptor may be due to the choice of parametrisation. In this chapter, an automated training method is described which utilises an iterative feedback scheme to overcome this problem. The aim is to build a compact contour model that describes the shapes in a training set. The more compact the model, the fewer shape parameters are required for accurate shape representation which leads to faster and more efficient image search and object tracking procedures. A more compact model also increases robustness by producing a more restricted solution space of feasible shapes.

In chapter 3, a consistent method for parametrising a shape is described and a linear eigenshape model similar to the PDM of Cootes *et al* is derived. In this chapter, the model is made more compact by eliminating some of the variability caused by control points shifting along the contour (which cause little change to the actual observed shape). This work has some similarity to the work of Williams [53] in that a covariance matrix associated with control point positions is learned from training data using an iterative learning process, although in his work the initial model is hand generated and a computationally expensive "generative" image fitting process is used. Furthermore, the methods are applied to hand-writing recognition of individual characters where a relatively simple B-spline model with only 4 control points can be used.

Hill and Taylor outline an approach to automating the placing of landmark points on training shapes [17]. In their work a two stage process is described – an initial scheme for generating a PDM and a refinement stage for making the model more compact. The initial scheme relies on an underlying method for finding correspondences between two shapes (e.g. using physically based vibration modes). Such a "corresponder" may fail in certain applications where the object shape can vary non-elastically (for instance the two shapes in figure 5.1). The refinement phase locally optimises each training shape's landmark points using a computationally expensive Simplex minimisation of an energy function. Each training shape is represented by t shape parameters representing the most significant eigenvectors of the current model where t is chosen heuristically (e.g. t can be chosen such that 90% of the variability of the training data is captured). The energy function encourages the landmarks to move closer to the mean shape (in terms of a Mahalanobis distance) if the improvement is not outweighed by an increase in representation error.

In this chapter, a simpler, alternative approach to automating the model building process is presented. An advantage of the method described here is that the complete contour shape is modeled as opposed to selected points on the boundary. Another benefit of this approach is that the implementation of the feedback learning scheme requires only two main modules to be implemented – an eigenshape analysis module (model building) and a tracking module (model fitting) – both of which are key elements of any comparable system. Thus, the system is "bootstrapped" allowing a more compact and subsequently more reliable model to be generated without engineering any new modules. Both methods reparametrise each shape in terms of the current shape model. The system described here utilises an additional "noise" process which allows the current shape model to change significantly without any loss in training shape representation. This additional step allows the system to "break free" from a poor initial model so that the initial model need not be close to the final optimal solution.

Results are shown illustrating the qualitative and quantitative benefits of utilising the new adapted eigenshape models over the models generated in previous chapters.

A related "bootstrapped" approach is described by Syn and Prager [54]. They described a semi-automated system for building a PCA model for 3D medical data sets. The model is updated incrementally using a FEM modal analysis to provide correspondences between recovered 3D

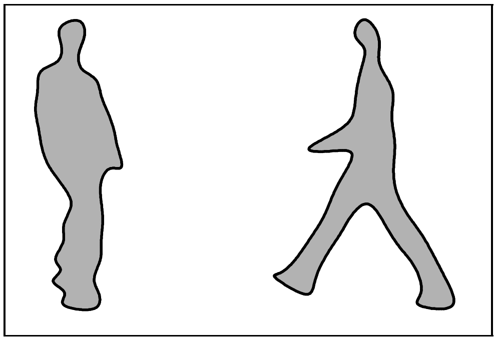


Figure 5.1: Two shapes where a matcher may fail

mesh descriptions and landmark feature points. The statistical component model is then used to improve the mesh shape recovery process.

5.2 Generating the initial model

In chapter 3, a method for parametrising an arbitrary shape (i.e. a closed boundary) is described based on one fixed point and the length round the contour. The fixed point used was the upper most point at which the principal axis crossed the object boundary. An N-point uniform cubic B-spline was used to represent each shape conveniently allowing every point on the boundary to be modeled without using an arbitrary dense set of boundary landmark points.

This method produced reasonably good results and the eigenshape analysis of the training shapes resulted in a significant reduction in dimensionality, suggesting that the initial parametrisation was reasonably consistent. However, the model still required a relatively large number of shape parameters for accurate shape representation (see figure 4.10). This was partly due to the problem of control points "shifting" along the object boundary producing little change in observed shape. Thus, similar training shapes may have slightly different nodal representations (due to variation in the material parameter values of corresponding boundary points) increasing the total variance of the training set.

Furthermore, principal component analysis attempts to linearise shape changes from the mean shape, which may in reality be non-linear. By reparametrising the shapes it may be possible to ensure that the shape changes are closer to a linear model. A similar effect is apparent in the "Cartesian-Polar Hybrid" PDM described by Heap and Hogg [55] where the choice of shape representation can significantly improve the resulting model.

Another way of looking at the problem is to consider the initial training set to lie within a lower dimensional, constrained shape space, within the original shape-vector space. This space is defined by the constraint that control points are equally spaced around the contour. The mean shape does not necessarily lie within this constrained space, resulting in a reduction in the compactness and consistency of the model. By relaxing the constraints on the training shapes the reparametrised training set can result in a more compact model where the mean shape is more representative of the "average" shape.

The initial eigenshape model is regarded as the first step in an iterative process. Consequently, the exact method of shape parametrisation (e.g. the choice of fixed point) will not be critical. For instance, if the shapes are already reasonably well registered, the fixed point may be the upper most boundary point.

5.3 Adaptively improving the model

In order to adapt the shape model an iterative learning process will be utilised. However, in order to proceed further an accurate contour fitting scheme will be required.

5.3.1 Accurate image search using the shape model

In chapter 4, an active search method for fitting a linear shape model to an image (from a sequence of images) containing an example of the object of interest is described. The method relies on certain *a priori* assumptions being made about the object shape. Specifically, the estimated shape is initialised to the mean shape with the variance of each shape parameter estimate set to the associated eigenvalue. In subsequent image frames the shape parameters are allowed to vary slowly by using a noise term for each shape parameter set proportional to the eigenvalue for that shape parameter. The Kalman filter mechanism can be regarded as a physical system where there are internal forces pulling the shape parameters towards the current shape estimate and external forces pulling the shape towards image features. The filter is suitable for robust and fast tracking but may lead to compromise solutions when the internal forces balance the image forces.

A method is required for accurately fitting the shape model to a (possibly pre-segmented) shape in an image. By treating a single image as a sequence of identical images, the tracking system can be adapted to give very accurate fitting at the expense of computational load. The resulting method is similar to Lowe refinement where the prior model at each iteration is set to the result from the previous iteration (see section 2.2.3). Each iteration of Lowe refinement is comparable to running the tracking system on one of the identical image frames. The method is computationally expensive but allows optimal accuracy to be obtained, given that the shape is to be approximated by an N point cubic B-spline. A diagram illustrating the modified tracking system (for accurate shape fitting) is shown in figure 5.2. Note that "global shape constraint" is relaxed to ensure a good fit is obtained. Figure 5.3 shows an example of accurate shape fitting comparing the initial fit obtained on the first frame and the more accurate fit on the final (identical) frame. There is an obvious improvement although the difference is not large. Note that when lock is lost over part of the contour the local search scale will lengthen allowing the contour to recapture a lost feature. This method ensures that the final contour is locked onto a suitable feature over the whole curve and hence a very accurate fit is obtained.

The Kalman filter mechanism allows *all* the shape modes to be used (as opposed to the usual subset of "significant modes") without the system failing, although in the presence of significant image noise the use of additional modes can increase errors (see figure 4.10).

5.3.2 Improving the model: Theoretical basis

The motivation for the method described in this chapter is based on the following assumptions:-

- The optimal parametrisation for each training shape is the parametrisation obtained by accurately fitting the optimal model to each shape.

- The optimal model is the model obtained from the analysis of the optimally parametrised training shapes.

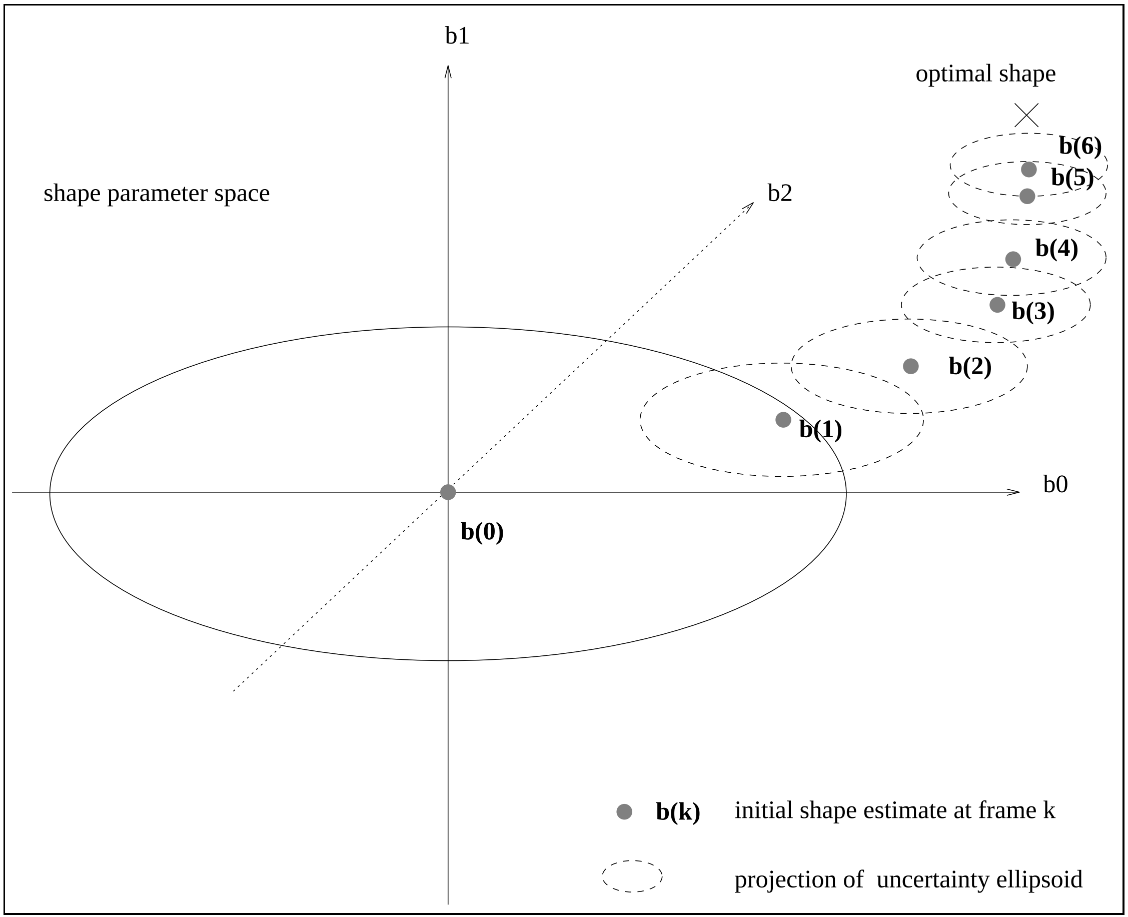

b1
optimal shape
b(6)
b(5)
shape parameter space
b2
b(4)
b(3)
b(2)
b(1)
b0
b(0)
b(k) initial shape estimate at frame k
projection of uncertainty ellipsoid

Figure 5.2: Accurate image fitting

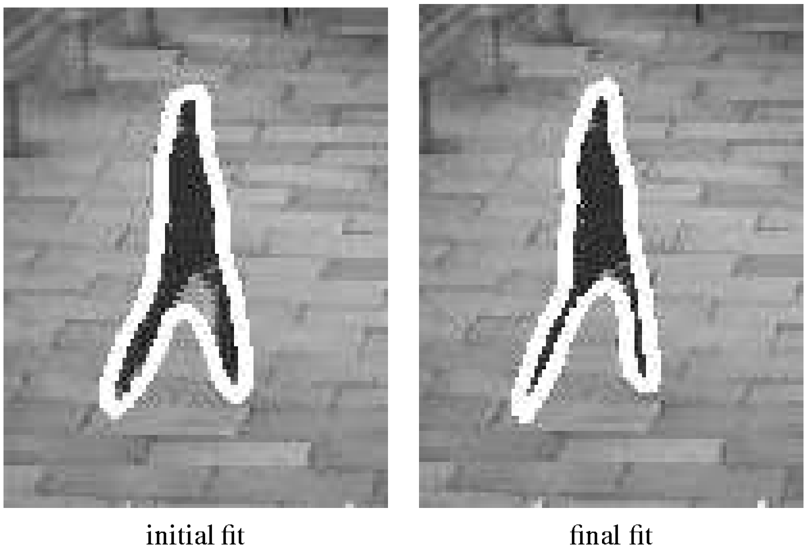


Figure 5.3: An example of accurate fitting

These assumptions appear to be reasonable. Consider the opposite case, where the parametrisation obtained from fitting with the optimal model is different from that used in the generation of the model. Such a model is based on an "inconsistent" parametrisation of the training shapes and suggests that it is not the best representation of the available training information. If the assumptions are satisfied then the results of the image search can provide a useful nodal description of a new image contour that can be directly compared with the nodal description of each training shape used in model generation.

5.3.3 Initial approach

An obvious approach to "bootstrapping" the eigenshape model is to utilise the accurate image search mechanism on the training images. The resulting shape parameters b_i, can be mapped into the corresponding shape vectors and this new training set used to calculate a new mean shape and covariance matrix and hence a new eigenshape model. The process is repeated until convergence (which may not be guaranteed) at an optimal solution. The scheme may be regarded as a closed loop energy minimisation scheme (see, for example, Haykin [56]) similar to a neural net and other learning scheme.

The method requires high quality (possibly pre-segmented) training images. For each image, the approximate object size, orientation and location within the image are known. A new set of training shape vectors can be obtained by running the active search method on these images. The new training shape vectors are aligned and a new covariance matrix generated. Note the parametrisation of the shapes is no longer explicitly calculated but implicitly derived from the current eigenshape model.

Each training shape can be reparametrised without affecting the apparent shape by allowing the control points to shift along the contour boundary. The feedback scheme tends to pull the control points towards the more significant modes of variation (which vary more easily) whilst maintaining the contour shape. The result is a more compact model. A diagram illustrating this effect is shown in figure 5.4.

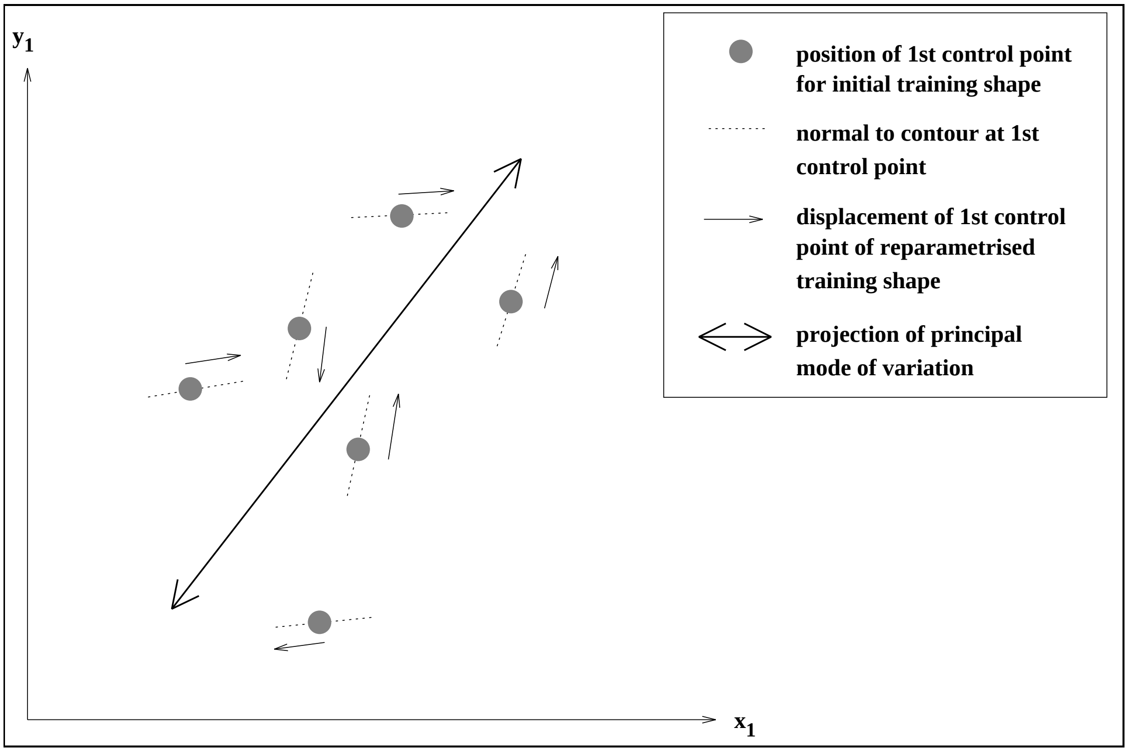

y_1
x_1
position of 1st control point for initial training shape
normal to contour at 1st control point
displacement of 1st control point of reparametrised training shape
projection of principal mode of variation

Figure 5.4: Diagram illustrating effect of reparametrisation

5.3.4 Improved iterative method

Even if the full set of $2N$ eigenmodes are utilised in the active search method, shape variations which do not occur within the initial training set will never become apparent in subsequent models. For example, supposing there are only two training shapes, the search space will effectively be a 1D shape space, since there is only one non-zero eigenvalue and the estimated shape parameters for the remaining $2N - 1$ modes will therefore be fixed at zero.

As the initial model is only an estimate of the optimal model an additional step is taken. The current eigenshape model is perturbed by a simulated noise process. The eigenvalues λ_i are updated as follows

$$\lambda_i{}' = \lambda_i + \sigma^2$$

This is equivalent to adding Gaussian isotropic noise with variance σ^2 to the boundary points of the training shapes. i.e. generating a new covariance matrix S' given by

$$S' = S + \sigma^2 \mathcal{H}^{-1}$$

This step allows (arbitrary) small perturbations of the nodal positions. This hybrid model allows fine detail that is not well represented by the original model to be more accurately recovered. The method is similar to that employed by Cootes and Taylor to combine the PDM with a finite element, physical model [2]. It is important to note that *all* the eigenmodes are used since the noise process ensures that no mode of variation can be regarded as insignificant. The Kalman filter active search mechanism allows the more significant modes to vary more easily so that all of the $2N$ modes can be employed without the method becoming unstable.

The parameter σ is initially set to around 4 pixels and subsequently decreased gradually (decaying exponentially at a heuristically chosen rate). A diagram illustrating the scheme is shown in figure 5.5. Using too large a value for σ would reduce the effectiveness of the current shape model which is required for accurate image fitting (and hence for shape parametrisation).

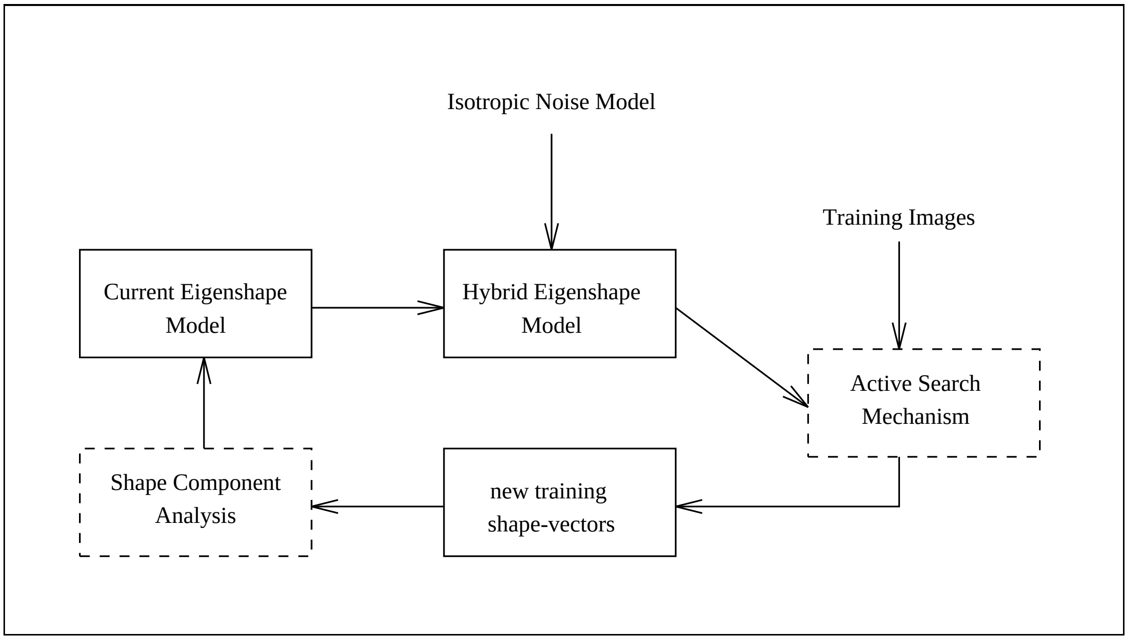


Figure 5.5: Iterative feedback scheme

5.3.5 Implementation

In this implementation the initial training set was generated using background subtraction and thresholding. The shapes were parametrised using the fixed-point method described in chapter

3 and a mean shape and covariance matrix calculated in the usual way. Subsequent image fitting was performed using the unprocessed training images and contrast was measured using the "fixed camera method" of chapter 4. The reason for using the unsegmented training images is that inaccuracies in the initial segmentation, due to the choice of threshold, are reduced. Each image was treated as a new image sequence of twenty identical frames.

5.4 Results

5.4.1 Single walk data set – the "specific model"

The data set contained 59 shapes (silhouettes) segmented from an image sequence of a pedestrian walking from left to right across the image. (This is the training set used for the "specific" model in chapter 4.) Background subtraction was used to segment the silhouette of the walker. Four of these training shapes are shown in figure 5.6.

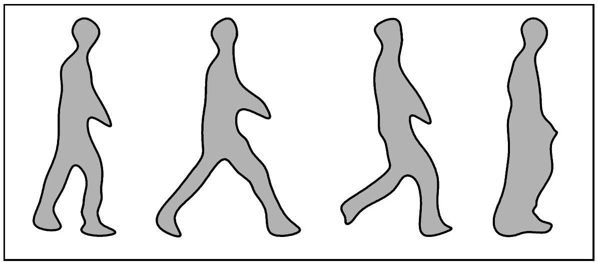


Figure 5.6: Training shapes from the single walk set

The feedback scheme described previously was implemented with and without the additional noise process. Each iterative step generated a new eigenshape model which was then used for subsequent active image search. The initial model is the "specific model" from chapter 4. A "compactness" measure was calculated for each model as follows:

$$\text{compactness} = \frac{\lambda_1 + \lambda_2}{\sum_{i=1}^{2n} \lambda_i} \times 100\%$$

where λ_1, λ_2 are the two largest eigenvalues in the model. The compactness measures the percentage the principal two "modes of variation" contribute to the total variance. A large compactness measure indicates that most of the variance is encapsulated by these two modes. The compactness of each model is shown in figure 5.7. The graph indicates that in both cases the compactness increases from under 65% for the initial model to almost 90% for the final, adapted model. The additive noise process has little effect on this increase in compactness.

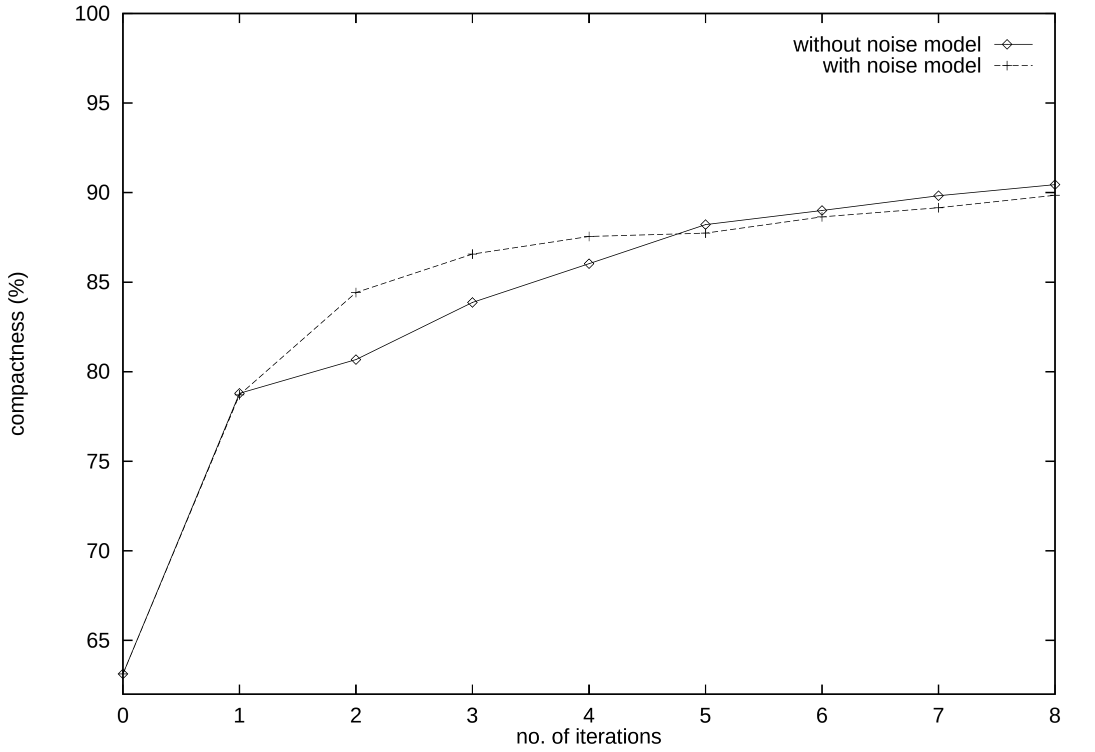


Figure 5.7: Compactness of single walk models

A "fitness" measure was also calculated at each iteration. This was a crude measure of how close the final contour lies to the true object shape after each image search. The average image contrast at sampled points on the contour was used and this "fitness" was averaged over the training set. A high average fit indicates that most of the contour points lie close to an edge and hence the segmentation should be accurate. The results for both methods are shown in figure 5.8. The plot shows that without the noise process the benefits of increasing compactness are offset by the decrease in average fit. However, the inclusion of the noise process generally results in a better fit reaching a stable maximum.

Note that these plots show that the iterative process converges quickly, due to the fact that the initial model is fairly good, with the significant improvements occurring within the first few

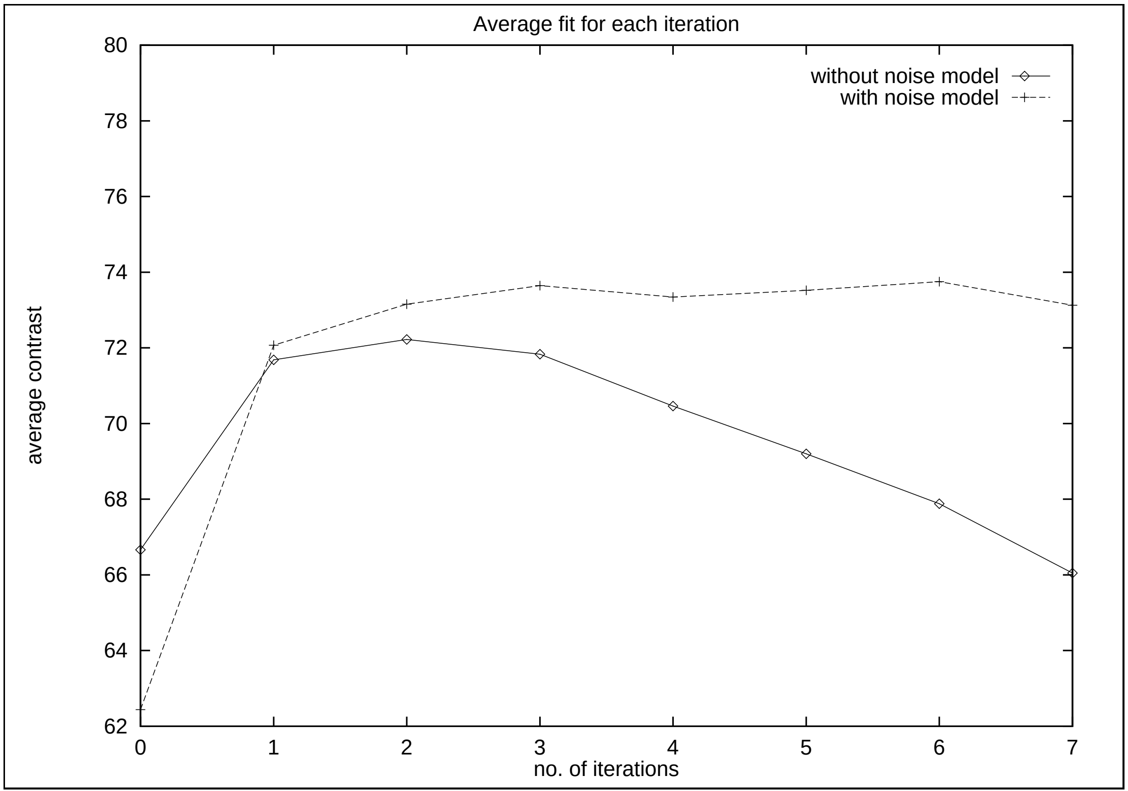


Figure 5.8: Average ‘fitness’ of single walk models

iterations.

Figure 5.9(a) shows a graphical representation of the effect of varying the principal shape parameter in the initial model. Figure 5.9(b) shows the principal mode of variation for the final adapted model. It is clear that there is more information encapsulated in the principal mode of the adapted model.

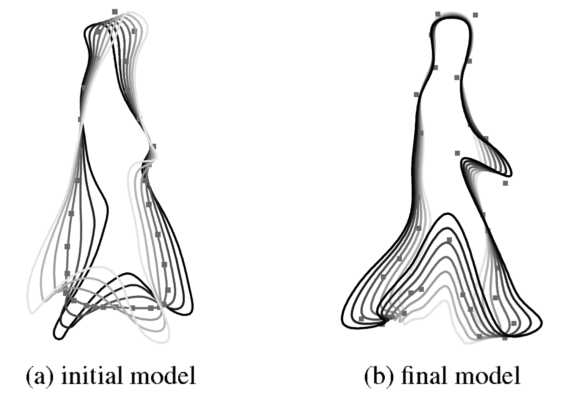


Figure 5.9: Principal modes of variation

5.4.2 Large data set – "generic model"

A second data set was generated containing 462 shapes of the silhouette of a pedestrian walking in a variety of directions. The training images from the "generic model" in chapter 4 were used. (A sample of the training shapes is shown in figure 5.10.) In this experiment the results of the two methods were very similar. This is probably due to the fact that the initial data set is very large and already quite noisy. Hence, there is no need to add simulated noise. Results are shown for the simpler scheme outlined in section 5.3.3.

Fitness and compactness measures were calculated as before and the results are shown in figures 5.11 and 5.12 respectively. Figure 5.13 shows the first 10 eigenvalues for each successive model. The principal variation modes of the initial and adapted eigenshape models are visualised in figures 5.14(a) and 5.14(b) respectively.

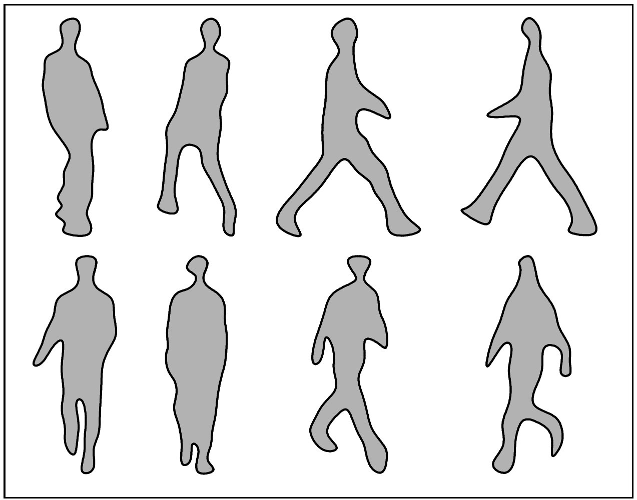


Figure 5.10: Training shapes from large training set

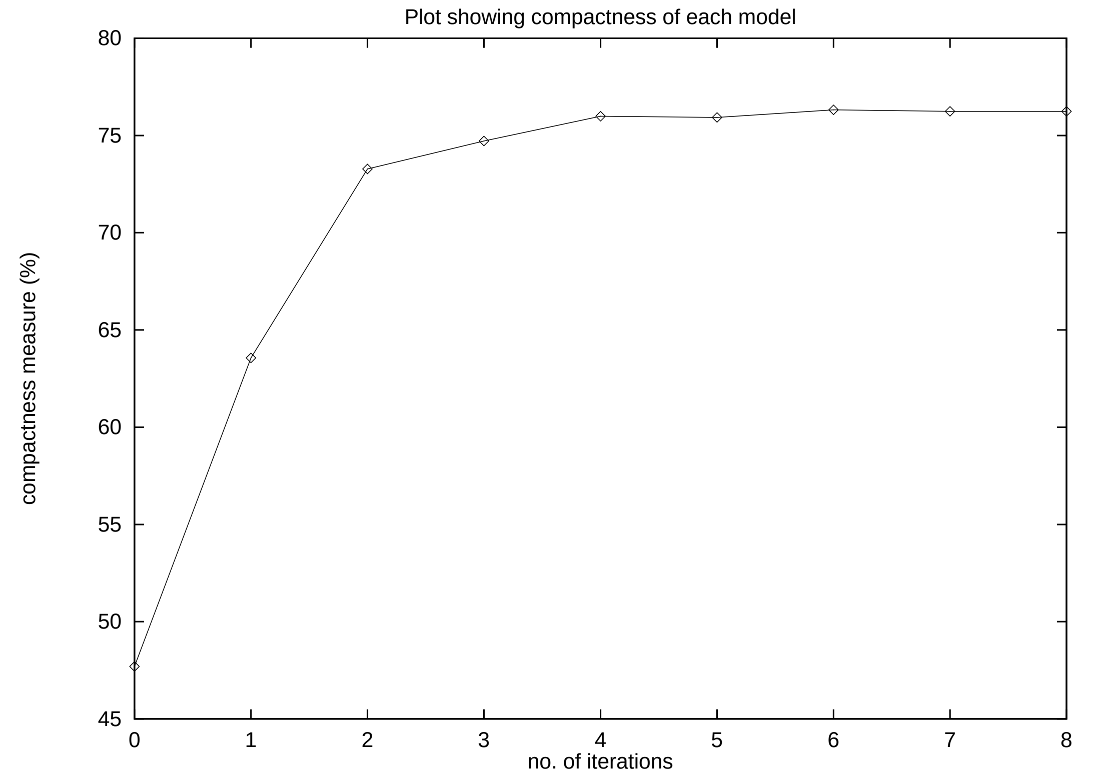


Figure 5.11: Compactness of generic models

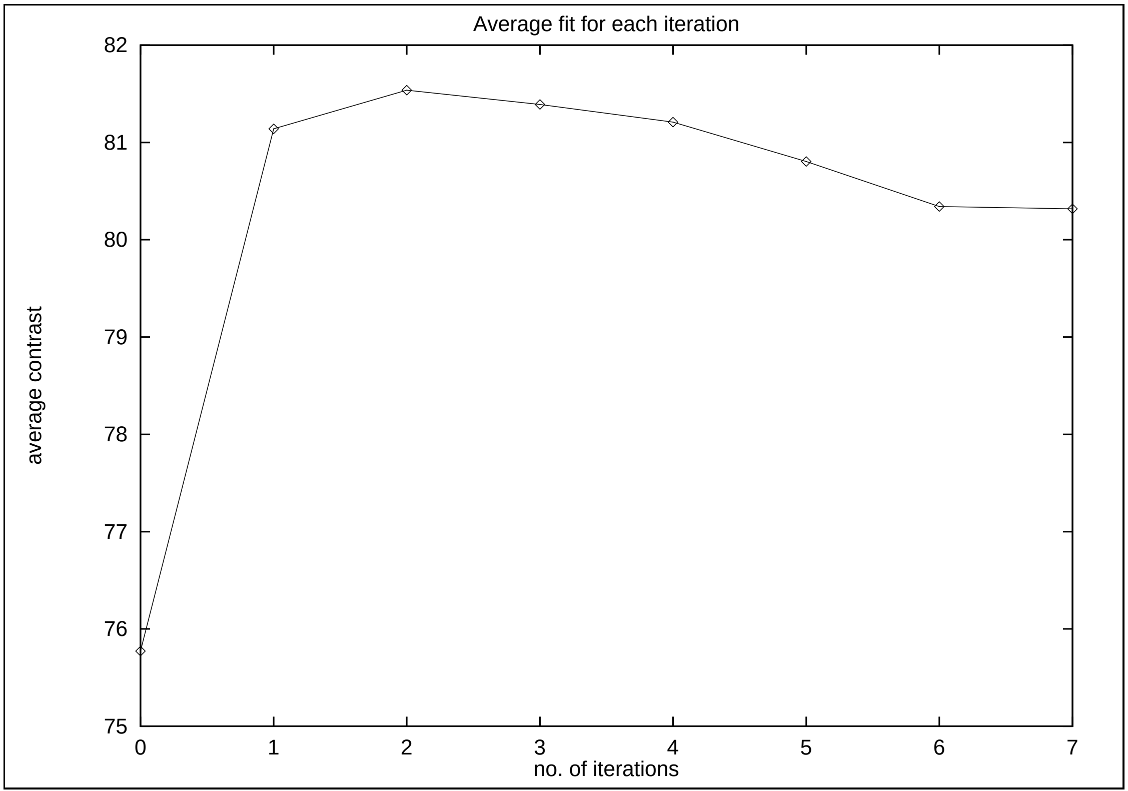


Figure 5.12: Fitness for generic models

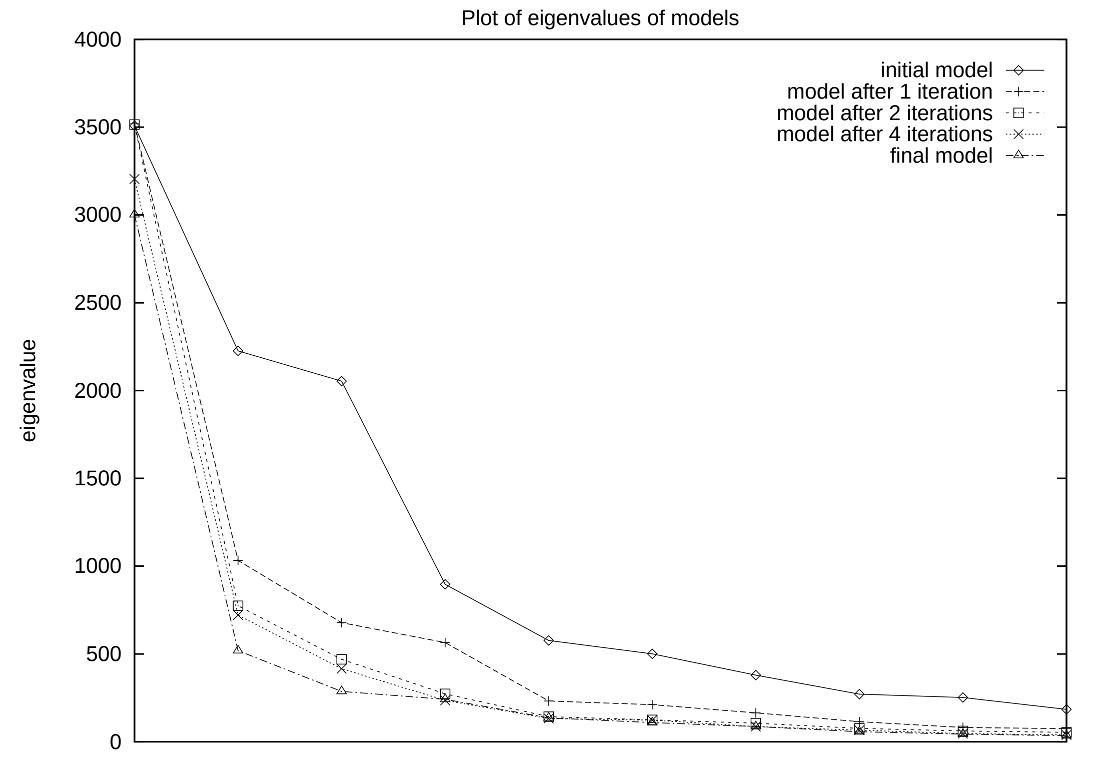


Figure 5.13: Plot showing the largest 10 eigenvalues for successive shape models

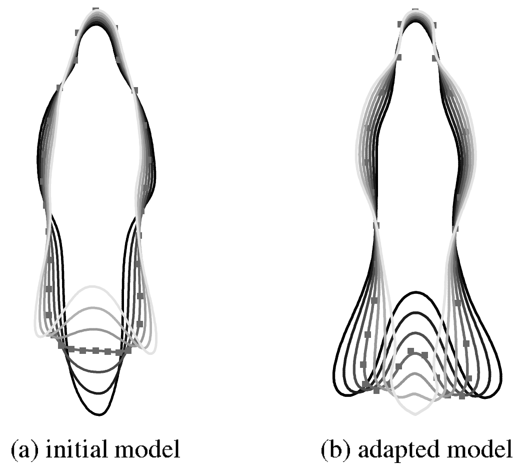

(a) initial model (b) adapted model

Figure 5.14: Principal modes of variation for generic models

5.5 Results of tracking with the adapted models

5.5.1 Quantitative results

The experiments from chapter 4, testing the tracking system on noisy input images, were repeated with the new "adapted" models. Figure 5.15 shows a plot of the output SNR against the input SNR for a noisy test sequence using the initial and adapted models ("generic" and "specific"). The noise was temporally uncorrelated. As before, the number of shape modes used for each model was chosen so that over 95% of the total variance of the appropriate training set was encapsulated. From the graphs it is clear that the new models are an improvement over the original ones with a significant increase in performance for all the noisy sequences tested.

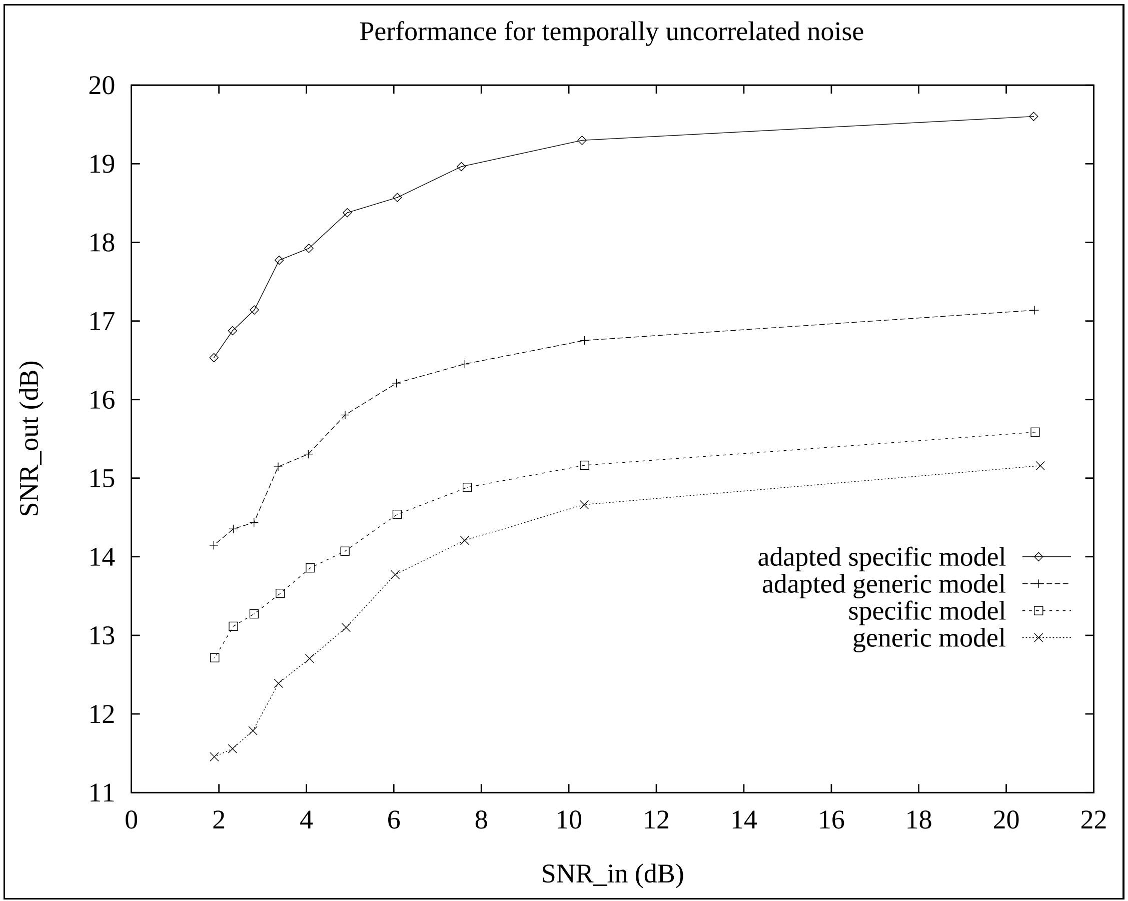


Figure 5.15: Plot showing accuracy of models

Figure 5.16 shows the results of the same experiments with temporally correlated noise (simulating scene occlusion). Again, the results show a significant increase in performance for the new adapted models.

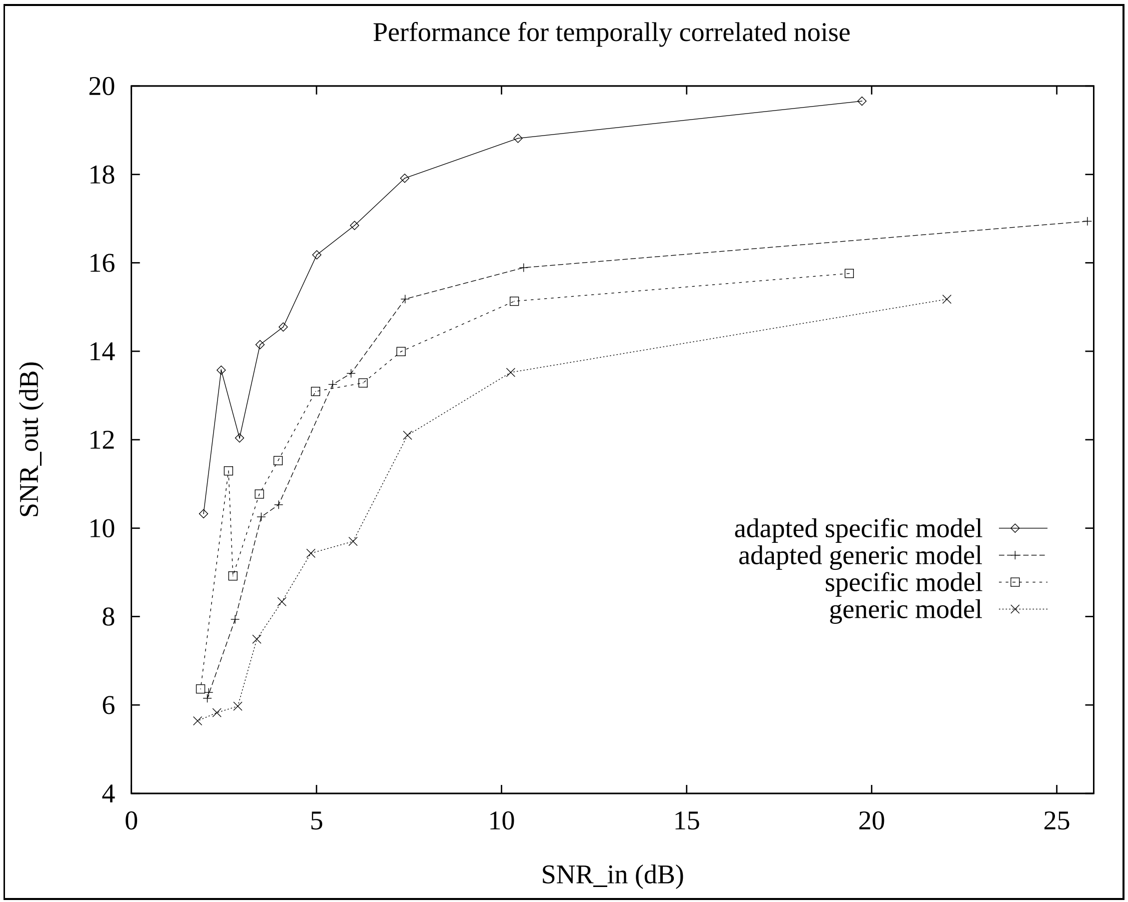


Figure 5.16: Plot showing accuracy of models with partial occlusion

5.5.2 Qualitative results

Results of applying the new adapted specific model, processing a test sequence of a person walking across the image (left to right), are given in figure 5.17. As before, the estimated contour is superimposed over the image. The system was run using only 4 shape parameters.

Two of the "difficult" sequences, along with the tracked contours superimposed, are shown in figures 5.18 and 5.19. The new adapted generic model was used in both cases with 5 shape parameters. The system performs well in both cases and the contour appears to be a better fit to the underlying pedestrian silhouette than obtained previously.

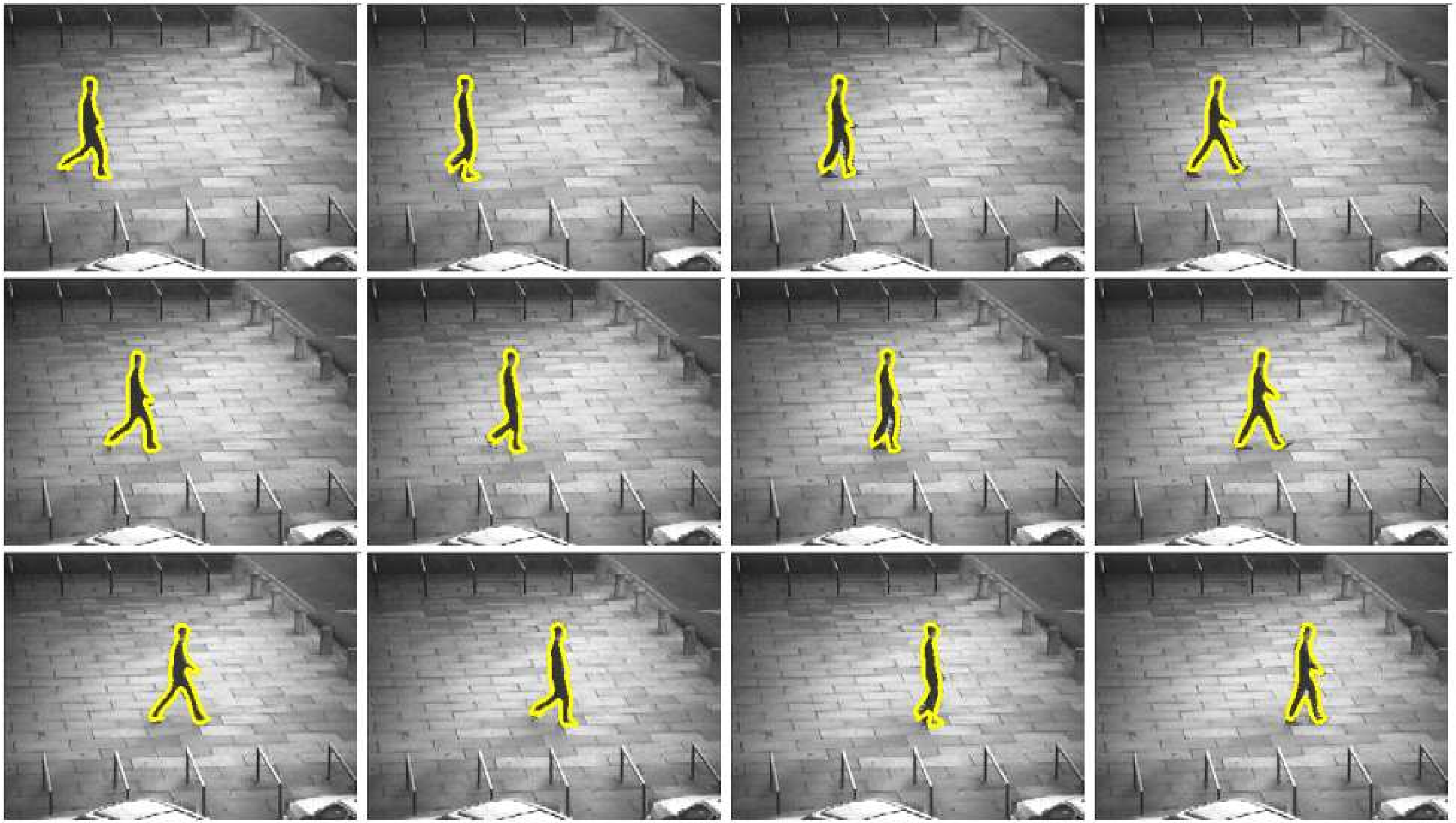


Figure 5.17: Results using adapted model on 2nd test sequence

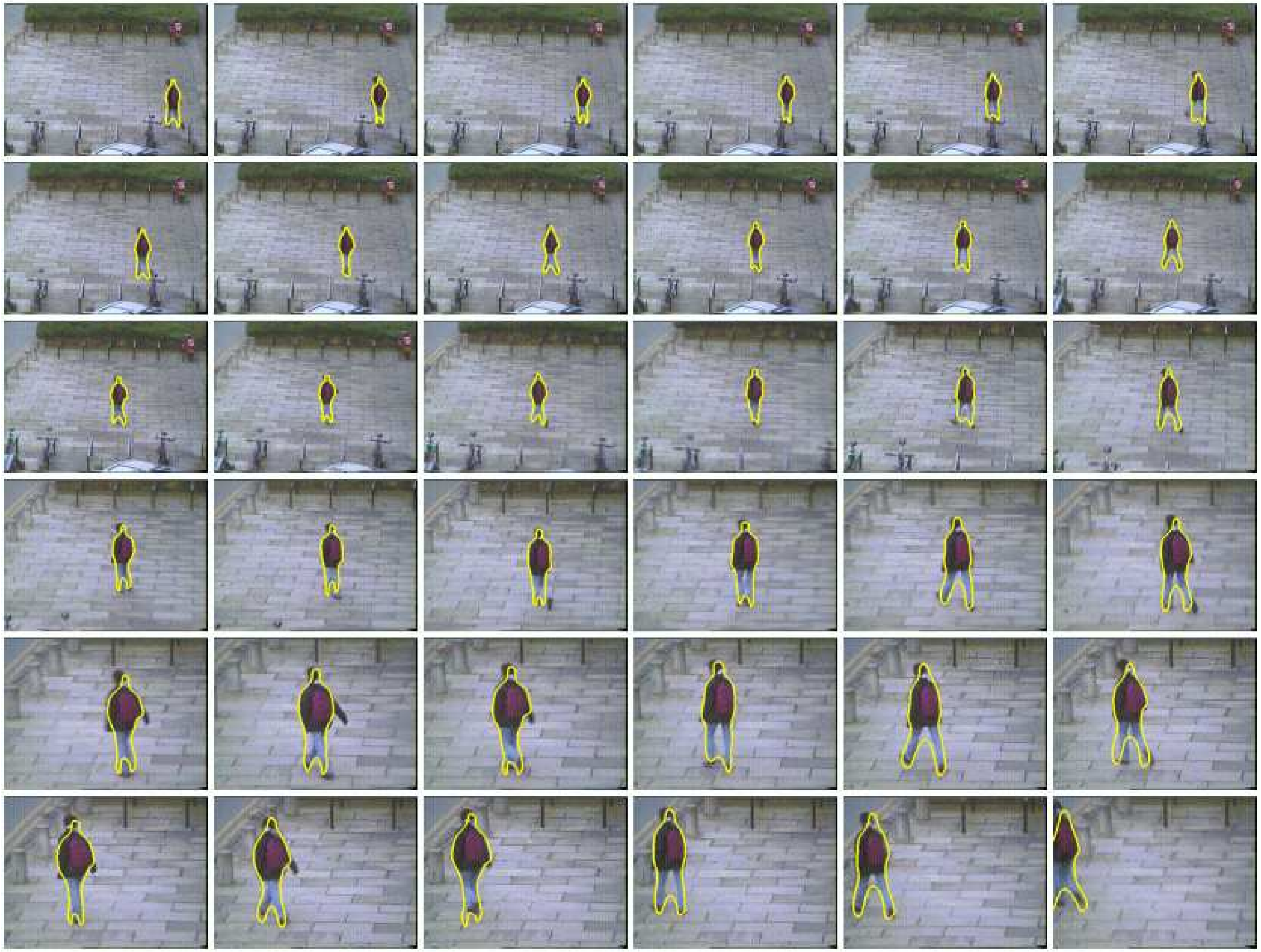


Figure 5.18: Results on test sequence with zooming camera and adapted model

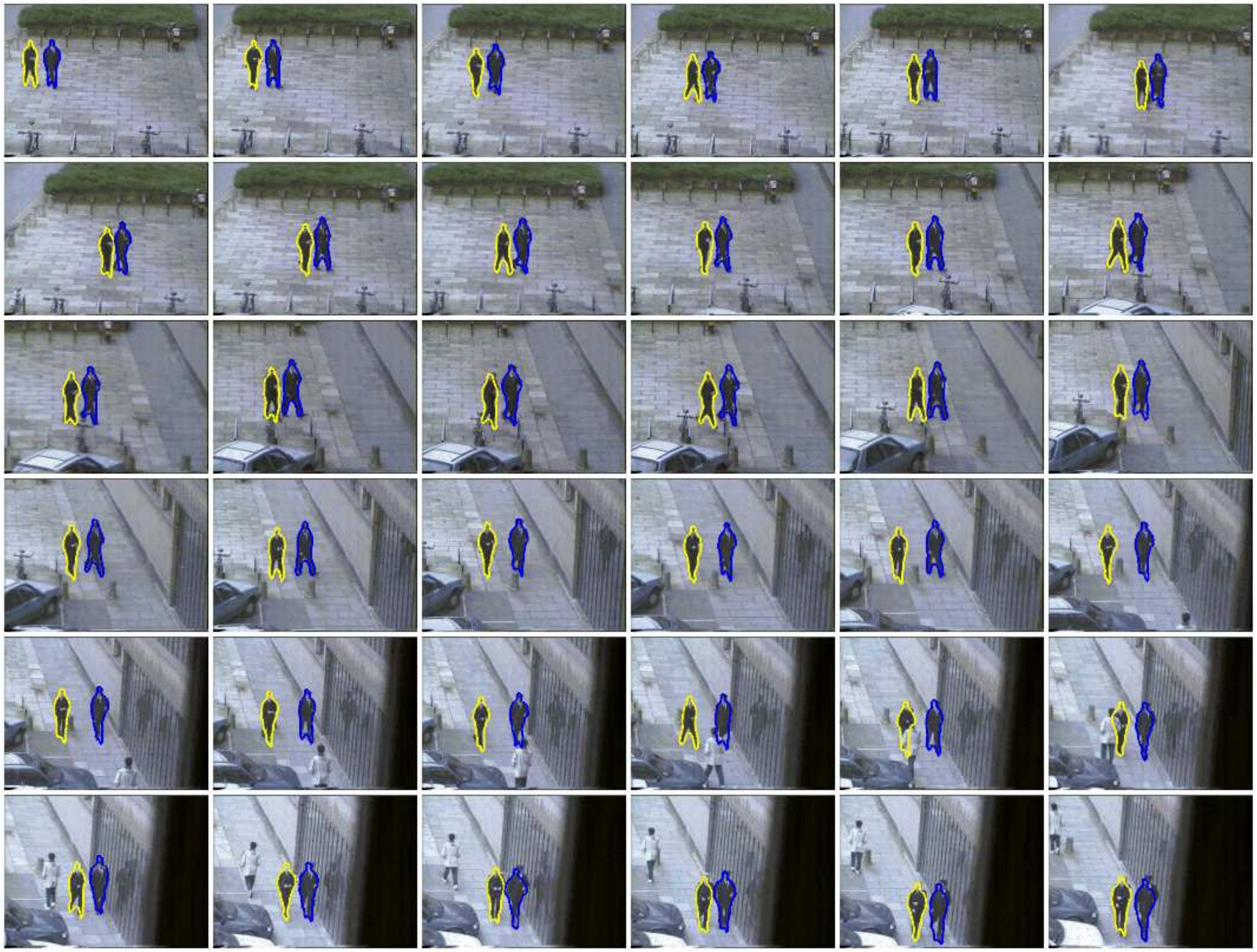


Figure 5.19: Results on test sequence with moving camera and adapted model

5.6 Discussion

In this chapter a novel method for generating a compact shape model has been described. The major advantage of this method is that once a rough initial model has been generated the refinement process can be run on unprocessed images (assuming a rough position, orientation and scale is known). Hence, the model refinement and training shape extraction steps can be combined so that the improved model can be used to extract more accurate training shapes which are then used to generate a more accurate model. This process will only work if the initial shape model is sufficiently robust. A poor initial model will allow the contour to become tangled and the resulting segmentation to be poor, causing the system to diverge from the optimal solution. This was not found to be the case when the initial model generated in chapter 3 was used and the system was found to converge quickly.

From figure 5.13 it is clear that the adapted models become more compact and that the total variance of the training data decreases resulting in a more robust model. The adapted models have been shown to give better results for processing new sequences that were not used in the training phase. Hence a compact, linear shape model has been automatically generated and this model has proved to be useful in the application of tracking human motion.

Chapter 6

A spatiotemporal extension

6.1 Introduction

Previous chapters have demonstrated methods for modeling and subsequent tracking of flexible shapes based on purely spatial representations. The tracking system described in chapter 4 predicts the position and shape of a contour using a simple stochastic model that assumes a stable underlying velocity. The changes in shape parameters between successive frames are assumed to vary randomly with zero mean. Consequently, the predicted shape at a given frame is set to the shape obtained from the previous frame. Hence the prediction is often inaccurate and is based on the results on the previous frame without taking into account any trends in the observed shape deformations over time.

In the application of tracking human motion it should be possible to obtain a more accurate prediction for the shape, based on the previous observations and domain knowledge about how the object deforms. This is particularly apparent in restricted environments such as in pedestrian scenes where all the objects of interest are walking people. One such spatiotemporal model is the WALKER model described by Hogg [3]. Hogg represents instantaneous shape in terms of joint angles of a 3D model. A complete walk cycle is modeled by periodic functions of these joint angles with respect to a walk cycle parameter. One problem with this approach is that a hand generated model is required for each activity of interest.

In this chapter the contour shape representation described previously is extended. A training set of motions is used to build a spatiotemporal model allowing more accurate temporal extrapolation of shape. By improving the estimate of object shape at a given frame, a smaller search window can be used for feature search, reducing the chances of the contour being "distracted" by background features and improving the robustness of the tracking system. Furthermore, a spatiotemporal model allows information to be integrated over time giving more reliable results. Such a model also has the potential to eliminate plausible shapes that do not deform over time as expected and are thus unlikely to be the object of interest.

The method described in this chapter is related to the recent work of Blake and Isard [57, 58] in which a contour tracker is trained on motion sequences to build a stochastic model. In their work, Blake *et al* generate an unconstrained complex 2nd order stochastic model. Such a system can not, in general, be decoupled into a set of independent orthogonal modes and hence the resulting tracking system will be computationally expensive for complex objects that deform in a high dimensional shape space. In contrast, the system described here is based on a physical model which can be decoupled into vibration modes that can be treated independently.

By considering an object as a physical system with internal forces it is possible to model the evolution of the system over time under the action of external forces. Hence, given a reasonably accurate physical model of an object, it is possible to predict how the object will deform over short time intervals (such as between image frames) assuming the external forces are not significant over this time interval. Such a physically-based approach is exemplified by the application of Finite Element Methods (FEM's) by Pentland and Horowitz [26] described in section 2.3. In this work an object represented by a nodal parametrisation is modeled as an elastically deformable physical object with assumed density and elastic properties (i.e. known stress and strain matrices). Modal analysis is used to generate a compact, reduced basis of "vibration modes" for object tracking and data approximation based on the assumed physical properties of the object.

In this chapter, a novel method is described for generating physically based vibration modes from a set of training examples of an object deforming, tuning the elastic properties of the model to reflect how the object actually deforms. The method calculates the optimal stiffness and damping matrices that describe the motion observed in the training data. The resulting "Trained Vibration Mode Model" provides a good basis for the types of motion represented in the training set (e.g.

walking). The model retains the benefits of conventional modal analysis (e.g. low dimensional parametrisation, decoupled filter mechanism for rapid tracking), whilst utilising the training information to improve accuracy. The training set removes the necessity for using theoretical physical assumptions about the object (e.g. modeling a walking person as a simple lump of elastic "clay") resulting in improved vibration modes that reflects how the object actually deforms.

6.2 Learning by example

6.2.1 Training data

It is assumed that training data can be generated in which nodal (or point) displacements for an object have been tracked over short intervals of time allowing derivatives to be calculated. It is also assumed that the nodal points have been matched throughout the training set and that the training information has been rotated and scaled to some normal frame (e.g. using the Hotelling transform, see [21]). Each training shape is represented by n nodes in d dimensions. Hence the training set consists of a set of matched, aligned shape vectors consisting of nodal (or point) positions observed over short intervals of time. e.g. a set of shape vectors $\mathbf{x}^{(\mathbf{k})}$ each representing n control points in $d = 2$ dimensions

$$\mathbf{x}^{(0)} = (P_x^1, P_y^1, \ldots, P_x^n, P_y^n)$$

with $\mathbf{x}^{(0)}$, $\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}$ observations of the nodes at time $t = 0, \Delta t, 2\Delta t$. From this data set, a set of nodal displacements $\mathbf{u}^{(k)}$ is extracted by subtracting off the mean shape vector. The corresponding nodal velocities $\dot{\mathbf{u}}^{(k)}$ and nodal accelerations $\ddot{\mathbf{u}}^{(k)}$ are then calculated by finite difference approximations.

One approach to generating this training data would be to utilise previous approaches such as standard modal analysis or other mesh-like deformable models (described by Terzopoulos *et al* [37]) applied to good quality training images. Alternatively point data can be hand-generated, although this would be laborious.

The method chosen for generating training data was to apply the spatial models generated in previous chapters to good quality training images.

6.2.2 Eliminating the mass matrix

The object to be modeled is currently assumed to have a constant (uniform) density ρ, and the mass matrix M is calculated in the usual way (see, for example, Bathe [25]) using

$$M_{i,j} = \rho \int H_i(u) H_j(u) du$$

where $H_i(u)$ is the interpolation function for the i'th nodal parameter. Without loss of generality, unit density is assumed with $\rho = 1$ since any uniform change in density can be incorporated into the stiffness matrix. Hence, the above mass matrix M is identical to the symmetric matrix $\mathcal{M}$ defined in chapter 3.

The mass matrix defines an inner product and an associated distance metric that measures the "error" between two parametrised curves ($d = 2$) or surfaces ($d = 3$) as follows.

$$\langle \mathbf{U}, \mathbf{U}' \rangle = \mathbf{U}^T \mathcal{M} \mathbf{U}' \tag{6.1}$$

where $\mathbf{U}$ and $\mathbf{U}'$ are the vectors of nodal displacements representing the two curves as in section 2.3.

In order to simplify the problem we consider the mapping

$$\mathbf{V} = \mathcal{M}^{\frac{1}{2}} \mathbf{U} \tag{6.2}$$

where $\mathcal{M}^{\frac{1}{2}}$ is the positive definite square root of the matrix $\mathcal{M}$. Note that $\mathcal{M}$ and $\mathcal{M}^{\frac{1}{2}}$ are both real, symmetric, positive-definite, invertible matrices.

Substituting equation 6.2 into equation 6.1 gives

$$\langle \mathbf{U}, \mathbf{U}' \rangle = \mathbf{V} . \mathbf{V}'$$

where $\mathbf{V}.\mathbf{V}'$ is the standard dot product. The training data is mapped to a new data set $\mathbf{v}^{(k)} = \mathcal{M}^{\frac{1}{2}} \mathbf{u}^{(k)}$.

Assuming an unbiased, homogeneous, isotropic Gaussian noise model for the unmapped data, it can be shown that the associated noise covariance matrix, R_U, is proportional to $\mathcal{M}^{-1}$ (see Blake *et al* [38]). The associated covariance matrix for measurements in "V-space", R_V, is given by

$$R_V = [(\mathcal{M}^{-\frac{1}{2}})^T R_U^{-1} (\mathcal{M}^{-\frac{1}{2}})]^{-1} = rI \tag{6.3}$$

i.e. some scalar multiple of the identity matrix.

Note that the density is assumed to be uniform with respect to the nodal parametrisation (as opposed to uniform spatial distribution of mass). Hence, for a given object it is assumed that nodes are equally distributed over the mass of the object. The mass matrix can be regarded as modeling the sensor characteristics, since an unbiased uniform isotropic sensor will measure M-orthogonal vibration modes independently (i.e. measurements for each mode are uncorrelated). For objects with unknown significantly non-uniform density, it is hoped that by using a feedback mechanism similar to that described in chapter 5, it will be possible to ensure the training shapes are reparametrised with the nodes equally distributed over the object's mass.

6.2.3 Generating vibration modes

We are not concerned with explicitly obtaining the mass, damping and stiffness matrices M, C and K but in generating the associated vibration modes of the system. Making the substitution defined in equation 6.2, the governing equation for the finite element system (equation 2.11) can be rewritten in the form

$$\ddot{\mathbf{V}} + B\dot{\mathbf{V}} + A\mathbf{V} = \mathcal{M}^{-1}\mathbf{S}$$

where

$$\begin{aligned} B &= \mathcal{M}^{-\frac{1}{2}}C\mathcal{M}^{-\frac{1}{2}} & \qquad \mathbf{S} &= \mathcal{M}^{\frac{1}{2}}\mathbf{R} \\ A &= \mathcal{M}^{-\frac{1}{2}}K\mathcal{M}^{-\frac{1}{2}} & \qquad \mathbf{V} &= \mathcal{M}^{\frac{1}{2}}\mathbf{U} \end{aligned}$$

and assuming Rayleigh damping

$$B = b_0 I + b_1 A$$

The basic idea of the training method is to assume there are no external forces (i.e. the observed deformations are simply a sum of the object's free vibrations) with some random noise present incorporating measurement noise as well as the effect of input and internal disturbance. Hence, the quantity

$$\langle M^{-1}\mathbf{R}, M^{-1}\mathbf{R}\rangle = (M^{-1}\mathbf{S}).(M^{-1}\mathbf{S})$$

(the observed "external acceleration") is minimised over the training set. The following error function is minimised

$$J(A, b_0, b_1) = E\left(|\ddot{\mathbf{v}}^{(k)} + B\dot{\mathbf{v}}^{(k)} + A\mathbf{v}^{(k)}|^2\right) \tag{6.4}$$

where $E(...)$ is the expectation (or averaging) operator over the data set and $|.|$ is the standard Euclidean norm.

In fact, this is an off-line system identification problem where the residual error covariance matrix (in "V-space") has been shown to be proportional to the identity matrix (equation 6.3). Hence the ordinary least squares estimate is also the minimum variance estimate (see, for example, Sinha and Kuszta [59]).

For a physically plausible solution, the stiffness matrix is symmetric and hence the matrix A is constrained to be a real, symmetric matrix. i.e. $A^T = A$. The symmetric constraint ensures the resulting modes are real and orthogonal and hence the 2nd order $dn \times dn$ system is decoupled into dn independent 2nd order systems. Note that in this formulation the stiffness matrix K is not further constrained to be banded as in the purely theoretical, physical model. Physically this corresponds to virtual springs attached between non-adjacent as well as adjacent points. Thus, an object is modeled to be a dense set of points (represented by some nodal parametrisation with n nodes) where each point can be displaced about a rest position and is connected via springs to every other point.

6.2.4 Solving the constrained minimisation problem

In order to solve equation 6.4 subject to the constraint $A^T = A$, the matrix A is parametrised in terms of $\frac{n}{2}(n+1)$ parameters $\{a_{i,j} \quad : \quad i \quad \geq \quad j\}$ and the *unconstrained* minimisation of $J(a_{0,0}, a_{1,0}, a_{1,1}, a_{2,0}, ..., b_0, b_1)$ is solved.

As the training set may be large, equation 6.4 is expanded to the form

$$J = \sum_{i,j} S^{22}_{i,i} + B_{i,j}(BS^{11})_{i,j} + A_{i,j}(AS^{00})_{i,j} + 2B_{i,j}S^{12}_{j,i} + 2A_{i,j}S^{02}_{j,i} + 2A_{i,j}(S^{01}B^T)_{j,i} \quad (6.5)$$

where the $n \times n$ matrices S^{**} need only be calculated once for a given training set and are given by

$$S^{00} = E(\mathbf{v}\mathbf{v}^T) \quad S^{01} = E(\mathbf{v}\dot{\mathbf{v}}^T) \quad S^{02} = E(\mathbf{v}\ddot{\mathbf{v}}^T)$$
$$S^{11} = E(\dot{\mathbf{v}}\dot{\mathbf{v}}^T) \quad S^{12} = E(\dot{\mathbf{v}}\ddot{\mathbf{v}}^T)$$
$$S^{22} = E(\ddot{\mathbf{v}}\ddot{\mathbf{v}}^T)$$

Analytic expressions for the partial derivatives of J are easily derived and a standard local optimisation routine used to perform the minimisation. A quasi-Newton conjugate gradient method was used (see, for example, Ciarlet [60]). The problem can be simplified a little by ignoring damping effects (i.e. setting $B = 0$). The assumption of Rayleigh damping can be extended to Cauchy damping by adding higher order terms to the series B(A).

Any minimisation scheme used to solve the problem may converge to a non-optimal local minimum. The minimisation scheme requires a reasonable initial estimate of the solution to ensure that the numerical solution is useful. To find the initial estimate we project the global unconstrained solution into the constrained solution space. The global solution $\tilde{A}$, $\tilde{B}$ minimises the error function

$$\tilde{J}(A, B) = E\left(|\ddot{\mathbf{v}}^{(k)} + B\dot{\mathbf{v}}^{(k)} + A\mathbf{v}^{(k)}|^2\right)$$

and is given by

$$\begin{pmatrix} \tilde{A} \\ \tilde{B} \end{pmatrix} = -\begin{pmatrix} S^{00} & [S^{01}]^T \\ S^{01} & S^{11} \end{pmatrix}^{-1} \begin{pmatrix} [S^{02}]^T \\ [S^{12}]^T \end{pmatrix}$$

The initial estimate, $A^{(0)}$, is calculated by projecting $\tilde{A}$ into the space of symmetric matrices. i.e.

$$A^{(0)} = \frac{1}{2}\left(\tilde{A} + \tilde{A}^T\right)$$

The initial estimates for b_0 and b_1 are calculated by solving the minimisation of $J(A, b_0, b_1)$ with A fixed equal to $A^{(0)}$. Alternatively, the untrained theoretical physics based model can be used to generate mass and stiffness matrices which can be used to calculate initial estimates for the matrices A and B.

Once the local optimisation scheme has converged the vibration modes ϕ_i are calculated from the eigenvectors of A, ψ_i, using

$$\phi_\mathrm{i} = \mathcal{M}^{-\frac{1}{2}}\psi_\mathrm{i}$$

and these trained vibration modes can be utilised in the usual way (see Pentland *et al* [26]).

6.3 Implementation

6.3.1 The local optimisation scheme

A conjugate gradient algorithm for optimising a function of several variables using 1st derivatives (NAG [1] function E04DGF) was used. The error function J takes $(dn(dn-1)/2+2)$ parameters corresponding to the stiffness matrix parameters $a_{i,j}$ and the damping parameters b_0 and b_1.

It is convenient to parametrise the symmetric matrix A using

$$A_{i,j} = \begin{cases} a_{i,j} & i > j \\ 2a_{i,i} & i = j \\ a_{j,i} & i < j \end{cases}$$

The free parameters are stored in a single concatenated state vector. The local optimiser routine requires a single function for evaluating J and its partial derivatives. The function is implemented using the following scheme:-

- Unpack the state vector to reconstruct the matrix A and the parameters b_0 and b_1.
- Construct $B = b_0 I + b_1 A$
- Evaluate J using equation (6.5)
- Calculate the matrix X given by

$$X = S^{02} + S^{01}B^T + AS^{00} + b_1\left(S^{12} + BS^{11} + AS^{01}\right)$$

- Calculate a matrix of partial derivatives using

$$\frac{\delta J}{\delta a_{r,s}} = 2\Big[X + X^T\Big]_{r,s}$$

- Calculate the remaining partial derivatives using

$$\begin{aligned} \frac{\delta J}{\delta b_0} &= 2\text{tr}\left(S^{12} + BS^{11} + AS^{01}\right) \\ \frac{\delta J}{\delta b_1} &= 2\text{tr}\left(AS^{12} + AS^{11}B^T + AS^{01}A^T\right) \end{aligned}$$

where tr(...) is the trace of a matrix.

[1] NAG is a registered trademark

- Pack the matrix $\frac{\delta J}{\delta a_{r,s}}$ and the terms $\frac{\delta J}{\delta b_0}$, $\frac{\delta J}{\delta b_1}$ into a state gradient vector.

The packing scheme simply reads off the lower triangle (including the diagonal) of the matrix term and concatenates the remaining two scalar terms into a single vector. By calculating all the partial derivatives simultaneously, the computational expense of the scheme is significantly reduced.

6.3.2 Reducing the initial dimensionality

When the number of nodes is large the method may appear computationally expensive. However in many cases the object shape does not vary arbitrarily within the high dimensional shape space and the dimensionality of the problem may be reduced by using the Karhunen-Loeve transform (i.e. Principal Component Analysis). This step involves reparametrising the training shapes $\mathbf{v}^{(i)}$ in terms of a truncated basis of n_s spatial eigenvectors and calculating vibration modes as before. This is achieved by transforming the covariance matrices S^{**} using

$$[S^{**}]' = P^T S^{**} P$$

where P is a matrix whose columns are the n_s most significant eigenvectors of S^{00}.

The resulting eigenvectors $\psi_{\mathrm{i}}{}'$ of the $n_s \times n_s$ matrix A are mapped into vibration modes in the full shape space using

$$\phi_{\mathrm{i}} = \mathcal{M}^{-\frac{1}{2}} P \psi_{\mathrm{i}}{}'$$

This step also ensures the problem is well defined in cases where the training set is small compared to the number of nodes used, ensuring that the global solution exists. The optimisation scheme was found to converge within 1 minute on a 100MHz R4000 Indy workstation.

6.4 Results

6.4.1 Artificial data – recovery of SHM

An artificial training set was generated in which a 2D point undergoes simple harmonic motion (SHM) along a 1D axis with a fixed frequency. 2D Gaussian noise was added and the resulting training set processed. Figure 6.1 shows a graph of the relative error in the recovered period of motion against the signal-to-noise ratio (SNR) of the training data (in dB). The relative error converges to zero as the signal-to-noise ratio increases. The method is fairly robust although for accurate modeling it is desirable for the training data to be as noise-free as possible.

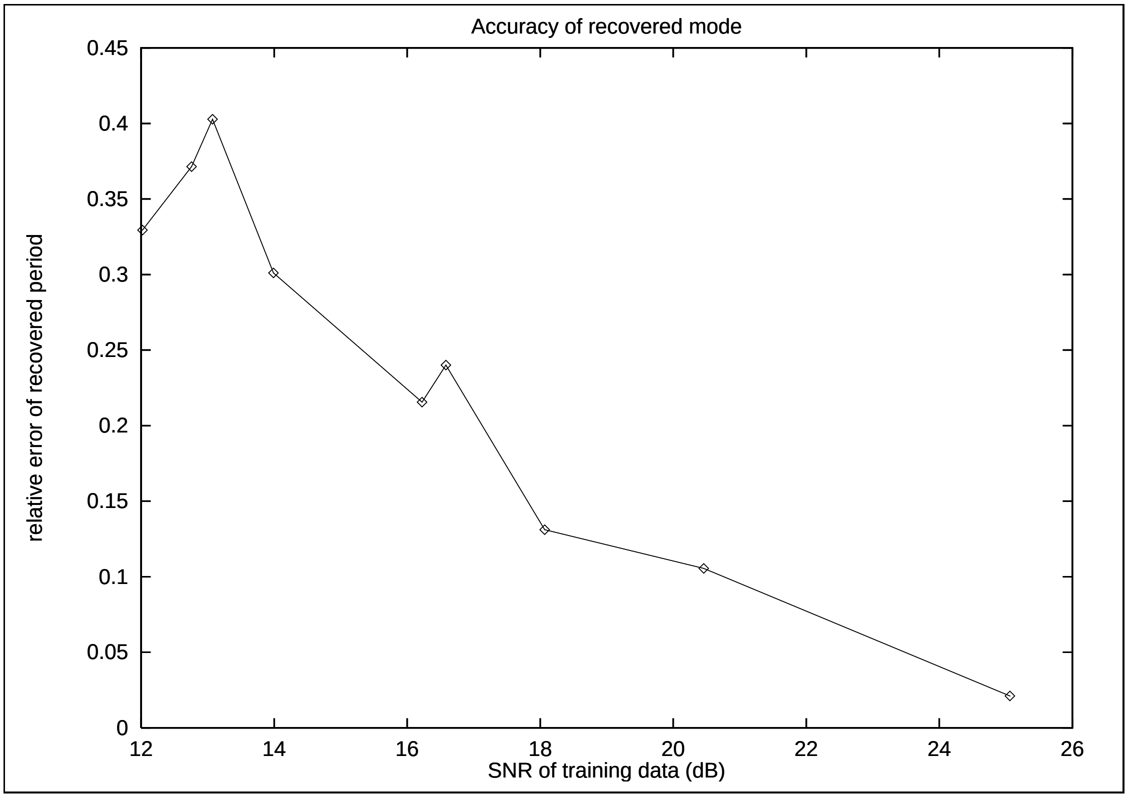


Figure 6.1: Recovery of artificial motion

6.4.2 Real data – one walk

The single walk shape model from chapter 5 was used on the original training images (containing a pedestrian walking across the image plane) to obtain a training set of spline control points for successive image frames.

A subset of the training set for this experiment is shown in figure 6.2. The sequence contains 57 shapes of a pedestrian walking from left to right across the image with each shape represented by a spline with 40 control points. The shapes were aligned about the principal axis and scaled to be a fixed height. The lowest frequency vibration modes generated from this training set are shown in figure 6.3.

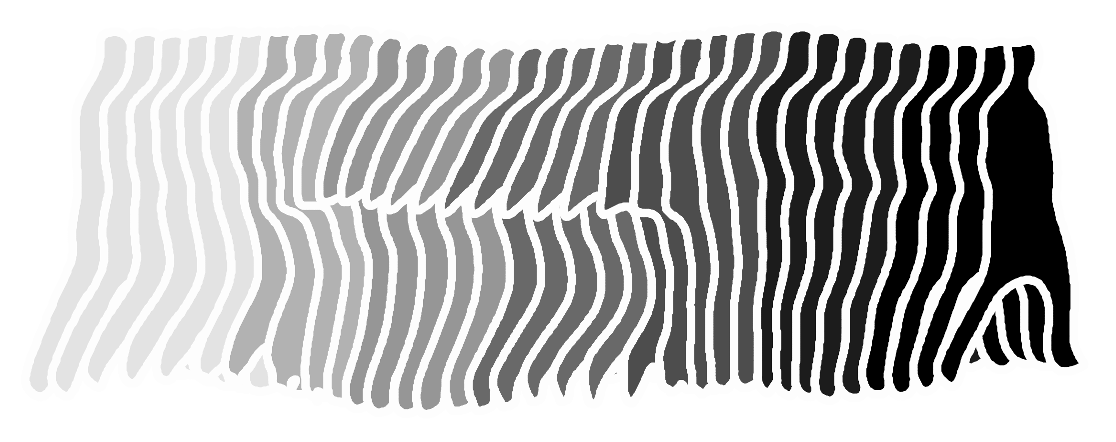


Figure 6.2: Training data

There is some similarity between these spatiotemporal modes and the spatiotemporal surface for a walking person generated by Niyogi and Adelson [44].

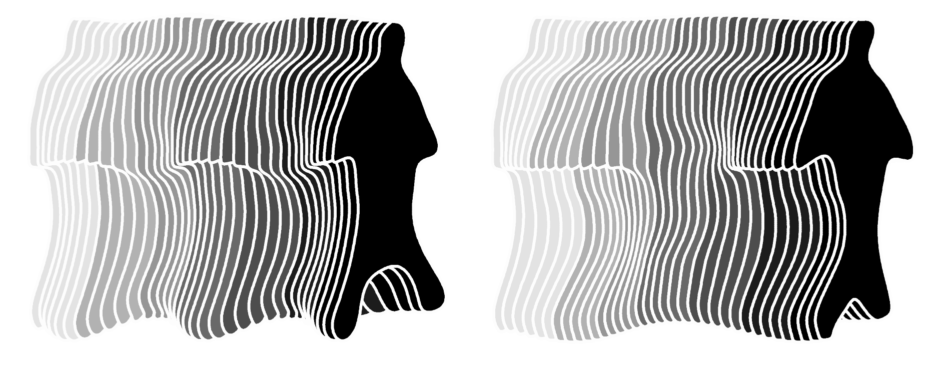


Figure 6.3: Low frequency vibration modes for single walk model

6.4.3 Real data - several walks

A "generic" pedestrian model was created using a training set consisting of a pedestrian walking in a variety of directions. The aim was to build a rough generic model which incorporates spatiotemporal vibration modes approximating the various types of motion observed. To account for the fact that the mean shape for each sequence varies between walks, the nodal displacements were taken with respect to the mean of each sequence (as opposed to the mean over all sequences).

Hence, for a training shape $\mathbf{x}^{(j)}$, taken from the k'th walk sequence the nodal displacement is given by

$$\mathbf{u}^{(j)} = \mathbf{x}^{(j)} - \overline{\mathbf{x}}^{(k)}$$

where $\overline{\mathbf{x}}^{(k)}$ is the mean shape vector for the k'th walk sequence.

A low frequency vibration mode is shown in figure 6.4. For visualisation purposes the vibration mode shows the nodal displacements relative to the mean shape over all the sequences.

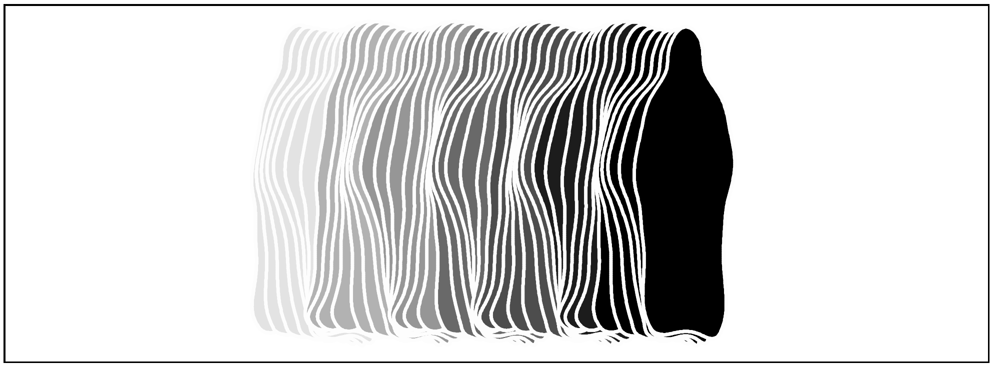


Figure 6.4: Low frequency vibration modes for generic model

6.4.4 Fitting the low dimensional model to new input data

A sequence of 10 consecutive data frames was selected from a new shape sequence not used in the training set. An attempt was then made to represent this data using the vibration modes with fixed amplitude and phase. Hence, only two parameters were calculated for each vibration mode in order to approximate the whole sequence. A least squares method was utilised minimising the errors in the nodal positions.

A graph of signal-to-noise ratio of the recovered motion (with respect to the original data) against the number of vibration modes used is shown in figure 6.5. Two experiments were carried out using the single pedestrian and generic pedestrian training sets. It is clear that the benefits of utilising additional modes decreases. Note the errors in the nodal positions are small (typically less than 2%) when a reasonable number of vibration modes are used.

Figure 6.6 shows another input sequence and the approximated sequence using the generic model. The nodal errors were minimised over the first 8 frames and the subsequent frames are

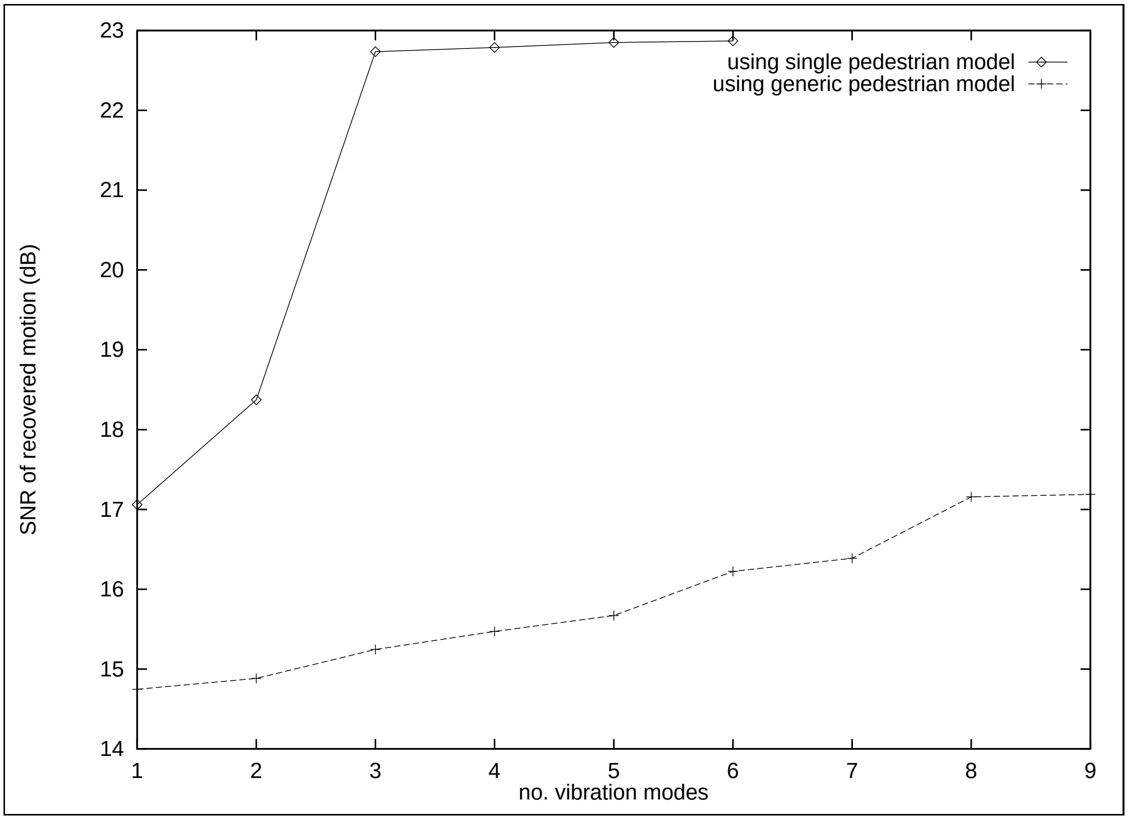


Figure 6.5: Fitting the spatiotemporal models to data

purely extrapolations. For simplicity, nodal displacements were calculated relative to the mean shape over the whole training set. As before, the amplitude and phase of each vibration mode is fixed over the approximated sequence.

6.5 Tracking with the spatiotemporal model

6.5.1 Modifying the tracking system

The vibration eigenmodes can be used as a basis for shape representation in exactly the same way as the eigenvectors obtained previously using spatial statistical analysis. The contour tracking system outlined in chapter 4 can be easily modified to use the trained spatiotemporal model.

As before, the contour is parametrised in terms of a set of m shape parameters $\mathbf{b} = (b_0, ...b_{m-1})^T$ where the shape parameters are now the coefficients for each vibration mode, $\phi_{\mathbf{i}}$. i.e.

$$\mathbf{x} = \sum_i b_i \phi_{\mathbf{i}} + \overline{\mathbf{x}}$$

A dynamic model is used in place of the stochastic shape model used previously. Each shape parameter is treated independently. The model for a given parameter can be expressed by the differential equation:

$$\frac{d}{dt}\begin{pmatrix} b_i \\ \dot{b}_i \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\alpha_i & -\beta_i \end{pmatrix}\begin{pmatrix} b_i \\ \dot{b}_i \end{pmatrix} + \begin{pmatrix} 0 \\ r_i \end{pmatrix}$$

where r_i is a zero-meaned Gaussian variable with variance μ_i and

$$\begin{aligned} \mathcal{M}^{\frac{1}{2}}\phi_{\mathbf{i}} &= \psi_{\mathbf{i}} \\ A\psi_{\mathbf{i}} &= \alpha_i \psi_{\mathbf{i}} \\ B\psi_{\mathbf{i}} &= \beta_i \psi_{\mathbf{i}} \end{aligned}$$

The noise variance μ_i can be chosen by examining the variance of the external acceleration for each vibration mode over the training set or more simply set using a physically based approach as follows

$$\mu_i = \frac{\mu}{|\alpha_i|}$$

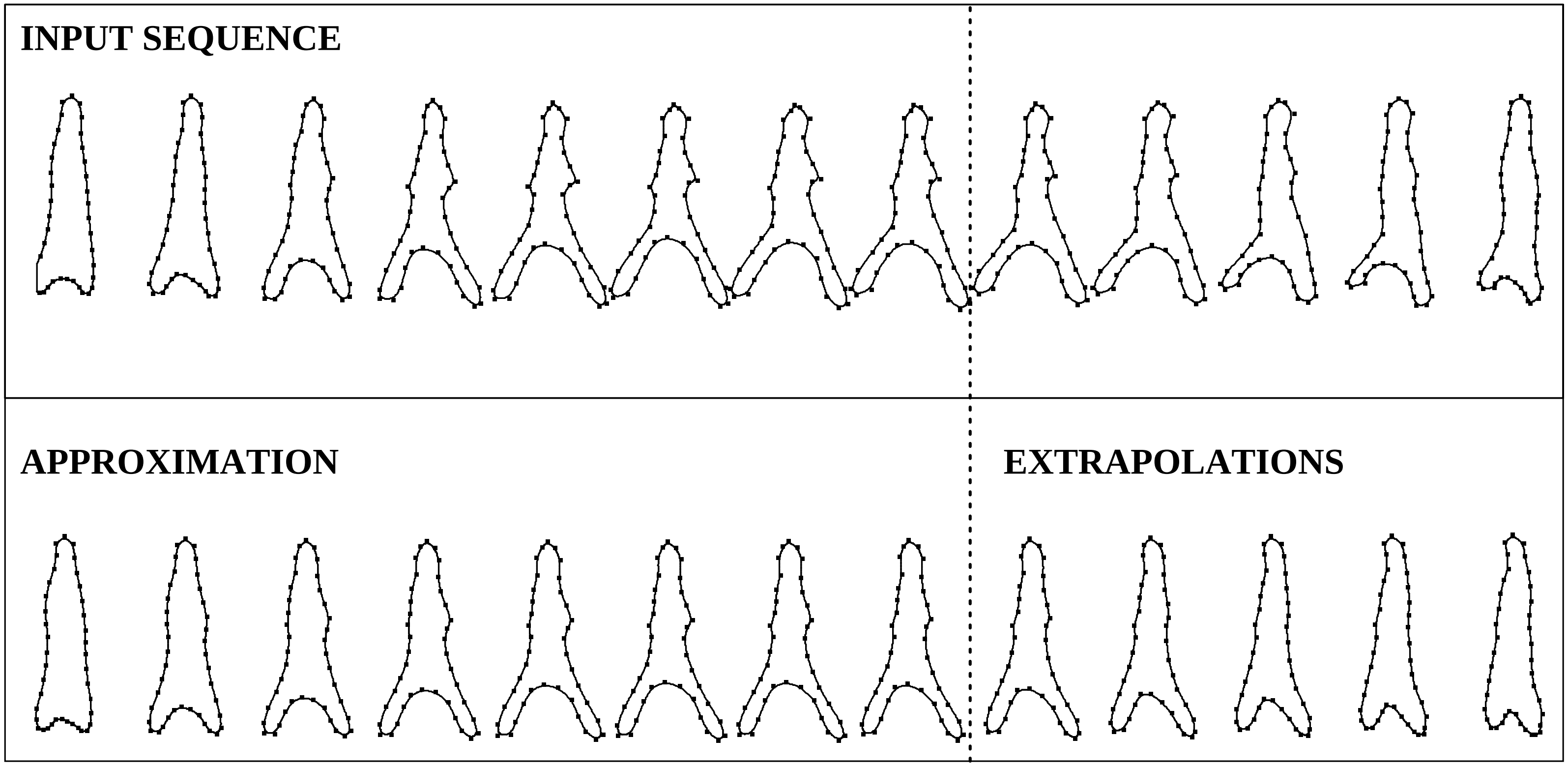


Figure 6.6: Modeling a walk sequence

i.e. the standard deviation of the noise term for the i'th mode is set inversely proportional to the frequency of that vibration mode , allowing the low frequency modes to vary more easily.

A second order Kalman filter is used to update estimates for each coefficient between image frames. Measurements are applied to each filter as described previously in chapter 4.

6.5.2 Results

Quantitative results

The “left-right” trained vibration mode model and the “left-right” adapted spatial model were compared on a segmented sequence containing a pedestrian walking from left to right across the image plane. The sequence was corrupted with temporally correlated noise as described previously in section 4.5.1. A graph showing the performance of the two methods with varying amounts of input noise is given in figure 6.7. Tracking was performed on the same noisy input sequence using each model and the experiments were repeated 20 times for each noise value. It is clear that the spatiotemporal model performs better over the whole range and the difference becomes more acute as the input SNR decreases. The additional information in the spatiotemporal model accounts for this increase in robustness. The noise process results in certain sequences being more difficult to track than others (e.g. when the legs are totally occluded). The same sequences were used to test both models and the results appear consistent over the trials. To obtain a smoother graph, the experiments would have had to be run over a prohibitively large number of trials.

A similar experiment was performed using the generic spatiotemporal model. In this case, the “rest” shape $\overline{\mathrm{x}}$ was also filtered using a simple running average. Temporally correlated noise was only added to the last 15 frames so that a good estimate for $\overline{\mathrm{x}}$ had been found before the system was tested. A graph showing the performance of the spatiotemporal model compared to the adapted spatial model is given in figure 6.8. Experiments were run 20 times for each noise value and it is clear that the spatiotemporal model proves to be more robust for the noisy sequences used.

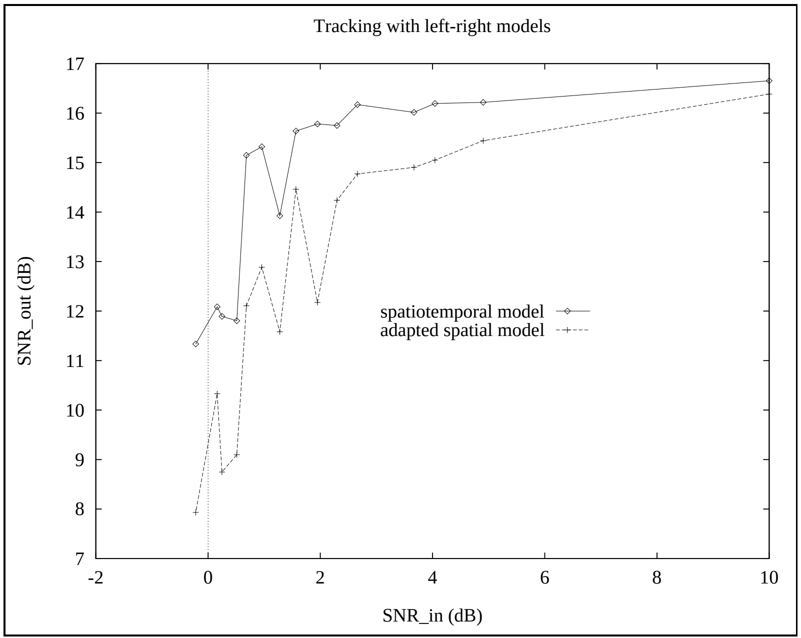


Figure 6.7: Graph showing the robustness of the left-right models

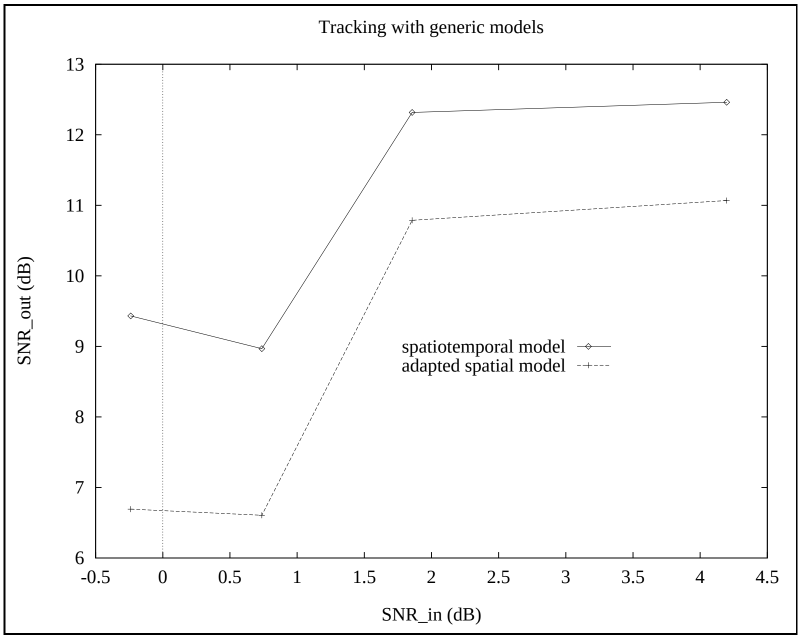


Figure 6.8: Graph showing the robustness of the generic models

6.5.3 Qualitative results

The above image sequence containing a pedestrian walking across the image plane was edited by replacing frames 20 to 30 with the background image. (The pedestrian thus disappears from the sequence for 10 frames, reappearing in the correct position and pose).

The results of tracking on this sequence are shown in figure 6.9 (frames 8, 10, 12, ..., 30 are displayed). The spatiotemporal model correctly estimates the shape of the pedestrian over the "missing" frames so that when the pedestrian reappears the estimated contour is close to the underlying pedestrian shape.

6.6 Discussion

In this chapter a method has been described for automatically generating physics based "vibration modes" for a specific deformable object using training information. The resulting modes are intended to represent the typical motions contained within the training set with a minimal set of M-orthogonal parameters. The method has been shown to be fairly robust to noise and has been applied to a real automatically acquired noisy training set. The use of training data removes the necessity for making a theoretical constant elasticity assumption resulting in improved vibration modes that reflect how the object actually deforms. The method described has potential uses for tracking, recognition and data compression of deformable or articulated objects undergoing complex motions.

The model has been shown to be useful for object tracking in noisy situations where there is partial occlusion. The advantage of using a "vibration mode model" is that a tracking filter mechanism consisting of m independent 2nd order systems can be utilised (each with a 1 dimensional parameter space). The system is robust and fast, requiring only slightly more computational expense than the spatial methods described previously. The increase in robustness is due to the tracking filter's ability to predict shape changes between image frames.

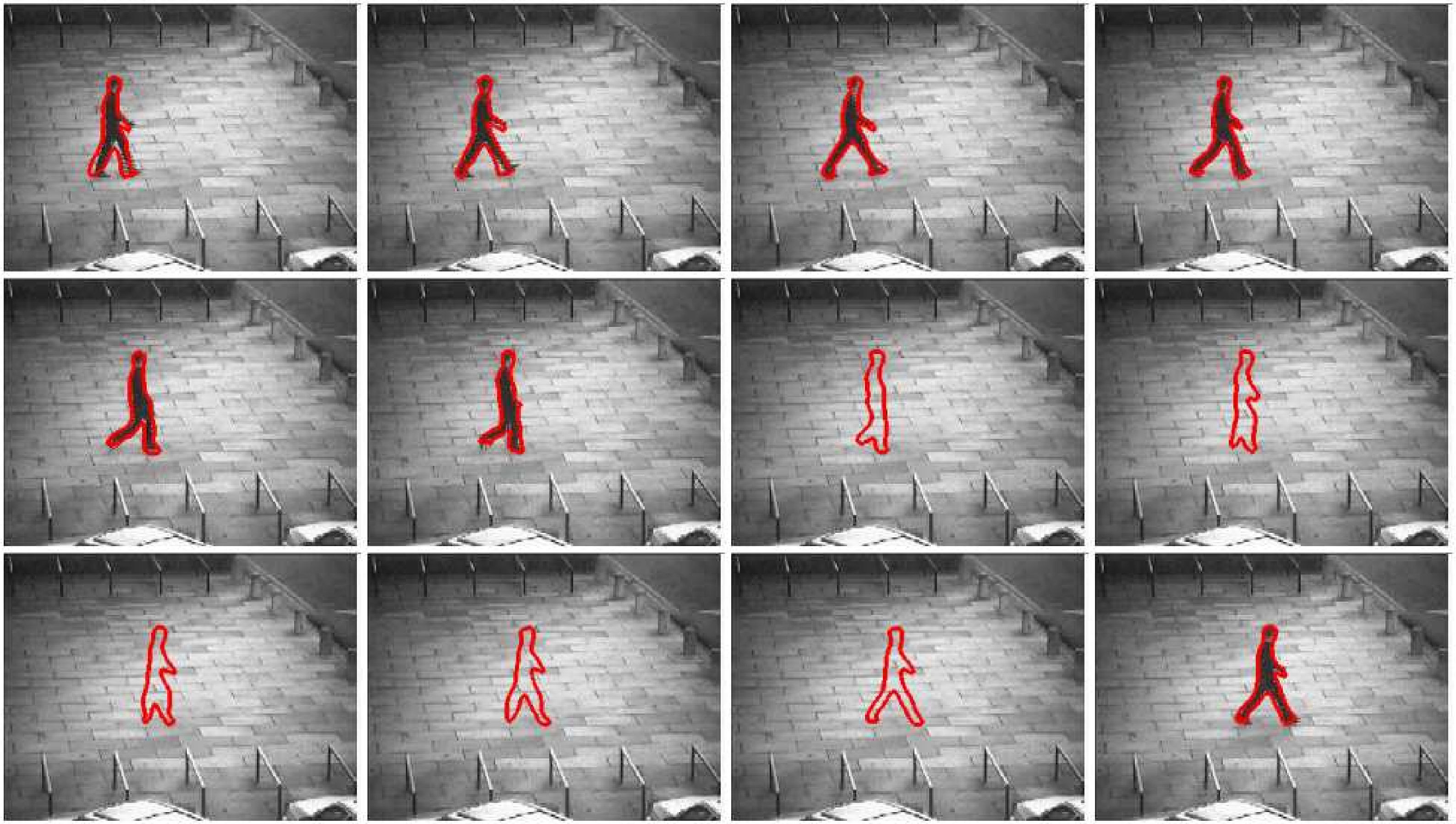


Figure 6.9: Tracking with missing data

Chapter 7

Conclusions

7.1 Summary of work

The work in this thesis addresses the problem of tracking one or more walking pedestrians in natural outdoor scenes. Deformable models are used to represent object shape and these models are learned, automatically, using training data. Results are included, obtained from a prototype tracking system, which demonstrate the potential of the methods for real-time surveillance.

In chapter 3, a method is described for automatically building a linear 2D shape model from sequences of training images of a moving object. The system automatically segments training shapes and labels these shapes using a B-spline representation. Large amounts of data are processed in near real time to generate a compact data set. Statistical component analysis of the spline data gives a simple but effective model. A novel method for performing principal component analysis on a continuous curve is derived, providing a robust theoretical framework for statistical analysis of parametrised contours.

An efficient mechanism for tracking the derived shape parameters is outlined in chapter 4. A Kalman filter mechanism is utilised and the system demonstrated by tracking the silhouette of walking pedestrians through sequences of images. The method has been thoroughly tested on real images and the effect of the system parameters investigated. Qualitative results show that the system successfully tracks several pedestrians in images taken with a moving and zooming camera, where conventional image subtraction methods fail.

In chapter 5, the linear shape model is adapted using an iterative feedback learning scheme. The method is used to resegment and reparametrise the training data producing a more accurate and compact linear shape model. Results are shown using data sets containing the silhouette of a person walking across the image plane and a more general training set containing a person walking in a variety of directions relative to the camera. The performance of the new models is compared with the previous shape models for tracking pedestrian silhouettes. The qualitative and quantitative results show the adapted models are more robust and more accurate.

The spatial linear shape model is extended to a novel spatiotemporal linear model in chapter 6. This model is based on an underlying finite element physical model of an object. Training sequences are used to learn the physical properties of the finite element model. The resulting vibration modes are intended to represent the typical motions contained within the training set with a minimal set of orthogonal parameters. The use of training data allows the theoretical constant elasticity assumption to be unnecessary, resulting in vibration modes that reflect how the object actually deforms. The spatiotemporal model is applied to the problem of tracking a walking pedestrian in noisy situations where there is significant occlusion. Results show that the new spatiotemporal model is significantly more robust than the adapted spatial models. The increase in robustness is due to the tracker's ability to predict shape changes between image frames.

7.2 Discussion

In this thesis, 2D models of shape have been successfully used to track a 3D deforming object from a variety of viewpoints. Changes in apparent shape due to variability in viewpoint are treated as flexibility in 2D shape. This approach benefits from the relative simplicity of 2D algorithms over more complex 3D approaches. In the application of surveillance and human motion analysis it is often not necessary to recover full Euclidean (or even projective) 3D descriptions of the object of interest.

The methods used involve learning techniques using training information. The advantage of such a methodology is that the system can be applied to new problems without requiring complex hand-crafted models to be regenerated. The information implicit in the models allows the system to track robustly in real-world situations where there is background clutter, imaging errors and occlusion. The trained models are still flexible enough to be applied to a broad range of image scenes although the system performs more robustly for "specialised" training sets applicable to a narrow range of shapes or deformations. The techniques described here have been applied to the problem of tracking the outline of a walking pedestrian. However the methods can potentially be applied to a wide range of applications (e.g. tracking farmyard animals, a beating heart muscle etc).

7.3 Future Work

By clustering the training data or simple classification based on direction of motion it may be possible to build a set of more accurate linear models that prove to be more reliable than one single generic model. The use of multiple models requires reliable techniques for switching models and an effective approach to this problem has been outlined by Ahmad *et al* for tracking hand gestures [61]. Future work may look at the potential of such an approach in the application of tracking pedestrians.

Further work is required to investigate whether the physical model identified using the method described in chapter 6 may be successfully applied to temporal medical data sets (such as a beating heart sequence). It is hoped that the recovered physical parameters of the model obtained from observed training motion may contain useful information for medical research and clinical diagnosis. The trained vibration mode method may also be significantly improved by incorporating a feedback scheme similar to the adapted spatial method outlined in chapter 5.

It is hoped that the methods described in this thesis will provide a sound basis for building a usable surveillance system. Further methods are required for controlling the initialisation and termination phase of the tracking process, for instance where objects become occluded for significant time periods. One approach to this problem is described by Hutber and Zhang [62]. Such a system will also require high level control systems for recognition of significant events (such as a car being stolen). Johnson and Hogg [63] have successfully used the output of the prototype tracking system, described in this thesis, to learn the distribution of trajectories in an outdoor pedestrian scene for event recognition.

References

[1] G Johansson. Visual motion perception. *Scientific American*, pages 76–88, June 1975.

[2] T F Cootes and C J Taylor. Combining point distribution models with shape models based on finite element analysis. In *British Machine Vision Conference*, volume 2, pages 419–428, 1994.

[3] D Hogg. Model-based vision: A program to see a walking person. *Image and Vision Computing*, 1(1):5–20, 1983.

[4] K Rohr. Incremental recognition of pedestrians from image sequences. *Computer Vision and Pattern Recognition*, pages 8–13, 1993.

[5] M K Leung and H Y Yang. First sight: A human body outline labeling system. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(4):359–377, 1995.

[6] Z Chen and H J Lee. Knowledge-guided visual perception of 3d human gait from a single image sequence. *IEEE Trans. on Systems, Man and Cyb.*, 22(2):336–342, 1992.

[7] A J Bulpitt. *A Multiple Adaptive Resonance Theory Architecture Applied to Motion Recognition Tasks*. D phil, Dept. of Electronics, University of York, 1994.

[8] N Murphy, N Byrne, and K O'Leary. Long sequence analysis of human motion using eigenvector decomposition. In *Proc. SPIE*, September 1993.

[9] M A Turk and A Pentland. Face recognition using eigenfaces. In *Proceedings of CVPR*, pages 586–591, June 1991.

[10] T F Cootes, C J Taylor, A Lanitis, D H Cooper, and J Graham. Building and using flexible models incorporating grey-level information. In *International Conference on Computer Vision*, pages 242–246, May 1993.

[11] T F Cootes and C J Taylor. Modelling object appearance using the grey-level surface. In *British Machine Vision Conference*, volume 2, pages 479–488, 1994.

[12] S Rowe and A Blake. Statistical background modelling for tracking with a virtual camera. In *British Machine Vision Conference*, volume 2, pages 423–433, September 1995.

[13] F Bookstein. *Morphometric Tools for Landmark Data*. Cambridge University Press, 1991.

[14] T J Cootes, C J Taylor, D H Cooper, and J Graham. Training models of shape from sets of examples. In *British Machine Vision Conference*, pages 9–18, September 1992.

[15] T F Cootes and C J Taylor. Active shape models – 'smart snakes'. In *British Machine Vision Conference*, pages 276–285, September 1992.

[16] T F Cootes, A Hill, C J Taylor, and J Haslam. The use of active shape models for locating structures in medical images. In Barrett H.H. and Gmitro A.F., editors, *Information Processing in Medical Imaging*, pages 33–47, 1993.

[17] A Hill and C J Taylor. Automatic landmark generation for point distribution models. In *British Machine Vision Conference*, volume 2, pages 429–438. BMVA Press, 1994.

[18] A Lanitis, C J Taylor, and T F Cootes. An automatic face identification system using flexible appearance models. In *British Machine Vision Conference*, volume 1, pages 65–74, 1994.

[19] C Kervann and F Heitz. Robust tracking of stochastic deformable models in long image sequences. In *IEEE International Conference on Image Processing*, volume 3, pages 88–92. IEEE Computer Society Press, November 1994.

[20] J C Gower. Generalized procrustes analysis. *Psychometrika*, (40):33–51, 1975.

[21] R Gonzalez and R Woods. *Digital Image Processing*. Addison-Wesley Publishing Co., 1992.

[22] D G Lowe. Fitting parameterized three dimensional models to images. *IEEE Trans on Pattern Analysis and Machine Intelligence*, pages 441–450, May 1991.

[23] J Haslam, C J Taylor, and T F Cootes. A probabilistic fitness measure for deformable template models. In *British Machine Vision Conference*, volume 1, pages 33–42, 1994.

[24] T F Cootes, C J Taylor, and A Lanitis. Active shape models: Evaluation of a multi-resolution method for improving image search. In *British Machine Vision Conference*, volume 1, pages 327–336, 1994.

[25] K Bathe. *Finite Element Procedures in Engineering*. Prentice-Hall, 1982.

[26] A Pentland and B Horowitz. Recovery of non-rigid motion and structure. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(7):730–742, July 1991.

[27] A Pentland and S Sclaroff. Closed-form solutions for physically based shape modeling and recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(7):715–729, July 1991.

[28] S Sclaroff and A Pentland. A modal framework for correspondence and description. In *Proc 4th International Conference on Computer Vision*, pages 308–313, May 1993.

[29] S Sclaroff and A Pentland. On modal modeling for medical images: Underconstrained shape description and data compression. In *Proc. IEEE Workshop on Biomedical Image Analysis*, June 1994.

[30] C Nastar. Vibration modes for nonrigid analysis in 3d images. In *European Conference on Computer Vision*, volume 1, pages 231–238, May 1994.

[31] C Nastar and N Ayache. Spatio-temporal analysis of nonrigid motion from 4d data. In *IEEE Workshop on Motion of Non-rigid and Articulated Objects*, pages 146–151. IEEE Computer Society Press, November 1994.

[32] A Gelb, editor. *Applied Optimal Estimation*. MIT Press, 1974.

[33] M Kass, A Witkin, and D Terzopoulos. Snakes: Active contour models. In *First International Conference on Computer Vision*, pages 259–268, 1987.

[34] F Leymarie and M D Levine. Tracking deformable objects in the plane using an active contour model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):617–634, June 1993.

[35] D Terzopoulos and R Szeliski. Tracking with kalman snakes. In A Blake and A Yuille, editors, *Active Vision*, chapter 1, pages 3–20. MIT Press, 1992.

[36] D Terzopoulos, J Platt, A Barr, and K Fleischer. Elastically deformable models. *ACM Computer Graphics*, 4(21):205–214, July 1987.

[37] D Terzopoulos, A Witkin, and M Kass. Symmetry-seeking models for 3-d object reconstruction. *Int.J. of Computer Vision*, 1(3):211–221, 1987.

[38] A Blake, R Curwen, and A Zisserman. A framework for spatio-temporal control in the tracking of visual contours. *International Journal of computer Vision*, 1993.

[39] R Curwen and A Blake. Dynamic contours: Real-time active splines. In Blake A. and Yuille A., editors, *Active Vision*, chapter 3, pages 39–57. MIT Press, 1992.

[40] J M Rehg and T Kanade. Digiteyes: Vision-based hand tracking for human-computer interaction. In *IEEE Workshop on Motion of Non-rigid and Articulated Objects*, pages 194–199. IEEE Computer Society Press, November 1994. IEEE Catalog No. 94TH0671-8.

[41] D Marr and H K Nishihara. Representation and recognition of the spatial organisation of three-dimensional shapes. In *Proc. of R. Soc. London*, volume B, pages 269–294, 1978.

[42] R Jain, W N Martin, and J K Aggarwal. Segmentation through the detection of changes due to motion. *Computer Graphics and Image Processing*, 11:13–34, 1979.

[43] X Li-Qun, D Young, and D Hogg. Building a model of a road junction using moving vehicle information. In *British Machine Vision Conference*, pages 443–452, September 1992.

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

[45] A Worrall and J Hyde. A fast algorithm for background generation. VIEWS Working Paper RU-03-WP-T.1.1.1.1-1.

[46] J D Foley and A Van Dam. *Fundamentals of Interactive Computer Graphics*. Addison-Wesley Publishing Co., 1984.

[47] M Sonka, V Hlavac, and R Boyle. *Image Processing, Analysis and Machine Vision*. Chapman and Hall, 1993.

[48] A Hill, A Thornham, and C J Taylor. Model-based interpretation of 3d medical images. In *British Machine Vision Conference*, volume 2, pages 339–349, 1993.

[49] R Bartels, J Beatty, and B Barsky. *An Introduction to Splines for use in Computer Graphics and Geomteric Modeling*. Morgan Kaufmann, 1987.

[50] P Cohn. *Algebra*. John Wiley and Sons, 1984.

[51] B K P Horn. *Robot Vision*. MIT Press, 1986.

[52] D Koller, J Weber, and J Malik. Robust multiple car tracking with occlusion reasoning. In *European Conference on Computer Vision*, volume 1, pages 189–196, May 1994.

[53] C K I Williams. *Combining deformable models and neural networks for handprinted digit recognition*. PhD thesis, Department of Computer Science, University of Toronto, 1994. (http://neural-server.aston.ac.uk/People/willicki/Welcome.html).

[54] M H Syn and R W Prager. Mesh models for 3 dimensional ultrasound imaging. Technical Report CUED/F-INFENG/TR 210, Cambridge University Engineering Department, Trumpington Street, Cambridge CB2 1PZ, December 1994.

[55] A J Heap and D C Hogg. Extending the Point Distribution Model using polar coordinates. In *Proc. CAIP*, pages 130–137, Prague, Czech Republic, September 1995.

[56] S Haykin. *Neural Networks: A comprehensive Foundation*. Macmillan College Publishing Co., 1994.

[57] A Blake and M Isard. 3d position, attitude and shape input using video tracking of hands and lips. In *Proc. ACM Siggraph*, pages 185–192, 1994.

[58] A Blake, M Isard, and D Reynard. Learning to track the visual motion of contours. *Artificial Intelligence*, 1995.

[59] N K Sinha and B Kuszta. *Modeling and Identification of Dynamic Systems*. Van Nostrand Reinhold Company, 1983.

[60] P Ciarlet. *Introduction to Numerical Linear Algebra and Optimisation*. Cambridge University Press, 1989.

[61] T Ahmad, C J Taylor, A Lanitis, and T F Cootes. Tracking and recognising hand gestures using statistical shape models. In *British Machine Vision Conference*, volume 2, pages 403–413, September 1995.

[62] D Hutber and Z Zhang. Multi-sensor multi-target tracking – strategies for events that become invisible. In *British Machine Vision Conference*, volume 2, pages 463–473, September 1995.

[63] N Johnson and D Hogg. Learning the distribution of object trajectories for event recognition. In *British Machine Vision Conference*, volume 2, pages 583–593, September 1995.