University of Leeds

SCHOOL OF COMPUTER STUDIES

RESEARCH REPORT SERIES

Report 94.11

An Efficient Method for Contour Tracking using Active Shape Models

by

A M Baumberg & D C Hogg

Division of Artificial Intelligence

April 1994

Abstract

There has been considerable research interest recently, in the areas of real time contour tracking and active shape models. This paper demonstrates how dynamic filtering can be used in combination with a flexible shape model to track an articulated non-rigid body in motion. The results show the method being used to track the silhouette of a walking pedestrian across a scene in real time. The active shape model used was generated automatically from real image data and incorporates variability in shape due to orientation as well as object flexibility. A Kalman filter is used to control spatial scale for feature search over successive frames and for contour refinement on an individual frame. Iterative refinement allows accurate contour localisation where feasible, although there is a trade-off between speed and accuracy. The shape model incorporates knowledge of the likely shape of the contour and speeds up tracking by reducing the number of system parameters. A further increase in speed is obtained by filtering the shape parameters independently.

1 Introduction

The Point Distribution Model (or Active Shape Model) described by Cootes et. al. [1] has been successfully used for model-based image interpretation (Cootes et. al. [2], Hill et. al. [3]). This paper deals with the problem of efficiently utilising such a model to track a moving and deforming object through a sequence of video frames. In this case, the object in question is a walking pedestrian observed from a variety of possible viewpoints.

Much of the work on contour tracking relates to the active contour model (or snake) introduced by Kass et. al. [4]. These deformable contours have been used successfully for image interpretation and tracking of non-rigid structures. (For example Leymarie and Levine use snakes in the tracking of living cells [5].) An extension to the snake, incorporating Kalman filtering (see for example Gelb [6], Maybeck [7]), is described by Terzopoulos and Szeliski [8].

Blake et al. [9] set out a mathematical framework for the tracking of contours represented by a B-spline or similar parametrisation based on earlier work by Curwen and Blake [10]. Affine invariance is used to limit the space of favourable shape deformations and a shape template is used to retain some shape memory.

Cootes and Taylor describe an iterative refinement technique for improving the fit of a shape estimate driven by edge information (using Active Shape Models [11]) and later driven by local grey-level profiles [12]. The method described here extends this work by incorporating a Kalman filter into an active shape model which has been acquired automatically. The active shape model used is based on the point distribution model of Cootes except that a B-spline is used to represent object shape. The method is fast and robust allowing gross shape deformations to be tracked across frames whilst accurately localising the object boundary where feasible (i.e. when "lock" is not lost).

In the case of tracking a walking pedestrian, the changes in shape between frames (taken at video rate) are large. Hence the contour can not be assumed to be slowly varying. 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, a model based approach is used.

2 Shape Model

2.1 Model Acquisition

The automatic acquisition of the flexible shape model has been described in previous work (Baumberg and Hogg [13]) where a simple segmentation scheme extracts a large set of noisy training shapes. A training shape is represented by a closed uniform B-spline with a fixed number of control points equally spaced around the boundary of the object. These control points are treated in a similar way to the "landmark" points of Cootes [1]. The training shapes are aligned to a reference shape and the deviations from the mean shape are analysed.

Principal component analysis of the aligned training data results in a model consisting of the mean shape vector $\overline{\mathbf{w}}$ and a subset of l eigenvectors $\mathbf{e}^{(i)}$ with associated eigenvalues λ_i corresponding to the l most significant independent modes of variation in the training data.

Each shape is represented by a spline, $(X(s), Y(s))$ with n control points $(W_x^{(i)}, W_y^{(i)})$. The associated shape vector of length $2n$, is defined by:

$$\mathbf{w} = (W_x^0, W_y^0, W_x^1, W_y^1, \ldots, W_x^{n-1}, W_y^{n-1})$$

The object shape can be parametrised by a set of l shape parameters b_i corresponding to the component of each mode. The corresponding shape vector with control points $(V_x^{(i)}, V_y^{(i)})$ in the model frame is given by:

$$\mathbf{w}' = L\mathbf{b} + \overline{\mathbf{w}}$$

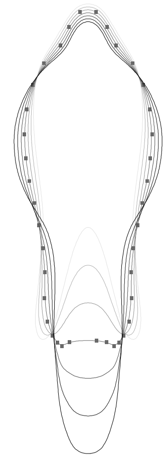


Figure 1: Effect of varying shape parameter b_0

where $\mathbf{b} = (b_0, b_1, \ldots, b_{l-1})$ and L is a matrix of eigenvectors given by:

$$L_{j,i} = \left[\mathbf{e}^{(i)}\right]_j$$

The effect of varying the parameter b_0 corresponding to the principal mode of variation of the model is shown in fig. 1

3 Theoretical Framework

3.1 State Space

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

$$\begin{pmatrix} W_x^{(i)} \\ W_y^{(i)} \end{pmatrix} = \mathcal{M} \begin{pmatrix} V_x^{(i)} \\ V_y^{(i)} \end{pmatrix} + \begin{pmatrix} o_x \\ o_y \end{pmatrix}$$

where the 2 x 2 matrix $\mathcal{M}$ is given by:

$$\mathcal{M} = \begin{pmatrix} a_x & -a_y \\ a_y & a_x \end{pmatrix}$$

Hence, the state space consists of l shape parameters b_i, corresponding to the modal components, the origin of the object (o_x, o_y), the velocity $(\dot{o}_x, \dot{o}_y)$ and the alignment [1] parameters a_x, a_y, incorporating rotation and scaling. The state space to shape vector transformation is given by:

$$\mathbf{w} = M(a_x, a_y)(L\mathbf{b} + \overline{\mathbf{w}}) + \mathbf{o} \qquad (1)$$

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

where $\mathbf{o} = (o_x, o_y, \ldots, o_x, o_y)$ and M is a $2n$ x $2n$ rotation and scaling matrix given by:

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

3.2 Observed Features

Although the object shape is represented by a continuous curve it is convenient to sample the curve at m regular intervals between control points. (Note: m is currently fixed, although different values may be chosen depending on the contour size in image pixels). Hence there are nm points (p_i, q_i) given by:

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

where

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

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

$$\sum_{j=0}^{n} \begin{pmatrix} G_{2i,2j} & G_{2i,2j+1} \\ G_{2i+1,2j} & G_{2i+1,2j+1} \end{pmatrix} \begin{pmatrix} W_x^{(j)} \\ W_y^{(j)} \end{pmatrix} = \begin{pmatrix} X(i/nm) \\ Y(i/nm) \end{pmatrix} = \begin{pmatrix} p_i \\ q_i \end{pmatrix}$$

At each new frame, measurements are made by searching along the normal to the estimated contour at the sample points. The search is restricted to a specified search window obtained from the filtering process (see sec. 5.7). 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. [9]. In order to measure the contrast at a given scale, a sobel operator on the subsampled image was found to be adequate. This method of measuring contrast does not assume the direction of the edge is known.

4 Stochastic Model

The shape part of the state vector is modelled as a simple discrete stochastic process as follows.

$$b_i^{(k)} = b_i^{(k-1)} + w_i^{(k-1)}$$

where $w_i^k \sim N(0, \mu_i)$ and b_i^k models the i'th parameter value at frame k . A dynamic model 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:

$$E(b_i b_j) = \begin{cases} \lambda_i & i = j \\ 0 & \text{otherwise} \end{cases}$$

where $E(...)$ is the expectation or mean value. As the variance of b_i over the training set is equal to λ_i, it is natural to set

$$\mu_i = \mu\lambda_i$$

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.

The origin of the object is assumed to undergo uniform 2D motion with a randomly varying force represented by an additive random noise process. This can be expressed by the differential equation:

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

where $w_x \sim N(0, q_x)$ and a corresponding equation for o_y.

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

4.1 Measurement Model

The curve points $\mathbf{p}$ are related to the state space parameters by:

$$\mathbf{p} = MG(L\mathbf{b} + \overline{\mathbf{w}}) + \mathbf{o} \tag{2}$$

where $M(a_x, a_y)$ is redefined to be the equivalent $2nm$ x $2nm$ rotation and scaling matrix and $\mathbf{o}$ is redefined to be the equivalent $2nm$ origin vector. Between successive frames the state parameters will change. The corresponding changes in the curve points are given by:

$$\Delta\mathbf{p} = (M + \Delta M)GL(\Delta\mathbf{b}) + (\Delta M)G(L\mathbf{b} + \overline{\mathbf{w}}) + \Delta\mathbf{o} \tag{3}$$

where Δb is the change in parameter b etc. The measurement process is described by the equation:

$$\Delta \mathbf{p}_{\text{observed}} = \Delta \mathbf{p} + \mathbf{v}_k$$

with $\mathbf{v}_k \sim N(\mathbf{0}, R_k)$ where R_k is the measurement covariance matrix.

5 Tracking Filter

The measurement model described by equations (2 and 3) is essentially non-linear (due to the dependence of M on a_x and a_y). Furthermore as the parameters are to be filtered independently it is necessary to decouple the shape, alignment and translation effects. The problem is simplified 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 using a Kalman filter

Alternatively, if the rotation, scaling and translation effects are sufficiently small, all the parameters may be updated simultaneously, treating the coupling effects as measurement noise. A full extended Kalman filter could be used although this would be computationally expensive (see discussion).

5.1 The Discrete Kalman Filter

The standard discrete Kalman filter may be used to update state (and covariance) estimates of a system with discrete measurements (see for example, Gelb [6]). For a standard measurement model,

$$\mathbf{z}_k = H_k \mathbf{x}_k + \mathbf{v}_k$$

the state estimate update is given by:

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

where $\mathbf{v}_k \sim N(0, R_k)$ and

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

and the covariance matrix P_k update is given by:

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

In this case, the model parameters $\mathbf{x}_k$ are filtered independently. This is achieved by ignoring off-diagonal elements of the system covariance matrix P_k. Hence P_k^{-1} is assumed to be diagonal and the covariance update equation becomes:-

$$[\sigma_i(+)]^{-1} = [\sigma_i(-)]^{-1} + {r_i}^{-1} \tag{4}$$

where ${r_i}^{-1} = (H_k^T R_k^{-1} H_k)_{i,i}$ and $\sigma_i = (P_k)_{i,i}$. Writing $H_{j,i} = \left[\mathbf{h}^{(i)}\right]_j$ we obtain:

$${r_i}^{-1} = \sum_{s,t} \mathbf{h}_s^{(i)} \mathbf{h}_t^{(i)} (R_k^{-1})_{s,t} \tag{5}$$

Similar results hold if sets of model parameters are assumed independent (i.e. the covariance matrix P is block diagonal) in which case σ_i is replaced by the i'th block matrix. In effect each parameter is filtered by assuming the other parameters are fixed at their current estimates.

5.2 Updating the Origin

The simplified measurement model for the origin filter (assuming the other parameters are fixed) is given by:

$$\begin{aligned} p_i' &= p_i - (M(\hat{a}_x, \hat{a}_y)G(L\hat{\mathbf{b}} + \overline{\mathbf{w}}))_{2i} \\ p_i' &= o_x + (\mathbf{v}_k)_{2i} \end{aligned}$$

The estimates of the state parameters $o_x, \dot{o}_x$ and the associated 2x2 covariance matrix P_{o_x} are updated by the appropriate Kalman filter equations. For example:

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

where $r^{-1} = \sum_{i=0}^{nm-1} (R_k^{-1})_{2i,2i}$. A similar filter is used for o_y.

5.3 Alignment Filter

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

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

where H is the $2nm$ x 2 measurement matrix defined by:

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

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

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 independent although for simplicity the system noise is assumed isotropic.

5.4 Shape Filter

The remaining parameters describing shape are filtered independently and simultaneously using the simplified measurement model given by:

$$\begin{aligned} \mathbf{p}' &= \mathbf{p} - \hat{\mathbf{o}} - MG(\overline{\mathbf{w}}) \\ \mathbf{p}' &= (MGL)\mathbf{b} \end{aligned}$$

The measurement matrix is thus given by $H = M(\hat{a}_x, \hat{a}_y)GL$. For each shape parameter the associated variance σ_i is updated by the equations 4 and 5. The shape estimates are updated using:

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

where $\Delta\mathbf{p}' = \mathbf{p}' - H\hat{\mathbf{b}}(-)$.

5.5 Global Shape Constraint

In order to ensure that the contour shape is feasible, the shape parameters are constrained to lie within a hyper-ellipsoid as described by Cootes [11]. The shape parameter estimates are updated independently using the Kalman filter equations and then appropriately constrained using:

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

The global constraint stabilises the system and prevents problems such as the contour becoming tangled (due to sections of the contour locking on to the wrong feature).

5.6 The 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,

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

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 [14] 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:

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

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

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

5.7 Automatic Control of Scale

The Kalman filter provides a mechanism for automatically setting the search scale (as demonstrated by Blake et al. [9]). 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:

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

where V(...) denotes the variance operator and $H = MGL$ (and similarly for $V(q_i)$). For simplicity, the alignment matrix M 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:

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

up to some maximum scale proportional to the height of the estimated contour.

6 Implementation Techniques

Once the state parameters have been updated the current estimate of the contour must be computed. An efficient mechanism for doing this is to map to the spline representation and then to rotate and scale the spline control points (i.e. use equation 1). The sampled curve points can then be calculated using the sparse matrix G. Moreover the sampled normals $\mathbf{n_i}$ are calculated directly from the spline control points via a (sparse) linear mapping (see for example Bartels et. al. [15]).

The $2nm$ x l matrix $[GL]$ can be precomputed. The shape measurement matrix H need not be explicitly calculated. To update the inverse shape covariance matrix the quantities r_i^{-1} are directly computed using:

$$\begin{aligned} r_i^{-1} &= [H^T R^{-1} H]_{i,i} \\ &= \textstyle\sum_{j=0}^{nm-1} (h_{2j,i})^2 C_{0,0}^{(j)} + 2h_{2j+1,i} h_{2j,i} C_{0,1}^{(j)} + (h_{2j+1,i})^2 C_{1,1}^{(j)} \end{aligned}$$

where $h_{r,s} = [GL]_{r,s}$ and the 2 x 2 matrices $C^{(j)}$ are given by:

$$C^{(j)} = \mathcal{M}^T A_j \mathcal{M}$$

7 Refinement

Once the state estimates are updated, the contour may still be inaccurate, for instance, if the change in shape between frames is too large for the filtering process. In order to overcome this an iterative procedure can be used by simply taking new observations at the updated scale and using the updated contour estimate.

After several iterations the state covariance matrix values may have become unrealistically small. In order to prevent this occurring the ϵ method is used (see Gelb [6]). The Kalman gain is modified by the addition of an "overweight" gain proportional to the parameter ϵ.

8 Experimental Results

Various image sequences of a person walking in several directions were used. The sequences were captured at 30 frames per second. The shape changes between frames was considerable with a complete walk "cycle" taking typically 20 or so frames. In each case the initial contour position and scale were calculated by a segmentation scheme using background subtraction (see earlier work [13]). The shape parameters were initialised to zero (i.e. the mean shape). Figures 2 and 3 show the system successfully tracking a walking pedestrian using 1 iteration (i.e. no refinement) and 3 iterations respectively.

Fig. 4 shows only the search window obtained when tracking a second walk sequence. Fig. 5 shows the search window obtained when the contour is iteratively refined on a fixed frame with and without overweighting. In the first case there is no overweighting and in the second case the Kalman gain is overweighted with $\epsilon = 0.5$.

In each case the same shape model was used with 14 shape parameters and $s_{max} = 14$. The system has been implemented on an R4000 Silicon Graphics Indigo and can run at 30 frames per second using 80 contour points and with 3 iterations per frame. The images were taken from stored "movie" files. The results show the system can successfully track a walking pedestrian using just one iteration processing over 40 frames per second in such a case.

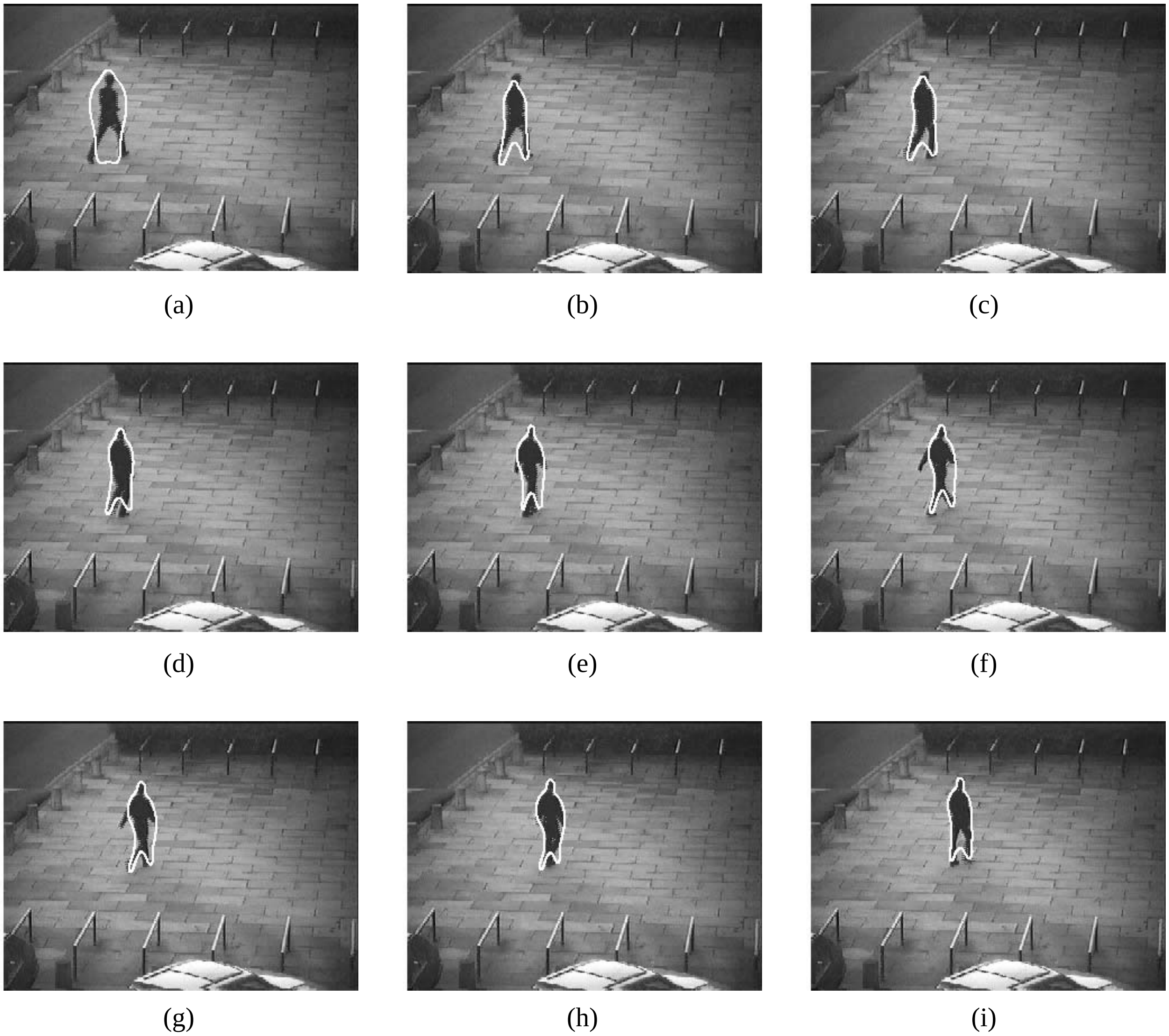
9 Conclusion / Discussion

An efficient method for tracking a non-rigid moving object using a trained contour model has been described. The tracker is object specific utilising a specific shape model based on a training set. However, the tracker will only work properly for poses and views that are sufficiently well represented in the training set. The knowledge incorporated in the model helps the tracker mechanism to respond quickly to large shape deformations and restricts the contour to a feasible shape. Iterative refinement improves the contour estimate after only a few iterations. The statistical filter reduces the number of iterations that would be required by allowing the contour to react rapidly when the shape estimate is uncertain.

The B-spline contour representation allows the efficient computation of the estimated contour points and their associated normals. Moreover the number of sample points can easily be increased (by increasing m) to improve accuracy on larger images whilst utilising the same model.

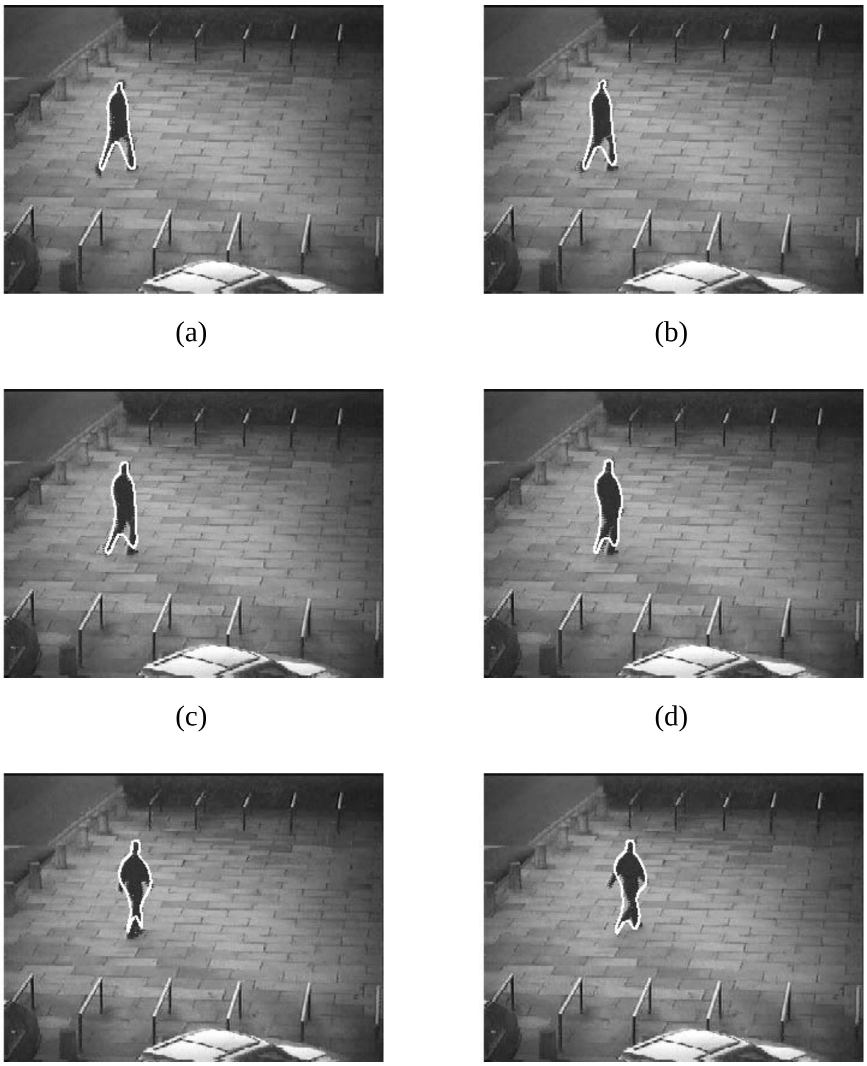
The computational complexity of the tracker is $O(l)$ where l is the number of shape parameters, requiring $O(lmn)$ operations where mn is the number of sampled curve points used. In comparison, the full (extended) Kalman Filter would require $O(l^3mn)$ floating-point operations.

Future research will look into improving the stability and speed of the tracker to allow the system to cope with more complex scenes with several pedestrians and to model a typical walk sequence. Further work is required to investigate how well the method works for other (possibly rigid) objects.



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 2: Tracking without refinement: (a) initial estimate, (b) to (i) frames 0, 2, 4, 8, 12, 16, 20 and 24



(a) (b) (c) (d) (e) (f)

Figure 3: Tracking with refinement: (a) to (f) frames 0, 1, 2, 4 , 8 and 16

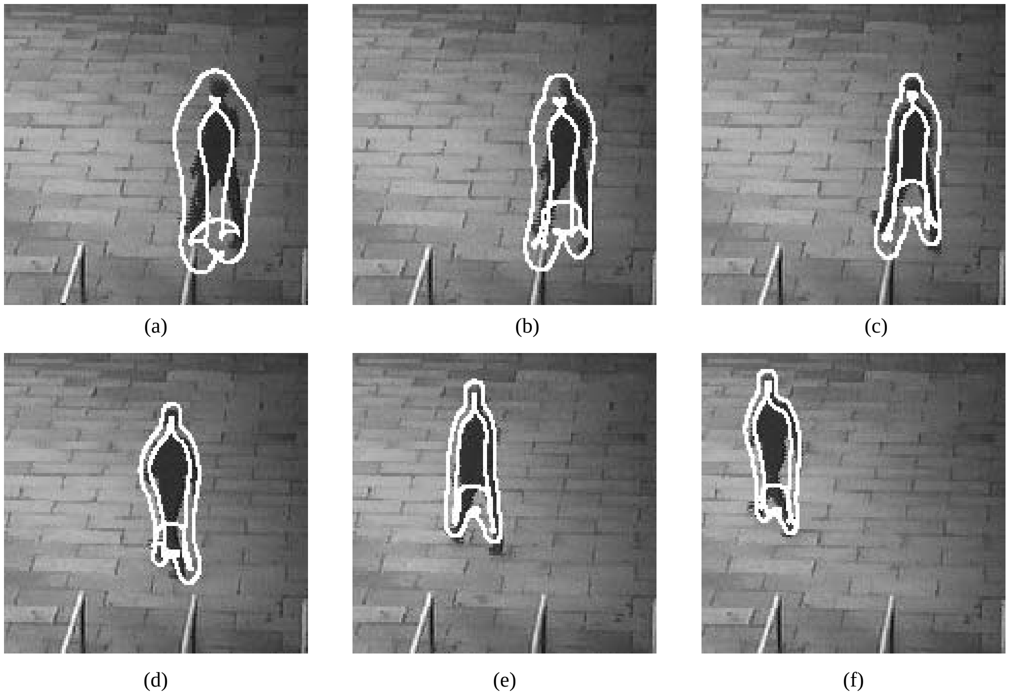


Figure 4: Search window: (a) to (f) frames 0, 1, 2, 16, 32 and 50

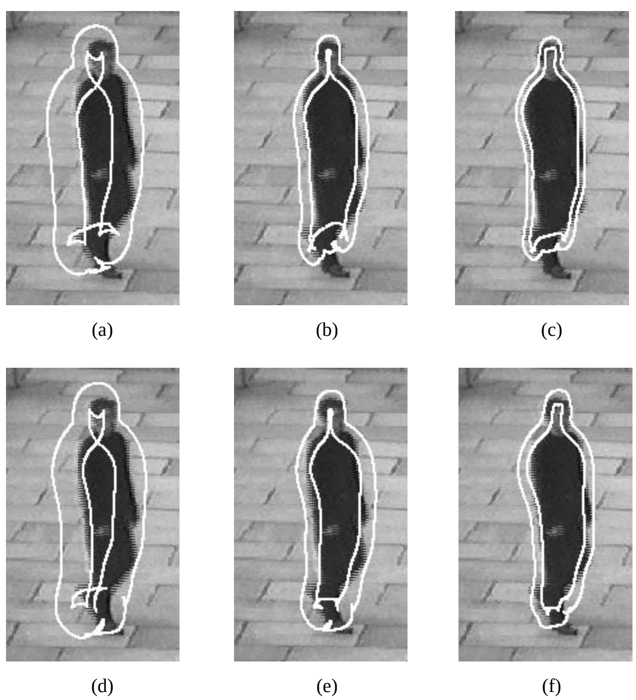


Figure 5: Iterative refinement: (a) to (c) $\epsilon = 0$, (d) to (f) $\epsilon = 0.5$

References

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

[2] Cootes T.F., Hill A., Taylor C.J., and Haslam J. 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.

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

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

[5] Leymarie F. and Levine M.D. 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.

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

[7] Maybeck P.S. *Stochastic Models, Estimation and Control*. Academic Press Inc., 1979.

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

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

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

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

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

[13] Baumberg A. and Hogg. D. Learning flexible models from image sequences. In *European Conference on Computer Vision*, May 1994.

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

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