ACTIVE CONTOURS FOR BEST FIT INTERPOLATION OF OBJECT TRACKS FOR MOTION PICTURE POST PRODUCTION

M. Sinclair, P. Hillman, J. Hannah, D. Renshaw
University of Edinburgh School of Engineering and Electronics

Abstract Tracking is an essential process required for many techniques used in special effects and post production. Standard trackers frequently produce poor results which are difficult to edit by hand. These trackers normally use only previous frames to find the target position in each frame even though the whole sequence is available off-line. This paper introduces active tracking contours which can find the motion of the target by fitting to the whole sequence simultaneously, rather than frame–by–frame. This gives the contour a high degree of resilience against noise, occlusions and false matches. Since these contours are represented as spline curves they are simple and intuitive for an operator to edit. The performance of active tracking contours is compared to standard tracking algorithms.

1 INTRODUCTION

Object tracking is often required for motion picture special effects and post production. It is used for techniques such as replacement, image stabilisation and matchmoving. The Block Matching Algorithm (BMA) [1] is the most common approach for tracking. Here, a reference block from a previous frame is compared to blocks in the current frame using a measure such as *normalised cross correlation*. The position of the block which has the maximum cross correlation with the reference block is taken to be the new position of the object in the current frame. Such block trackers often lose track. The process can therefore be quite laborious: the human operator must stop the tracker and input the correct position of the block every few frames.

There are three major causes of errors in tracking: where the block is obscured giving poor correlation in the current frame; where the correct match is not the global maximum but only a local maximum; and where small amounts of noise cause the best match to be a few pixels away from the actual position of the object. The latter results in small errors in the track position which result in tracking jitter, while the former both tend to cause total loss of tracking. A Kalman Filter [2] can often be used to smooth tracks, but often fails to prevent loss of tracking caused by multiple maxima or poor correlation.

The BMA is most commonly used in real time applications such as vehicle control or surveillance [3]. In these systems, tracking must be progressive: the object position must be estimated without reference to future frames and any error made in previous estimates cannot be corrected. However, in movie post production, the entire sequence to be tracked is available off-line and it is not necessary to track progressively.

For some techniques in post-production it is desirable to edit the tracked path, either to correct for an error in the tracking or to create a different path for a new object based on the path taken by an existing object. Such editing can be difficult. Since the path is defined by a new point in each frame, every frame must be altered individually.

In this paper, we present a new approach to object tracking for special effects in post production. We fit a contour of best fit to the entire sequence, interpolating between two end points indicated by the user. Since this contour has a small number of control points, it is simple to edit. Contours also have inherent smoothing properties, making them resistant to jitter in tracks, and also robust to areas where there is poor correlation or a false global maximum: the contour will follow peaks in local maxima if these form a suitable path between the end points.

This paper is organised as follows: in the next section, active contours (snakes) and their use to find object boundaries in images is reviewed. Adaptations required for use in fitting object paths to image sequences are explained. The algorithm employed for tracking is then summarised. Results of this block tracking algorithm are finally compared to standard techniques.

To avoid confusion, the term ‘snake’ will be used to refer to a contour which is used to find boundaries in still images and ‘tracking contour’ to refer to the new contour described in this paper and used to find paths of objects in image sequences.

2 ACTIVE CONTOURS

2.1 Snakes to find object boundaries

Snakes were originally proposed by Kass *et al* [4] to model object boundaries in images. Edge detectors often produce many false edges in the image, and there may be parts where there is no detectable edge. A snake is resistant to false and missing edges, since it finds a path of minimal energy along edges.

A snake is driven by two opposing forces: the *internal force*, proportional and normal to the curvature of the snake, which prevents the snake bending too tightly by trying to push the snake into a circle, and the *external force*, derived from the image edge map, which acts to push the snake towards the edges in the image. Typically, this force is proportional to the derivative of the image edge map, so the snake is pushed towards peaks in the edge image, which correspond to edges in the original image.

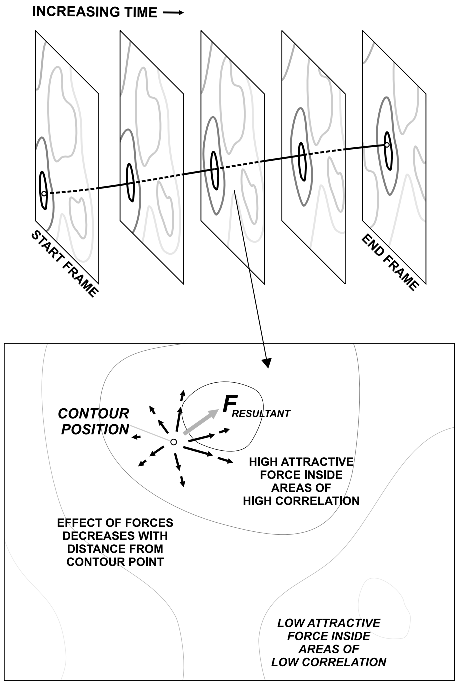


Figure 1: Schematic of an active tracking contour

Typically, these two are forces are assumed to act on control points, which are connected by straight lines (such as in the implementation by Lobregt and Viergever [5]) or by a smooth spline. The control points are considered as masses upon which the forces act.

The algorithm is iterative: the snake is initialised to lie close to the edges in the image, either by hand or by some alternative technique. In each iteration, the external and internal forces acting on each control point are calculated from the edge image and from the local curvature of the snake through the control point respectively. Following classical Newtonian mechanics, the force acting on the snake causes it to accelerate towards a new position.

Since the snake can change size, contour points are added or removed to keep the length of the sections between them within set limits. The process repeats until the snake is at rest.

Other information, apart from edges, can be used as the external force field. Malpica *et al* [6] used optical flow as an external energy field to cause the snake to settle along the boundary of differently moving objects. Zhou *et al* [7] derived the external energy field from colour and texture information.

2.2 Adaptation to tracking contours

Our tracking contours are slightly different from snakes: In a tracking contour, the start and end points in the first and last frames, respectively, are fixed by the user, so only the intermediate position of the contour can change. The contour is defined by control points which move in space but maintain even spacing in time (*eg* if there are 20 frames between the start and the end and there are 8 control points then there will always be a control point every 2.5 time steps).

Fig 1 shows a schematic of a tracking contour. The contour begins and ends at positions set by the operator and passes through each intermediate frame. The start and end positions are also used to extract reference blocks for block matching. The tracking contour is moved by the application of forces within each frame, created by good matches (high correlation) to the reference block. The strength of the force decreases with distance between the current contour position and the area of good match. Areas of poor match (low correlation) also have weaker forces than good correlation.

The contour is not always pulled towards the global maximum correlation in each frame. If this area is too far away it will instead by pulled towards a local maximum. This unique property gives the tracking contour resilience against areas of false high matches.

2.2.1 The external force

Whereas snakes are pulled towards edges in images, tracking contours are pulled towards areas which have a high correlation with the reference block. Normalised Cross Correlation (Phase Correlation) is used to compare blocks in the image. Given a reference block R and a test block T, normalised cross correlation c is given by

$$c = \frac{1}{2} + \frac{\sum_{ij}\left((R_{ij} - \overline{R}) \cdot (T_{ij} - \overline{T})\right)}{2\sqrt{\sum_{ij}\left(R_{ij} - \overline{R}\right)^2 \cdot \sum_{ij}\left(T_{ij} - \overline{T}\right)^2}} \qquad (1)$$

where the ij term indexes each pixel within the block and $\overline{R}$ and $\overline{T}$ indicate the mean intensity of the reference block and test block respectively. Cross correlation values usually lie on the range $(-1, 1)$. The division by 2 and addition of $\frac{1}{2}$ produces a value which lies on the range $(0, 1)$

The external force acting at a particular point at position $\boldsymbol{p}$ in a given frame n of an image is calculated as follows: Each pixel at position $\boldsymbol{t}$ within a given distance of $\boldsymbol{p}$ is considered to exert a force $\boldsymbol{f}$ on $\boldsymbol{p}$, given by

$$\boldsymbol{f} = \frac{(\boldsymbol{p} - \boldsymbol{t}) \cdot c}{\|\boldsymbol{p} - \boldsymbol{t}\|^{\alpha}} \qquad (2)$$

where α is some constant greater than 1 and c is the normalised cross correlation between the block centred at point $\boldsymbol{t}$ and the nearest reference block. That is $\boldsymbol{f}$ acts in the direction $\overrightarrow{\boldsymbol{tp}}$ and is attenuated according to distance and according to the magnitude of the match. Thus, matches which are near the curve have more influence than those that are farther away.

The total force acting on $\boldsymbol{p}$ is the sum of all forces from nearby pixels. Since finding the cross correlation values in computationally intensive they are cached for reuse in subsequent iterations.

Two minor modifications are made to ensure better results. We have found that large areas of medium correlation tend to dominate over small areas of high correlation, causing the resultant force to act away from maxima and towards more average values. This can be prevented by ignoring all pixels which have a correlation lower than a set value (typically about 0.7). Also, pixels are ignored where the correlation value is lower than that given for the block based around point $\boldsymbol{p}$: Thus, the force cannot act to pull $\boldsymbol{p}$ towards a worse match than its current position.

2.2.2 The internal force

Snakes have large numbers of control points upon which the external forces act. Using large numbers of control points in tracking contours would make them harder to edit and make the contour less smooth. It is important that the position of the control points (and hence the position of the contour) be influenced by every frame in the sequence, not just those nearby the control points. Thus, some algorithm for distributing the forces acting upon tracking contour in each frame of the sequence to the control points is required. This is achieved while implicitly implementing the internal forces on the contour. By distributing the force from the frame to every control point, smoothness is maintained. The distribution is weighted so that control points closer in the sequence to the frame receive more force than those further away in time.

Let t_n be the time index for frame n, t_{c_k} be the time index for control point k. Let m be the maximum difference between the time index of a control point c and t_n. The force f_{k_n} distributed to control point c_k from frame n is given by

$$f_{k_n} = u f_n \left(1 + m - |t_n - t_{c_k}|\right)^{\beta} \qquad (3)$$

where β controls the rate of fall-off (and therefore the magnitude of the implicit internal force) and u is a weighting factor given by

$$u = \left(\sum_h \left(1 + m - |t_n - t_{c_h}|\right)^{\beta} \right)^{-1} \qquad (4)$$

for each control point h on the contour.

3 ALGORITHM

The algorithm for implementing the tracking contour is shown in Fig. 3. Note there is a force included proportional to the velocity of the snake. This is a damping factor which ensures that the contour eventually comes to rest if there are no forces acting upon it, and prevents it oscillating. Also, the forces are not calculated in one step but in multiple passes. The reason for this is illustrated in Fig. 4. In this case there is an area of false matches close to the initial position of the contour, shown as a dotted line. If forces from the entire contour are considered

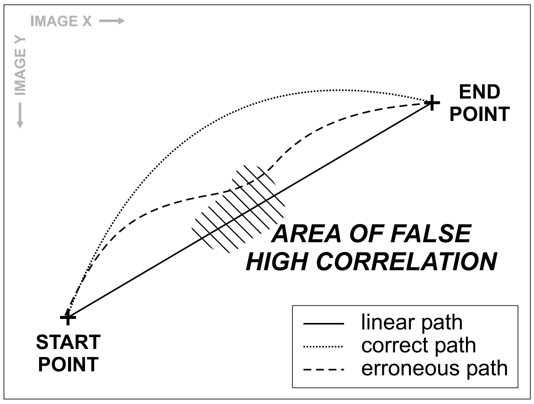


Figure 4: Where the entire contour is settled simultaneously, the contour may be attracted towards areas of false high matches in the centre of the contour (the time axis is not shown in this figure)

from the beginning, the contour will be attracted towards this area even though it is some distance from the true position of the contour; the correct matches will be too far from the position of the contour for it to be attracted towards them. Since the position of the contour is known at the end points, frames close to these points are used first. Subsequently more points are used to refine the contour position. This ensures that the contour is always close to points being considered. By the time the position of the contour in the middle of the sequence is considered, it will have moved away from the area of false matches.

Fig. 2 shows this process in operation. Initially, the contour forms a straight line between the end points. In the first iteration, only the first and last frames are in the active set and so only these two frames act on the contour. As time progresses, more frames are included in the active set until all frames are acting on the contour.

In the case that the final path of the contour is close to the initial linear path, this process is not necessary and the contour can be resolved in a single step for speed.

4 RESULTS

The performance of our contour tracker has been compared to common method of block tracking. In all cases, the reference block is never updated: the block taken from the initial frame marked by the user is used for the entire sequence. a 16 $\times$ 16 block was used for all sequences.

4.1 Standard BMA

A standard Block Matching Algorithm was run on the data: The first frame was used as a reference block, and the position of the best match in each subsequent frame found using the Normalised Cross Correlation technique (equation 1). Searching takes place in a search aperture

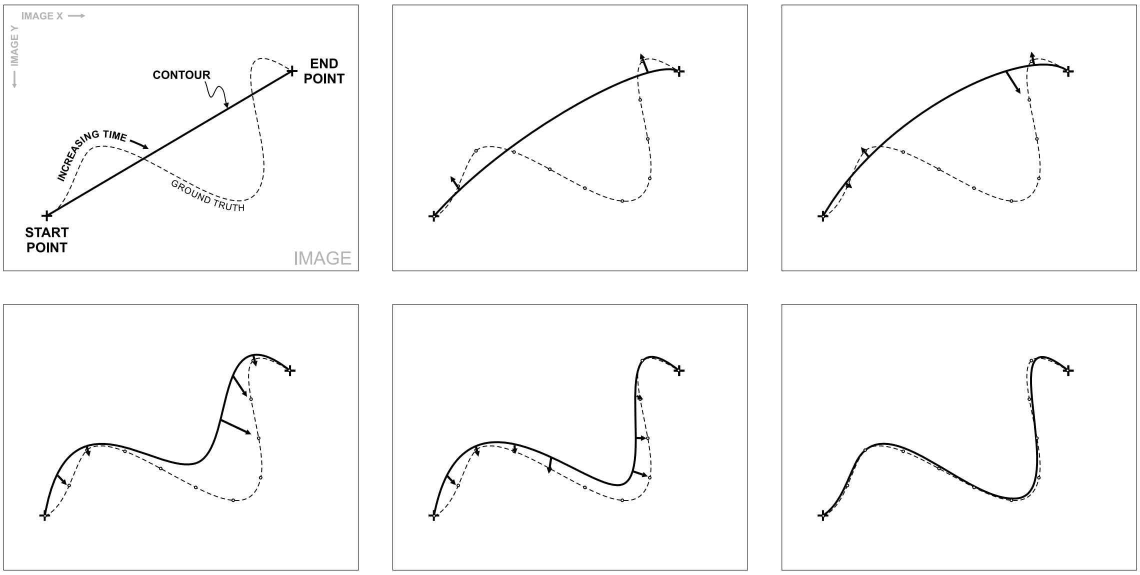


Figure 2: Progressively more and more frames provide forces (shown as arrows) which cause the contour to move

1. User indicates start S_{ij} and end E_{ij} positions of the snake in the start frame s and end frame e
2. Extract reference blocks around points S_{ij} and E_{ij} from the start and end frames
3. Initialise the snake as a straight line between S_{ij} and E_{ij} with n evenly spaced control points
4. Set of active frames = $\emptyset$
5. Set of inactive frames = all intermediate frames
6. While there are frames in inactive set
   (a) Remove first and last frames from inactive set and add to active set
   (b) For several hundred iterations or until snake settles:
      i. For each frame in active set
         A. Let P_n be the position of contour in frame n
         B. Find $\boldsymbol{f_n}$, the force acting on point P_n (see section 2.2.1)
         C. Distribute force $\boldsymbol{f_n}$ across all control points (see section 2.2.2)
      ii. For each control point k
         A. Let $\boldsymbol{T_k}$ be $\boldsymbol{f_k} - \rho \boldsymbol{v_k}$ where $\boldsymbol{v_k}$ is the velocity of of control point k and ρ is a constant
         B. New velocity of contour point $\boldsymbol{v_k} \leftarrow \boldsymbol{v_k} + \boldsymbol{T_k}/m$ where m is a constant
         C. New position of contour point $\boldsymbol{c_k} \leftarrow \boldsymbol{c_k} + \boldsymbol{v_k}$

Figure 3: Algorithm for contour tracking

centred around the match in the previous frame. The output from the tracking is the position of the best match in each frame.

Since our tracking contour has two fixed points rather than only one from the initial frame, the BMA was run twice: once using the initial frame as a reference block and progressing forwards through the sequence, and again using the final frame as a reference block and progressing backwards through the sequence.

4.2 Kalman tracker

Tracking with Kalman filtering increases smoothness and provides more robust tracking than the standard BMA approach. For comparison purposes, a Kalman filter was used with a simple linear velocity model and no cross terms (*i.e.* no dependency between the the x and y ordinates of the position, velocity and error estimates). In each frame, the position of the best match (again using the Normalised Cross Correlation technique) is found. The Kalman filter then predicts the target position in the subsequent frame and the search begins about this location. Using the Kalman estimated position for the target rather than the position of the best match helps to prevent the tracking losing the target - the correct position of the target is more likely to remain within the search aperture even if the tracker drifts slightly. The output from the tracking is the position component of the corrected Kalman state vector.

Again, tracking took place in both forward and reverse directions.

4.3 Curve of best fit

One of the advantages of our tracking contour is the ease of editing of the position, simply by moving the control points of the resultant curve. Instead of using a tracking contour, it would be possible to obtain a best fit curve to the tracked results. Therefore, the performance of the tracking contour has been compared to that of a curve fitted to the Kalman filtered output.

The curve of best fit is approximated as follows: Initially, a curve is fit linearly between the first and last points found by the Kalman filter. The curve has the same number of control points as was used with the tracking contour. Again, the control points maintain a fixed position in time and can move in (x, y) only.

To fit the curve, each control point is moved to the position which minimises the total curve error. The error in each frame is found by finding the Euclidean distance between position of the contour and the position of the Kalman filtered output. The total curve error is the error in each frame summed over all frames. The process iterates until no control point is moved between iterations.

4.4 Results

We tested our algorithm on two sequences [8], both of which contain smooth curved motion.

4.4.1 The *carpark* sequence

The *carpark* sequence (720×576, 25fps interlaced, 100 frames, 256 greys) was captured with a security camera, videotaped onto analogue tape and then digitised. Figs. 5(e) and 5(f) show the position of the hand marked blocks in the start and end frames respectively: these are the fixed points for the contour fit and the frames from which the reference blocks are taken.

The tracking contour fit to the car is shown in Fig. 6, which shows four frames of the sequence superimposed with the contour drawn on top. Clearly, the contour has correctly tracked the motion and does not deviate from the true motion of the car.

4.4.2 The *cyclist* sequence

The *cyclist* sequence (720×576, 25fps interlaced, 155 frames, 256 greys) was captured with a digital video camera. Fig. 7 show key frames of the sequence. The hand marked positions for the start and end frame are shown in Figs. 7(e) and 7(f). The tracking contour fit to this data is shown in Fig. 8. Here, there is a minor deviation which occurs when the cyclist passes the large vehicle, but the motion is correctly tracked.

4.5 Numerical results

Since we have used real sequences as test data, ground truth has been estimated by hand tracking each intermediate frame of the sequence.

The difference in each frame between our reference ground truth and the position estimated by each of the trackers is shown in Fig. 9 for the *carpark* and *cyclist* sequences. The backward BMA and Kalman trackers can be seen to lose track in the *cyclist* sequence. The tracking contour for the *cyclist* sequence appears to match the ground truth particularly closely.

Table 1 compares the performance of all the algorithms to the ground truth. The Root Mean Squared error was used as a measure: The Euclidean distance between the ground truth position and the position estimated by the technique is measured. The RMSE is given by the root of the average of the squared Euclidean distance.

Our technique shows a clear improvement over Kalman filtered tracking, and, in the case of the *cyclist* sequence, proves to be the most accurate method for tracking altogether.

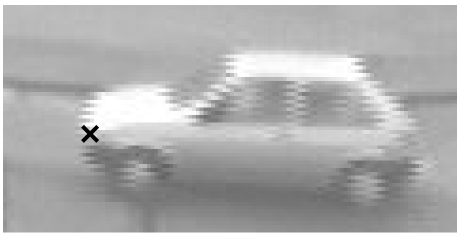
Since we are attempting to fit a curve to the underlying data, the comparison with the track obtained by fitting a

(a) Start frame: frame 1825

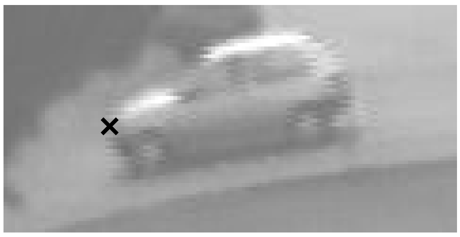
(b) Frame 1850

(c) Frame 1875

(d) Final tracking frame: frame 1925



(e) Detail showing hand-marked position in frame 1825



(f) Detail showing hand-marked position in frame 1925

Figure 5: The *carpark* sequence

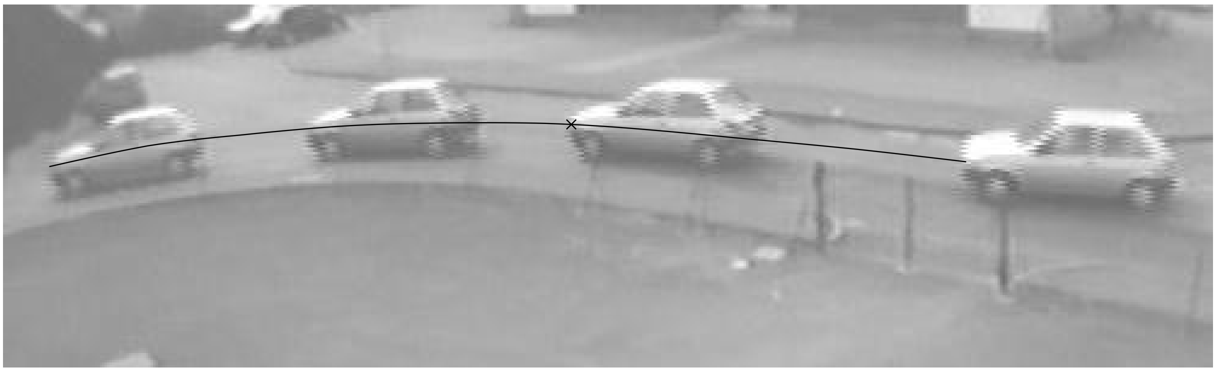


Figure 6: Active contour fit to *carpark* sequence shown over composite image of multiple frames in the sequence. The position of the contour in frame 1865 is marked with a cross

curve to the Kalman data is most interesting: our active contour fits the actual motion much better than a curve fitted to any other form of data.

4.6 Discussion

The errors presented in table 1 show that our contour is marginally better as a tracker than Kalman tracking and is superior to curve fitting to a Kalman track. However, the reference track used to compare these results was estimated by eye. There is therefore some error in this reference which will affect the results. Additionally, there are problems with the *carpark* sequence caused by interlacing and synchronisation problems. The resulting jitter in the sequence is smoothed in the Kalman, contour and curve fits, but neither in the standard BMA algorithm nor in the reference 'ground truth.' This is undoubtedly why the standard BMA tracker appears to perform so well on this sequence. The appearance of spikes in the error plots (Fig. 9) for all the techniques except for the standard BMA suggest the presence of this jitter. The apparent error is caused by smoothing across the jitter.

It can be seen from Fig. 9 that only the tracking contour accurately fits to both ends of the data. All other trackers drift off as they track, causing an increasingly large error. This is particularly clear in the tracks for the forward Kalman filter. Since the ends of tracking contours are fixed, they ensure that the track will not diverge off at either end. As expected, the point of greatest divergence for the contour is towards the centre of the sequence.

Figure 10 illustrates an example from the *cyclist* sequence where the active contour can perform better than any "best match" case. This figure shows a surface plot of correlation matches to the reference block in a small area of the image. Here, the global maximum is marked with a vertical cyan line in the centre of the image. This is some distance away from the correct match, shown in red, which is close to a second peak in the correlation surface. The tracking contour, shown in blue, has been attracted towards this lower peak by previous and subsequent frames.

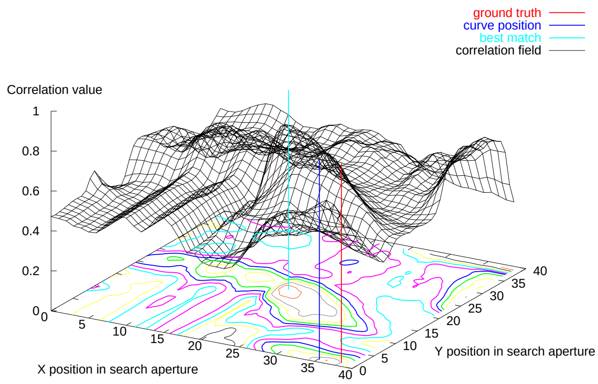


Figure 10: When the true match is not the global maximum but a lower local maximum, the contour can locate the correct position

Therefore the contour lies closer to the ground truth than the best match. This confirms that tracking contours are able to track where the best match may not be at the global maximum but at a local maximum some distance away.

5 CONCLUSIONS AND FUTURE WORK

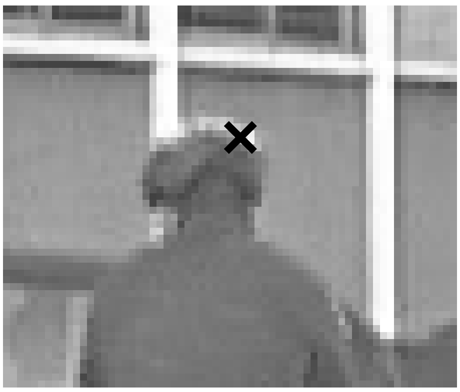
This paper has presented a new active contour model for tracking the path of objects in off-line sequences. The contour is represented as a spline curve which can be easily edited to correct for errors or to modify the path of the object. The active contour approach produces a significantly better fit than fitting a spline to a standard system such as a Kalman tracker. Also, the success of the Kalman filter is highly dependent on the selection of the noise parameters, which is less intuitive than selecting the parameters for an active contour. Furthermore, the tracking contour appears robust to changes in the parameter values, removing the need for fine parameter adjustments. Tracking contours appear to be a valuable method for hand-assisted

(a) Start frame: frame 0

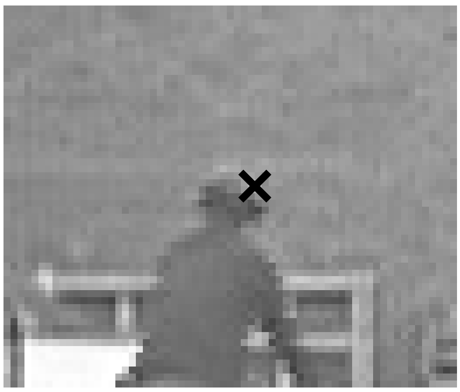
(b) Frame 50

(c) Frame 100

(d) End frame: Frame 154



(e) Detail showing hand-marked position in frame 0



(f) Detail showing hand-marked position in frame 154

Figure 7: The *cyclist* sequence

	Standard BMA		**Kalman Filter**		**Curve fit to**	**Active**
	Forwards	**Backwards**	**Forwards**	**Backwards**	**best Kalman**	**Tracking contour**
Carpark	2.225	1.8294	2.6620	2.6409	2.6176	2.4639
Cyclist	1.5984	(Track fails)	1.5411	(Track fails)	1.5393	1.3251

Table 1: Comparative performance of different algorithms: RMSe compared to ground truth

Figure 8: Active tracking contour fit to *cyclist* sequence shown over composite image of multiple frames in the sequence. Position of the contour in frame 120 is marked with a cross

off-line tracking, and an attractive tool for special effects and post production.

Future work will investigate the possibility of using varied reference blocks, perhaps by mixing the initial and the final reference block.

ACKNOWLEDGEMENTS

Peter Hillman was supported by EPSRC grant ref: GR/506578/01. Thanks to Peter Grecian and Piotr Stanczyk of the Moving Picture Company for helpful discussions.

References

[1] R. Gonzales and R. Woods, *Digital Image Processing*. Addison–Wesley Publishing, 1993.

[2] M. Efe, D. Atherton, and J. Bather, “Adaptive kalman filters for manoeuvering target tracking,” in *IEE Colloquium on Target Tracking and Data Fusion*, pp. 4/1–4/7, 9 June 1998.

[3] C. Haworth and D. Renshaw, “Image–sequence motion estimation for the dervish mine-clearance vehicle,” *IEE Proceedings on Science, Measurement and Technology*, vol. 148, pp. 89–94, May 2001.

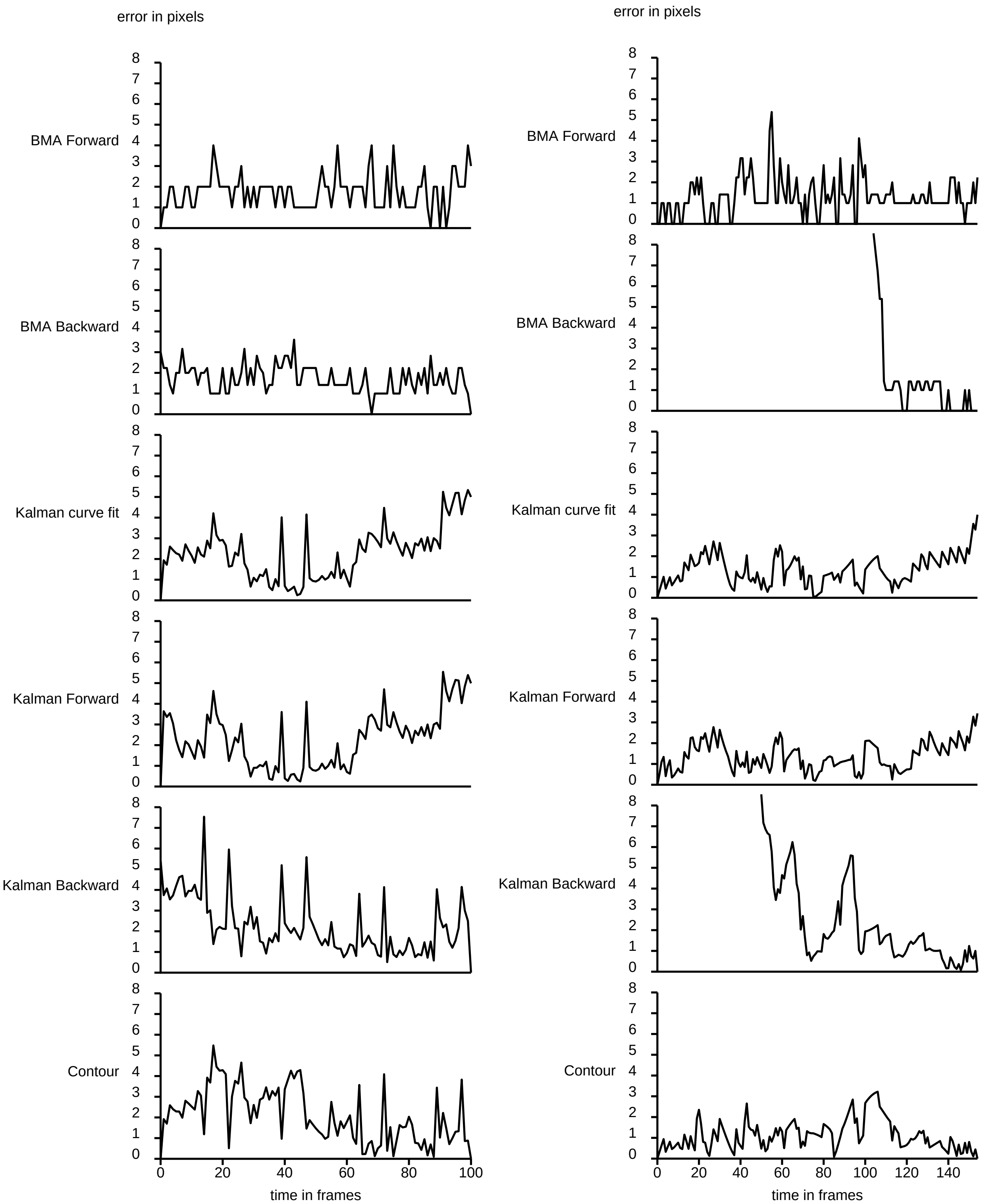
[4] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” *International Journal of Computer Vision*, vol. 1, pp. 321–331, 1998.

[5] S. Lobregt and M. A. Viergever, “A discrete dynamic contour model,” *IEEE transactions on Medical Images*, vol. 14, pp. 12–24, March 1995.

[6] N. Malpica, A. Santos, E. Perez, M. Garcia-Fernandez, and M.Desco, “A snake model for anatomic m-mode tracking in echocardiograhy,” in *Proceedings of the 3rd International Symposium on Image and Signal Processing and Analysis*, vol. 2, pp. 722–726, 18-20 September 2003.

[7] Q. Zhou, L. Ma, D. Chelberg, and M. Celerik, “Color and texture priors in active contours for model-based image segmentation,” in *Proceedings of the 3rd International Symposium on Image and Signal Processing and Analysis*, vol. 2, pp. 690–695, 18-20 September 2003.

[8] C.Haworth, A. Peacock, and D.Renshaw, “Performance of reference block updating techniques when tracking with the block matching algorithm,” in *Proceedings of the International Conference on Image Processing*, vol. 1, pp. 365–368, October 2001.



(a) The *carpark* sequence

(b) The *cyclist* sequence

Figure 9: Graph of error (Euclidean distance between ground truth and tracked position) against time of each tracking system