Fragment Based Tracking for Scale and Orientation Adaptation

V. Srikrishnan, T Nagaraj, Subhasis Chaudhuri
Electrical Engineering Department
V.I.P Lab
Indian Institute of Technology Bombay, Mumbai-400076, India
krishnan@ee.iitb.ac.in

Abstract

In this work, we propose a simple yet highly effective algorithm for tracking a target through significant scale and orientation change. We divide the target into a number of fragments and tracking of the whole target is achieved by coordinated tracking of the individual fragments. We use the mean shift algorithm to move the individual fragments to the nearest minima, though any other method like integral histograms could also be used. In contrast to the other fragment based approaches, which fix the relative positions of fragments within the target, we permit the fragments to move freely within certain bounds. Furthermore, we use a constant velocity Kalman filter for two purposes. Firstly, Kalman filter achieves robust tracking because of usage of a motion model. Secondly, to maintain coherence amongst the fragments, we use a coupled state transition model for the Kalman filter. Using the proposed tracking algorithm, we have experimented on several videos consisting of several hundred frames length each and obtained excellent results.

1 Introduction

Visual tracking remains as one of the most challenging research areas in computer vision. Fast object motion, changing appearance, clutter and occlusion makes tracking an object in a video a very difficult task. However, considerable progress has been made in this area. An exhaustive survey and taxonomy of tracking methods can be found in [19]. Tracking approaches can be roughly categorised into blob, feature and contour based methods. Contour based methods[11] are useful for getting the complete boundary of the object but are much slower than blob based approaches. For many applications, complete shape information is not required and hence a simple geometric shape like a rectangle or an ellipse enclosing the object is sufficient. Our work is based on the blob tracking approach.

Initially, blob tracking methods used variants of a simple sum of square differences(SSD) measure to minimise the difference of pixel values between the initialised object and some patches, which was usually a small window centred on the previous location of the target[15]. To make the tracker robust to clutter and improve tracking speed, motion models and probabilistic appearance models were introduced[12]. Using simple histograms and the Parzen window method to model the target appearance, Meer *et al.* proposed the mean shift based tracker[8]. This tracker, popularly called the mean shift tracker, is very simple to implement and can handle relatively fast motion. Moreover, the tracker is very fast and realtime tracking of several targets is possible. Integral histograms[16] have also been used for tracking. Some recent works in tracking have explored the possiblity of using detection and trajectory estimation methods to perform tracking[14]. Tracking by detection approaches suffer from the limitation that they are too specific for the target and a good amount of offline training is needed for generating a good set of features. Moreover, online updation of the features is also a problem.

There are two main problems associated with the blob based approach for tracking. First, there is the problem of defining a simple yet discriminative appearance model. The histogram based appearance models are simple, but they suffer from the problem of lack of spatial description of the target appearance. This makes them quite susceptible to clutter. Some recent works which tackle this problem are [3][5]. Another major problem with the blob based approaches is to adapt scale and orientation of the bounding box or ellipse to the constantly changing size of the target in real life videos. In this work, we address this particular problem of scale and orientation change. Specifically, the key contribution of this work is the use of a fragment based tracking approach within a Kalman filtering framework to handle scale and orientation changes.

The outline of the rest of the paper is as follows. In section 2, we discuss the different methods proposed by researchers to solve this problem. The proposed method is de-

scribed in section 3. Results and discussions are presented in section 4. Conclusion and scope for future work is presented in section 5.

2 Related work

As we have mentioned earlier, work on tracking can be classified into contour, blob and feature tracking. It is out of scope of this work to review the entire literature. As excellent and recent review of the different tracking algorithms is given in [19]. For the blob tracking technique, the mean shift based tracker[8] proposed by Meer *et al.* has become extremely popular among the researchers. The reason for its popularity is its ease of implementation and speed. Real time tracking of objects is possible. Also, researchers have used the mean shift tracking algorithm within the particle filtering framework to achieve robustness, for example[9]. However, the original mean shift formulation had no way of handling orientation. To adapt to scale, at each frame the tracking algorithm was run multiple times at 10% scale change. As has been reported in [7], such a procedure led to rapid shrinking of the tracker. Therefore, using the mean shift framework and making it adaptive to scale and orientation without sacrificing its speed has been the major thrust area of researchers. In [20], the authors have used a modified version of colour correlograms to model the appearance of the target. By this, they are able to track object rotations and small changes in scale. This however is not very useful in handling the kind of scale changes which occur in practical situations as exemplified in figure 5. In [7], the idea is to perform a search within a range of the ellipse scale using scale space. Orientation handling is not defined which restricts the application of the work. Zivkovic and Krose[21] present an extension of the original mean shift algorithm which computes the local mode as well as the covariance matrix around it. They have used this to define a 5 degrees of freedom tracker for scale and orientation change. In [19], the author has proposed a tracking algorithm based on asymmetric mean shift procedure. Though both of these works track scale and orientation changes succesfully, there is another problem associated with blob tracking. Both methods use the histogram of the target as initialised in the first frame. Such an appearance model is extremely susceptible to clutter and not robust against occlusion. Moreover, the usage of a histogram to model the target loses all information regarding the spatial distribution of colours.

Adam *et al.*[3] have proposed an interesting approach using fragments to track people in presence of partial occlusions. The algorithm is quite simple and is as follows. The target is divided into a number of overlapping fragments. The position of each fragment with respect to the centre of the target is fixed and known. Tracking is carried out by finding for each fragment, the best match within a local neighbourhood. The similarity measure of each fragment is ranked and converted into a voting scheme to find the target centre. Therefore, even if some of the fragments are lost due to occlusion, the target is located by using the unoccluded fragments. One limitation of this approach is scale change is handled in the same heuristic manner as in [8]. Moreover, orientation change cannot be handled in this framework. Another interesting parts based approach toward tracking is [17]. In this work, the authors propose a framework for tracking human by decomposing the human body into three sections and fitting an ellipse for each part. They then use the particle filtering framework to ensure robust tracking in presence of occlusion and clutter. Scale change is not handled however. In a related work, the authors in [13] have used a mean shift fragment based approach. Similar to [3], they are able to handle partial occlusion. There is no provision for scale or orientation change within this framework.

As mentioned earlier, we use a fragment based approach in this work. The reason for using fragments is that they capture the spatial distribution of the object's appearance. Though in the use of fragments, our work is similar to [3], we have a crucial difference with their approach. In constrast to their work where the fragment positions are fixed, we permit relatively free movement of fragments. This enables us to capture scale and orientation change much more effectively.

3 Proposed Method

We use a fragment based approach to describe the appearance of the target. The key idea in this work is to use the positional information of the constituent fragments to infer the scale and orientation of the target. The outline of the algorithm is given in algorithm 1. The basic idea is to divide the target into a number of fragments, track the individual fragments and finally combine back the fragments to get the target localisation. We now describe the different steps of the proposed algorithm in detail.

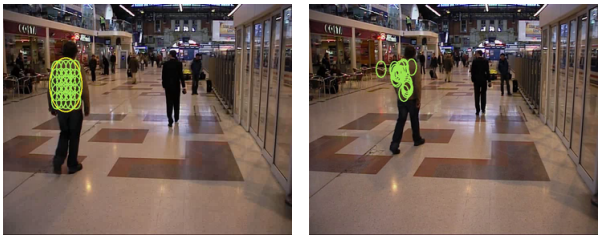
3.1 Initialisation and Appearance Modelling

We assume that the initialisation of the target is done in the first frame, either manually or by some detection technique. The target is then subdivided into a number of overlapping fragments. The number and size of the fragments play an important part in the performance of the tracker. In our work, we set the width and height of each fragment to be approximate 35% of the target's width and height respectively. Each fragment centre is shifted from its neightbour centre by half the fragment width in horizontal direction and half the fragment height in vertical direction, respectively.

Algorithm 1 Algorithm for tracking scale and orientation change

Require: $\mathbf{X}_0$, W_T and H_T

Set $W_F = 0.35 * W_T$, $H_F = 0.35 * H_T$. Set $M = 2{*}W_T/W_F{-}1$, $N = 2{*}H_T/H_F{-}1$, $Frags = M{*}N$. Initialise 2 constant velocity Kalman filters.
Calculate $\mathbf{x}_i$, $i = 1 \ldots Frags$
while $frames \leq END$ **do**
  $\mathbf{y}_i$ = MeanShift($\mathbf{x}_i$), $i = 1 \ldots Frags$
  Use $\mathbf{y}_i$ as measurement for Kalman filter
  Predict and update $\mathbf{x}_i$ using Kalman filter.
  for each centre **do**
    if centre lies within 10% times previous localisation **then**
      add to list L of centres for calculating moments
    end if
  end for
  find image moments using centres in L
  find major axis, minor axis and angle using first and second central image moments
  if if magnitude of scale change exceeds 25% of current scale **then**
    reinitialise all fragments
  end if
  align all fragments along the estimated angle
end while



(a) Frame 1 (b) Frame 25

Figure 1. (a)Figure showing target and multiple fragments initialisation as the first step of the algorithm. (b) A single fragment to the extreme left has drifted off. This fragment is an outlier and should not be considered for target localisation.

The justification of these choices is given in the next section. An example of target and fragment initialisation is shown in figure 1(a). In this figure, which is illustrative of the initialisation scheme, the fragment size is half the target dimensions, and with the shifts as described earlier we have a total of 9 fragments.

The appearance of each fragment is modelled by a weighted histogram as in [8]. The weights are obtained by using a kernel which gives more importance to the pixels located at the centre of the fragment as compared to the pixels at the boundary. Following [3], we could also use simple unweighted histograms for target representation. The choice of representation governs the tracking algorithm for the individual fragments. It is to be noted that the proposed approach does not depend on the representation of the individual fragments. Therefore, either of these two methods can be used for representation. For our choice of appearance model, the tracking algorithm used is the well known mean shift algorithm[8]. The centre of the i-th fragment is denoted by $\mathbf{x}_i = (x_i, y_i)$. The target centre is denoted by $\mathbf{X}$. Target and fragment dimensions are denoted by (W_T, H_T) and (W_F, H_F), respectively.

3.2 Fragment Tracking

The tracking algorithm consists of two modules. The first module consists of a number of mean shift based trackers for moving the individual fragments. The second module consisting of the Kalman filter is responsible for preventing fragments from drifting off.

Mean Shift Tracking

We use the mean shift based tracking algorithm[8] for tracking the individual fragments of the target. For the sake of completion, we briefly describe the model. For the complete details, the reader is referred to [8]. Let the photometric feature of interest be u. For our case, we use the RGB colour space. Assuming that the initialisation is placed at the origin, the model histogram consisting of M bins in constructed as

$$p_M(u) = C \sum k\left(||\frac{\mathbf{x}_i}{h}||^2 \right) \delta[I(\mathbf{x}_i) - u], u = 1 \ldots M \quad (1)$$

where I is the image, C is some normalising constant and the function $k(.)$ is the kernel function with certain properties. We have used the Epanechnikov kernel. The model histogram p_M is defined in a similar fashion. Tracking is done by using the convergence in the previous frame as the initialisation in the current frame and minimising the Bhattacharya distance between the two histograms.

The mean shift tracker is widely used by vision researchers for its speed. Speed is an important criterion for

tracking as there are multiple fragments within the target. One disadvantage of the mean shift tracker in its original form is its vulnerability to clutter. As has been discussed earlier, histograms are unable to capture the spatial information within the target. Therefore, the mean shift tracker, which uses weighted histograms, is easily distracted by clutter. For our case. this problem is especially serious for the fragments lying on the object boundary. During initialisation, such boundary fragments learn a part of the background as well as the actual target. During tracking in subsequent frames, these boundary fragments sometimes drift off the target completely. This is shown in figure 1(b). The fragment to the extreme left has drifted off the target completely. This is potentially disastrous for target localisation. Though we perform outlier detection of the different fragments before estimating the position and orientation of the target, it is desirable to prevent such a problem from occuring. This provides the motivation for the use of a Kalman filter[4].

Kalman Filter Dynamics

Specifically, we use Kalman filter for two purposes. Firstly, the Kalman framework uses a motion model which adds robustness to the tracker. Kalman filter was also used within the mean shift framework in the original work[8]. Secondly and more importantly for our purpose, the Kalman filter can also be used to maintain a degree of coherence among the different fragments.

We use a constant velocity Kalman filter model. We assume that the motion in each of the x and y directions is independent. Therefore, we have two Kalman filters which have the same structure. Therefore, without loss of generality, for the constant velocity model, the state vector $\mathbf{X}$ is defined as

$$\mathbf{X} = \{x_1, x_2, \ldots, x_N, \dot{x}_2, \dot{x}_2, \ldots, \dot{x}_N\} \tag{2}$$

Here we have assumed that there are N fragments constituting the target. The state transition model is

$$\mathbf{X}_{t+1} = \mathbf{F}\mathbf{X}_t + \mathbf{w}_t \tag{3}$$

where $\mathbf{F}$ is the state transition matrix and $\mathbf{w}_t$ is the randomness in the motion. The random vector $\mathbf{w}_t$ is assumed to follow a Gaussian distribution with zero mean and covariance matrix $\mathbf{Q}_t$. The state transition matrix $\mathbf{F}$ and the noise covariance matrix $\mathbf{Q}$ play an important part in maintaining fragment coherence. By fragment coherence, we mean that the fragment locations and their motions are not completely independent of each other. This assumption is quite intuitive because we expect the fragments to have some spatial proximity to each other, even though their motions are otherwise completely unconstrained. We impose this constraint by making $\mathbf{F}$ and $\mathbf{Q}$ to be non diagonal, where we have dropped the subscript for the noise matrix $\mathbf{Q}_t$. The diagonal elements of $\mathbf{F}$ are given a weight $0 < w \leq 1$ and the non diagonal elements are given a weight $(1-w)/(N-1)$. Generally the weight w is kept close to 1, we have used a value of 0.95. The reason for this is that for smaller values of w, the fragments would perform a kind of cyclic pursuit motion[18] which is not desirable. Therefore, the current state of a particular element depends not only its state at the previous time, but the neighbours also have a small but definite influence. Similarly, the covariance matrix $\mathbf{Q}$ is also non-diagonal. For $\mathbf{Q}$ however, the non diagonal elements are made slightly larger in value compared to the diagonal elements. The reason for making non-diagonal elements larger than diagonal elements is that we are less sure about the position of a given element of the state vector given its neighbours, but more certain given the previous position. We have used the concept of coherence for the first N elements of the state vector, that is for the positional information only. This could be extended to include the speed too.

Figure 2. Figure showing outlier elimination. The fragments in red are the outliers. The fragments in green are used for localising the target. The box in blue decides the outliers.

Following the standard practice in Kalman filtering, we use only the positional information as the measurement. The locations of the fragments after the mean shift iterations constitute the measurements. The measurement model is

$$\mathbf{Y}_t = \mathbf{X}_t + \mathbf{v}_t \tag{4}$$

where the true state vector is corrupted by additive white Gaussian noise $\mathbf{v}_t$ with zero mean and covariance $\mathbf{R}$. In the next subsection, we discuss the process of obtaining scale and orientation from a given set of fragments.

3.3 Outlier Elimination and Target Localisation

Despite the use of the Kalman filter to maintain the spatial proximity of the fragments as discussed in the previous

section, it is not possible to prevent drift of some of the fragments. Therefore, the first step in localisation would be the removal of the outlying fragments.

We use a simple method for detecting and removing the outlier fragments. This is demonstrated in figure 2. This is a frame extracted from the sequence shown in figure 6. The fragments marked red are the outliers and are neglected. Given the centre and orientation of the target in the previous frame, we discard those fragments which lie outside 10% of the previous localisation. Furthermore, we also discard those fragments whose similarity measure with the corresponding model histogram is below a certain threshold T. The similarity measure used is the Bhattacharya distance. Keeping in mind that Bhattacharya distance has a maximum and minimum value of 1 and -1 respectively, we have set $T = 0.5$. After the removal of outliers, we use a blobs-based approach to find the best ellipse fit to the target. For in-lying fragments, we find the elliptical blob and coalesce all such blobs to get a target blob. For this combined blob, we find the centralised image moments. Using these moments, we find the centroid, which gives the centre of gravity and the covariance matrix, which is interpreted as the spread of the blob. The centroid is taken to be the centre of the target and the scale and orientation are obtained using the moments and the covariance matrix. The fragments shown in green are considered for taking image moments. The search space is shown in blue.

As an alternative approach, once can use an ellipse fitting approach[10]. First, we find four points $P_1^i = (x_i + h_x, x_i + h_y), P_2^i = (x_i - h_x, x_i + h_y), P_3^i = (x_i + h_x, x_i - h_y)$ and $P_4^i = (x_i - h_x, x_i - h_y)$, where the superscript denotes the fragment index. We then fit an ellipse to all such points P_j^i. This gives the scale and orientation of the target for the current frame. However, we found that the image based approach gives the better fit to the target. However, we found the moments based approach to be more effective in fitting the target dimensions.

4 Results and Discussions

We now present the implementation details and show the results on a number of challenging sequences. The fragment dimension is chosen to be approximately 30-35% of the dimension of the target. The target is manually initialised in the first frame of each sequence. Each fragment is shifted by half its width from the fragments to its side and by half the height to the fragments above or below it. For the Kalman filter, we used a weight value of $w = 0.95$. In our implementation, we had a provision for re-initialising fragments if the number of fragments classified as outliers exceeded a half of the total number of fragments. However, this reinitialisation was never needed to be done in any of the experiments. Reinitialisation of fragments is carried out if the scale either increases or decreases by 25% of the current scale.

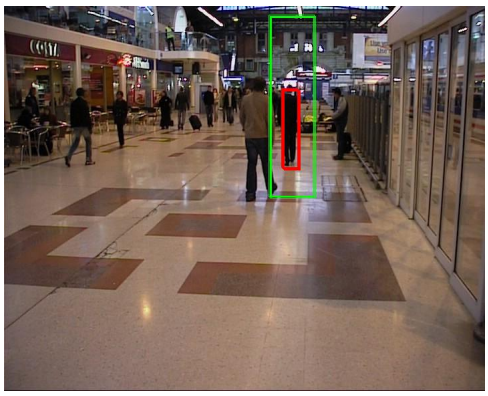


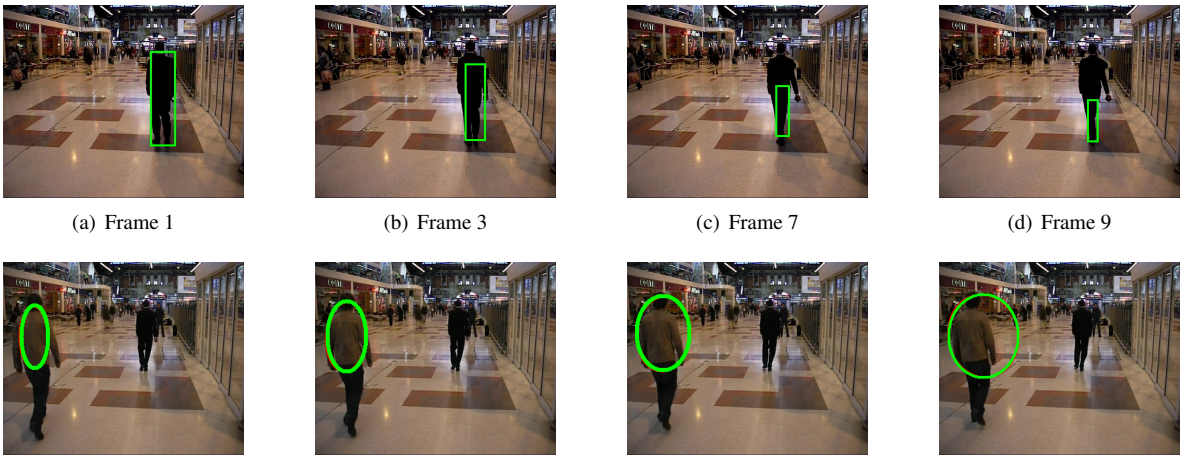
Figure 3. Failure of the tracking method of [3] without scale handling.

In figure 4(a-d), we show the failure of the original mean shift based tracker[8]. The sequence has been extracted from the PETS 2006 database[1] from frame number 1470 to 1610. The sequence consists of a person walking rapidly away from the camera. For adapting the tracker to scale, we used the same strategy as in [3][8]. For each frame, we ran three mean shift trackers, one at the same scale as the previous frame and one each at 20% increase and decrease in scale. We see that the tracker shrinks rapidly to a very small box and track is completely lost. The same problem has been reported in [7]. Figure 3 shows the result of fragment based approach with fixed position of fragments within the target.

In figure 5, we show the results obtained using the proposed tracker. We see that the tracker is completely able to track the target through the drastic scale change.

Figure 6 shows the results of tracking using the proposed algorithm on another sequence taken from the PETS 2006 database. For comparison purposes, in figure 4(e-f) we show the results of camshift algorithm[6] on the same sequence . We have used the implementation provided in the OpenCV[2] library. This sequence clearly shows the failure of the camshift algorithm. We tried different settings of the parameters but could not get acceptable tracking of the shown target.

In the previous sequences, the target to be tracked was undergoing scale changes only. In figure 8, we present the results on a tracking sequence where the target undergoes orientation change. This sequence is of 150 frames in length. Note that the object is tilted by 90 degrees and then comes back to its original orientation. We are succesfully able to track the orientation change. Another sequence showing tracking through orientation change is shown in



(a) Frame 1 (b) Frame 3 (c) Frame 7 (d) Frame 9

(e) Frame 1 (f) Frame 2 (g) Frame 3 (h) Frame 4

Figure 4. Failure of mean shift and camshift[6] trackers. (a-d)Mean shift tracker fails with 10% scale adaptation at each frame. (e-f) Camshift also fails to maintain scale of the target. Note that both the algorithms fail very rapidly.

figure 7.

5 Conclusion and Future Work

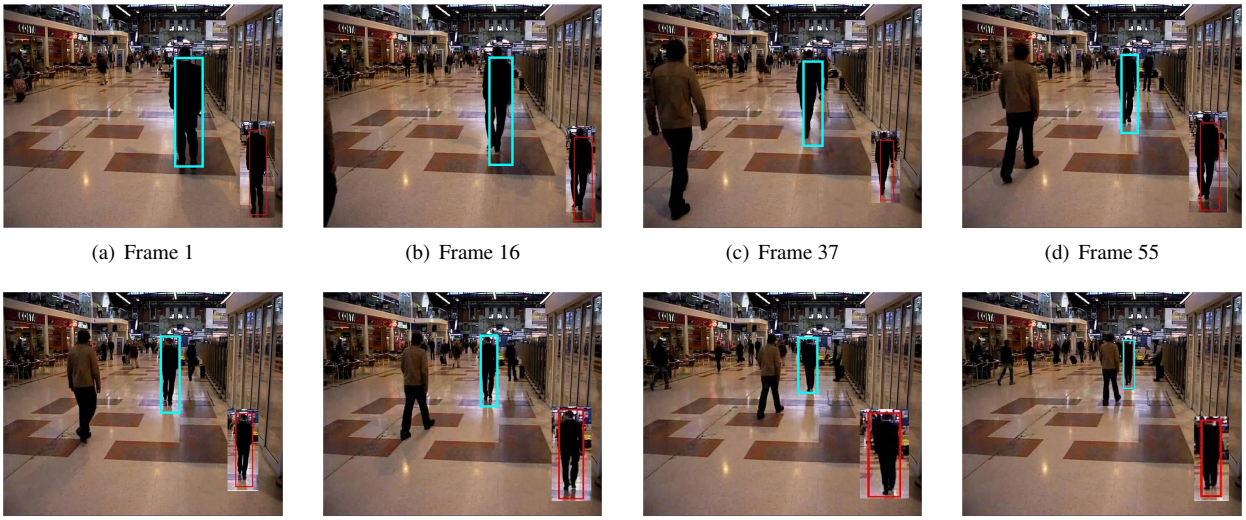
In this work, we have proposed a simple yet effective fragment based approach for tracking objects in video through rapid scale and orientation change. One limitation of the proposed algorithm is the inability to handle partial occlusion. In [3], the authors have proposed a fragment based method for tracking object through partial occlusion. However, as mentioned earlier, it should be noted that the authors assume fixed locations of the fragments with respect to the centre of the target. This immediately precludes the possibility of tracking the target's orientation change. We plan to address the issue of handling partial occlusion within the current framework in a future work.

Acknowledgments

Funding from J.C. Bose National fellowship is gratefully acknowledged.

References

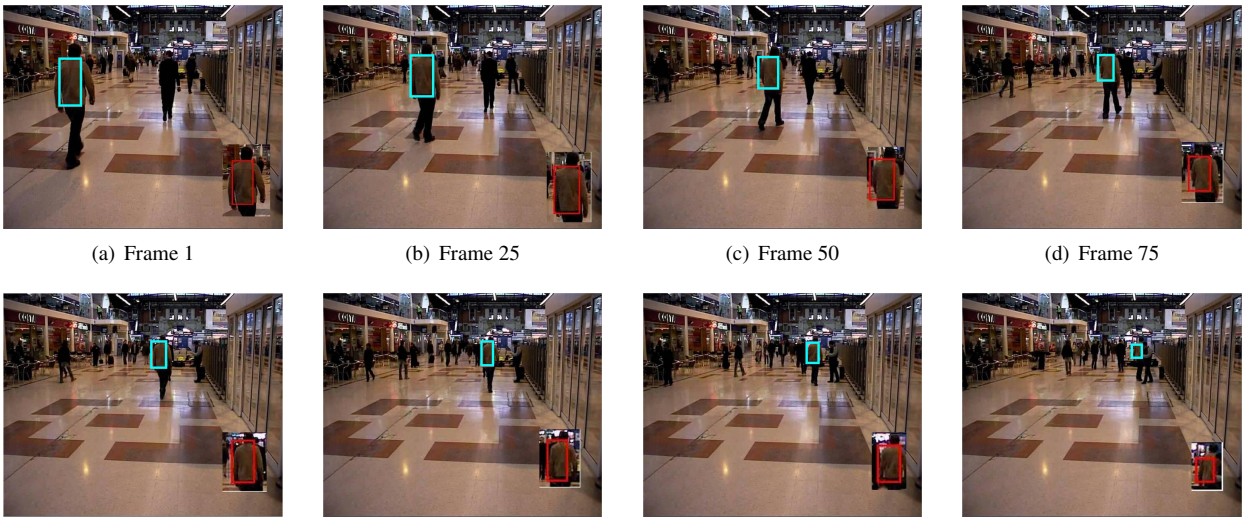
[1] `http://www.pets2006.net/`.
[2] `http://sourceforge.net/projects/opencvlibrary/`.
[3] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *CVPR '06*, pages 798–805, 2006.
[4] Y. Bar-Shalom and X.-R. Li. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., New York, NY, USA, 2001.
[5] S. T. Birchfield and S. Rangarajan. Spatiograms versus histograms for region-based tracking. In *CVPR '05*, pages 1158–1163, 2005.
[6] G. Bradski. Computer video face tracking for use in perceptual user interface. In *Inter Technology Journal*, 1998.
[7] R. T. Collins. Mean-shift blob tracking through scale space. In *IEEE CVPR*, volume 2, pages 234–240, 2003.
[8] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. PAMI*, 25(5):564–575, 2003.
[9] K. Deguchi, O. Kawanaka, and T. Okatani. Object tracking by mean shift of regional color distribution combined with the particle-filter algorithm. In *ICPR*, 2004.
[10] A. Fitzgibbon, M. Pilu, and R. B. Fisher. Direct least square fitting of ellipses. *IEEE Trans. PAMI*, 21(5):476–480, 1999.
[11] D. Freedman and Zhang. Active contours for tracking distributions. *IEEE Trans. Image Proc*, 13(4):518–527, April 2004.
[12] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi. Robust online appearance models for visual tracking. *IEEE Trans. on PAMI*, 25(10):1296–1311, 2003.
[13] J. Jeyakar, R. V. Babu, and K. R. Ramakrishnan. Robust object tracking using local kernels and background information. In *ICIP*, 2007.
[14] B. Leibe, K. Schindler, and L. V. Gool. Coupled detection and trajectory estimation for multi-object tracking. In *IEEE CVPR*, 2008.
[15] K. Nickels and S. Hutchinson. Estimating uncertainty in ssd-based feature tracking. *Image and Vision Computing*, 20:47–58, 2002.
[16] F. Porikli. Integral histogram: A fast way to extract histograms in cartesian spaces. In *CVPR '05*, pages 829–836, 2005.
[17] J. Satake and T. Shakunaga. Multiple target tracking by appearance-based condensation tracker using structure information. In *ICPR*, 2004.
[18] A. Sinha and D. Ghose. Brief paper: Generalization of nonlinear cyclic pursuit. *Automatica*, 43(11):1954–1960, 2007.
[19] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4):13, 2006.
[20] Q. Zhao and H. Tao. Object tracking through color correlogram. In *PETS*, 2005.
[21] Z. Zivkovic and B. Krose. An em-like algorithm for color-histogram-based object tracking. In *IEEE CVPR*, 2004.



(a) Frame 1 (b) Frame 16 (c) Frame 37 (d) Frame 55

(e) Frame 73 (f) Frame 82 (g) Frame 112 (h) Frame 138

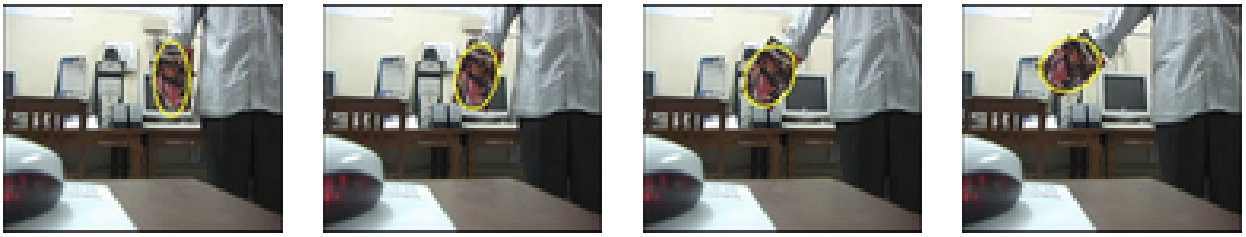
Figure 5. Result of tracking using the proposed algorithm on a sequence extracted from the PETS 2006 sequence. Note that despite the drastic change in scale, target localisation is maintained. The first frame is offset from the PETS sequence by 1469 frames.



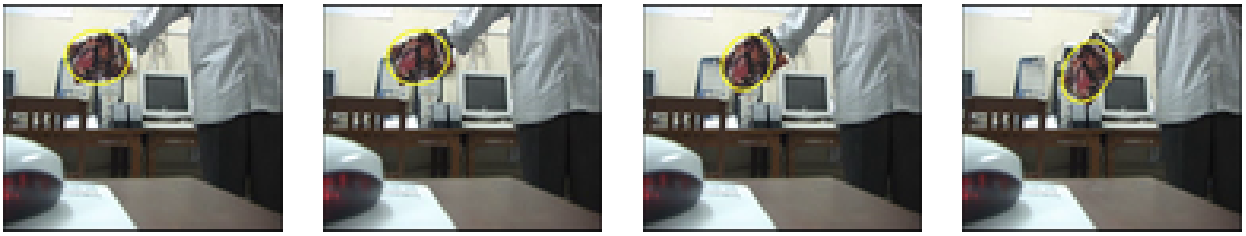
(a) Frame 1 (b) Frame 25 (c) Frame 50 (d) Frame 75

(e) Frame 95 (f) Frame 120 (g) Frame 150 (h) Frame 174

Figure 6. Result of tracking using the proposed algorithm on a second sequence extracted from the PETS 2006. Here the target size undergoes a rapid decrease.



(a) Frame 1 (b) Frame 20 (c) Frame 40 (d) Frame 60



(e) Frame 80 (f) Frame 100 (g) Frame 120 (h) Frame 140

Figure 7. Tracking results for the book sequence. There is a significant orientation change of the book, as also can be seen from the orientation of the ellipse.

(a) Frame 1 (b) Frame 5 (c) Frame 8 (d) Frame 15

(e) Frame 21 (f) Frame 30 (g) Frame 37 (h) Frame 42

Figure 8. Tracking results for the person sequence. The full sequence is of 400 frames length. The bending action is repeated throughout. We show one complete cycle only.