

# A Fragment Based Scale Adaptive Tracker with Partial Occlusion Handling

Nikhil Naik  
College of Engineering, Pune.  
Pune-411005,India  
naiknd06.extc@coep.ac.in

Sanmay Patil  
College of Engineering, Pune.  
Pune-411005,India  
patilsg06.extc@coep.ac.in

Madhuri Joshi  
College of Engineering, Pune.  
Pune-411005,India  
maj.extc@coep.ac.in

**Abstract**— We propose a fragment based algorithm for efficient target tracking under significant scale variation and partial occlusion. In contrast, none of the previous multiple part or fragment based algorithms are both scale adaptive and robust to partial occlusion. In our algorithm, the target is divided into a number of overlapping image fragments. Their color histograms are compared with those of candidate fragments within the neighborhood in subsequent frames. The candidate fragments with maximum histogram similarity with the template fragments contribute towards selection of location of best match within that neighborhood. Its position is further calibrated by using a localization method. We implement a systematic scale adaptive tracking scheme which is robust to significant changes in the target size. Extensive experimental results based on real life sequences and test datasets are presented which demonstrate the excellent tracking accuracy achieved by our algorithm at real time.

## I. INTRODUCTION

Human motion tracking is an active area of research in computer vision. Various challenges in this area include change in appearance, rapid object motion, changes in illumination, occlusion and clutter. It also has a wide range of applications including smart surveillance; perceptual user interface, activity analysis for security and commercial purposes, content based image storage and retrieval, video conferencing, classification and recognition from motion etc.

The various tracking algorithms can be roughly divided into 4 categories [1] - region based tracking, active contour based tracking, feature based tracking and blob based tracking. In this paper, we focus on blob based tracking algorithm. It is much faster than contour based algorithms and is especially useful when it is sufficient to identify the target by a simple rectangle or ellipse rather than with an exact shape.

In our paper, we use a fragment based tracking approach in which the target is divided into a number of overlapping image fragments. These fragments are then used for selection of a best match from candidate fragments in subsequent frames. It is robust to *both* partial occlusion and variations in scale, in contrast to all the previous trackers which employ multiple fragments or parts for tracking, to the best of our knowledge.

## A. Previous Work

The blob based tracking model suffers from following problems. The histogram based models used to represent the object are vulnerable to clutter and occlusions due to the non-spatial nature of the data. Also in real life scenarios, the target undergoes significant variations in scale. The blob based methods fall short in adapting to these changes.

A popular approach used to overcome these difficulties has been the use of multiple parts or fragments of the target for tracking. In [2], a multi-part representation is used to track ice-hockey players, dividing the rectangular box which bounds the target into two non-overlapping areas corresponding to the shirt and trousers of each player. A similar three part based approach is used in [8]. These solutions cannot be effectively implemented for a generic target. In [3], the multi-part target model algorithm can handle scale changes. However it is susceptible to partial occlusion and clutter.

One of the important blob based tracking techniques is the mean shift tracking algorithm [4] proposed by Meer et al. It has become popular due to its simple implementation and speed. It has been incorporated in the multiple part framework to overcome its problems like local basin of convergence. In [5], the authors use multiple fragments which are controlled using mean shift and the most reliable fragment is used for tracking. A similar approach is used in [11]. However these approaches do not address the issue of scale variation. Another interesting algorithm is [6] in which multiple fragments are moved to the nearest minima using mean shift. They use constant velocity Kalman filter to maintain coherence among the fragments. Even though this algorithm is robust to scale changes, it cannot handle partial occlusion.

Another important fragment based algorithm is FragTrack[7], proposed by Adam et al. The algorithm uses a fixed structure of overlapping fragments. Tracking is carried out by finding for each fragment, the best match within a local region. The similarity measure of each fragment is ranked and then they minimize a robust statistic to find the target centre. They are able to handle partial occlusion well. However for scale variation, they adopt a heuristic approach of enlarging and shrinking the template by 10%, and choosing the position and scale which give the lowest score. This procedure leads to

a rapid shrinking in the tracker size if the object being tracked is uniform in color.

We develop a similar voting based algorithm. However our tracker differs from FragTrack in following ways. We propose a systematic mathematical scheme for robust adaptation to the target scale changes using the Bhattacharyya histogram similarity metric. We also develop a 'full explanation' scheme (explained in Section IV) which solves the dilemma of partial versus full explanation faced by FragTrack[7].

### B. Our contribution

- We propose a fragment based tracker which is robust to partial occlusion.
- We develop a systematic scheme for scale adaption so that the algorithm can efficiently track a target undergoing significant scale variation.
- We demonstrate the excellent tracking accuracy of our algorithm using real life scenarios and challenging tracking sequences from the CAVIAR test dataset [9].

The rest of the paper is arranged as follows. In Section II and III, we describe the algorithm in detail. In Sections IV and V, the implementation details and results are given. Finally in Section VI, some conclusions are drawn.

## II. FRAGMENT BASED TRACKING

### A. Initialization

It is assumed that the target is initialized in the first frame manually or using some detection method. It is then subdivided into a number of overlapping template fragments. We select rectangular fragments with width 50% and height 25% of the template width and height respectively. Each fragment centre is shifted from its neighbor centre by half the fragment width in horizontal direction and half the fragment height in vertical direction. Now a color histogram of each fragment is calculated and stored as a template. Figure 1(a) depicts the fragment initialization on the target at the beginning of tracking.

### B. Selection of Best Match

After initialization, let  $O(x_{n-1}, y_{n-1})$  be the estimated location of the center of target rectangle  $I_{n-1}$  in the previous frame. Now, in the current frame, a larger rectangle  $T_n$  of both height and width twice that of the target rectangle is defined around  $O(x_{n-1}, y_{n-1})$ .



Figure 1(a). Fragment initialization Template is made up of 21 overlapping fragments.



Figure 1(b). Search neighborhood  $T_n$  (red) with  $I_{n-1}$  in green and candidate fragments in blue. One of the nominees shown in brown



Figure1(c). Selection of the localization circle  $S_n$  in black.  $C_n$  shown in blue and one of the candidates for  $R_n$  in dashed yellow.

It is made up of non-overlapping rectangular fragments of the same size as the original fragments. This rectangle  $T_n$  works as the search area in which the best match to the template is obtained. Figure 1(b) shows how  $T_n$  is selected.

A histogram of each of these candidate fragments is calculated. It is compared with histogram of one of the template fragments and a similarity measure is obtained using the Bhattacharyya similarity measure. (For mathematical discussion on Bhattacharyya measure, please refer to section III.) Now the candidate fragment with the best score i.e. maximum similarity obtains one 'vote' and no vote is given to other fragments. This process is repeated for each template fragment.

After vote distribution is done,  $T_n$  is divided into a number of overlapping rectangles (called 'nominees'), each of the size of  $I_{n-1}$  (see Figure 1(b)). Now the votes for each nominee are calculated by adding number of candidate fragments with votes contained by it. The nominee with highest number of votes, denoted by  $C_n$  is selected for further fine-tuning of location. It has the same size as  $I_{n-1}$

### C. Localization

We observed that that selection of  $C_n$  as the final location of target reduces the tracking accuracy due to the rigid nature of the fragments template. So once  $C_n$  is selected, a search radius  $\gamma$  is identified using the equation-

$$\gamma = 0.25 * \min(C_W, C_H) \quad (1)$$

where  $C_W$  and  $C_H$  represent the width and height of  $C_n$  respectively. A circle  $S_n$  is defined with radius  $\gamma$ , as shown in Figure 1(c). Now a rectangle of the same size as  $C_n$  is selected with center at every pixel on or inside the circumference of  $S_n$  and its histogram is calculated. It is compared with the histogram of whole  $I_{n-1}$  using the Bhattacharyya measure. After the comparisons are complete, a rectangle with the best score i.e. most similar histogram is selected as the best match  $R_n$ .

### D. Scale Adjustments

Let  $R_n$  be the location of target selected from localization in step C. We have the histogram comparison score of  $R_n$  and the template. The Bhattacharyya measure equation used by us gives a score of 0 for best match while 1 represents perfect mismatch. Let  $u_n$  be the score of  $R_n$ . Now we obtain

value  $u$ , the immediate average of histogram comparison scores. It is given by

$$u = (u_n + u_{n-1} + u_{n-2} + u_{n-3})/4 \quad (2)$$

We also simultaneously obtain an average of all the previous histogram comparison scores, given by  $A$

$$A = (u_{n-1} + u_{n-2} + \dots + u_1)/(n-1) \quad (3)$$

Now the following relation is used to decide the size of the final rectangle  $I_n$  with center  $O(x_n, y_n)$

$$I_n = \eta \cdot R_n \quad \text{if } u \geq (A + \epsilon) \quad (4)$$

$$I_n = R_n \quad \text{else}$$

where  $\eta$  is the reduction factor in area and  $\epsilon$  is a small positive constant between 0 and 1. In this way, in every fourth frame, the size of the target rectangle is adjusted using the above relation. When the actual size of target becomes smaller than the tracker rectangle, the value of immediate average  $u$  increases due to inclusion of background information. Therefore it satisfies the inequality in Equation (4) and the area is reduced by a predefined factor of  $\eta$ .

### III. HISTOGRAM SIMILARITY MEASURE

#### A. Calculating color histograms

The task of calculating the color density function of the fragments is formulated as follows. The feature  $v$  represents the color of the candidate fragment. The probability of color of a template fragment is modeled by its histogram  $q_v$ . Let the candidate fragments centered at  $y$  ( $y=1, 2, \dots, N$ ) be modeled by their color histograms  $p_v(y)$ . Now the task is to find the discrete location  $y$  whose associated density  $p_v(y)$ , is the most similar to the template density  $q_v$ .

#### B. Bhattacharyya measure

In order to calculate the similarity measure we first take into account the Bhattacharyya Coefficient, whose discrete form is given by [10],

$$P[p_v(y), q_v] = \sum_{v=1}^M \sqrt{p_v(y) q_v} \quad (5)$$

For example, if two distributions are identical,

$$P[p_v(y), q_v] = \sum_{v=1}^M \sqrt{p_v(y) q_v}$$

$$= \sum_{v=1}^M \sqrt{p_v(y) p_v}$$

$$= 1 \quad (6)$$

The similarity measure is defined as metric distance (see [4]) between the candidate and the template histogram and it is given by:

$$D(y) = \sqrt{1 - P[p_v(y), q_v]} \quad (7)$$

We have employed expression (7) for calculating the similarity measure. Minimizing the distance between two distributions is equivalent to maximizing the similarity. So a lower score means more similarity and the maximum mismatch gives a score of 1.

---

### Algorithm 1: Algorithm for fragment based tracking

---

**Given:**  $x_0, y_0, H, W, \epsilon = 0.05, \eta = 0.95$

- 1: Calculate  $q_0$ , the color histogram of initial template defined by  $x_0, y_0, H, W$
  - 2: Set  $STEPW = 0.5 * W$  &  $STEPH = 0.25 * H$   
Divide given template in overlapping fragments  
**while** end of template **do**  
     $x = x + 0.5 * STEPW, y = y + 0.5 * STEPH$ , Define fragment  
**end while**
  - 3: Calculate  $q_i, i = 1 \dots$ , template fragment
  - 4: **while** frames  $\leq$  END **do**.
  - 5: Define search area of size 400% of initial template with initial template at the center.
  - 6: Divide search area in non-overlapping fragments  
**while** end of search area  
     $x = x + STEPW$  and  $y = y + STEPH$ , Define fragment  
**end while**
  - 7: **for** each template fragment **do**  
    **for** each search fragment **do**  
        calculate  $p_j, j = 1, 2, \dots$ , search fragment.  
         $b = \text{compare}(q_i, p_j)$ .  
    **end for**  
    set vote=1 for the search fragment which has  
     $b = \arg \min b$ .  
**end for**
  - 8: Select nominees i.e. search windows of 8 fragments  
    **for** each search window **do**  
        **if** vote=1 **then** increment the vote count  
    **end if**  
**end for**
  - 9: Determine the probable location  $C_n$  using maximum vote count.
  - 10: Set localization area of radius  $0.25 * \min(C_W, C_H)$  around center of  $C_n$  and define rectangles at each point
  - 11: **while** less than or equal to radius **do**  
    calculate  $q_k, k = 1, 2, \dots$ , radius  
     $u_n = \text{compare}(q_k, q_{I_{n-1}})$   
**end while**
  - 12: The rectangle with minimum  $u_n$  is best match.
  - 13: Set localization of rectangle  $I_n$  at  $O(x_n, y_n)$ . Reinitialize all the fragments
  - 14: Find the immediate average  $u$  and running average  $A$
  - 15: **if**  $u \geq (A + \epsilon)$  **then**  $I_n = \eta \cdot R_n$  **end if**  
    **else**  $I_n = R_n$  **end else**
  - 16: Calculate histogram of  $I_n$  i.e.  $q_{I_n}$
  - 17: **end while**
- 

### IV. IMPLEMENTATION DETAILS

We now present the implementation details. In all experiments we have used 21 overlapping template fragments in initialization and 32 non-overlapping fragments in the search rectangle  $T_n$ . We use the hue component from the HSV color space. 15 bins were used for representing the histograms of individual fragments. For scale reduction, the



Figure 2. The performance of our tracker from a real life outdoor sequence. Tracking accuracy is maintained as the target undergoes rapid changes in scale as it can be seen in the inset from frame 368 onwards.



Figure 3. Demonstration of scale adaptive tracking using the CAVIAR dataset test sequence ThreePastShop2cor.mpeg. Frame 1 is offset from the CAVIAR video by 390 frames. A magnified view in inset from frame 225 onwards

value of  $\varepsilon$  was used as 0.05 and the reduction factor  $\eta=0.98$  was employed. We also included a 'full explanation' scheme. When the target is occluded partially, value of  $u$  increases suddenly. So no scale reduction is applied if  $u_n$  is more than 1.5 times of  $u_{n-1}$ , even if equation (4) is satisfied. We get the correct position of partially occluded target using the unoccluded fragments at the correct scale i.e. using full explanation. This facilitates tracking when target becomes unoccluded again. We do not employ scale reduction during partial occlusion i.e. partial explanation. This solves the partial versus full explanation dilemma explained in [7].

## V. RESULTS AND DISCUSSION

In Figure 2, we demonstrate the performance of our tracker as the target undergoes significant scale changes. Note the systematic reduction in size of the tracker as the target moves away. The tracking window does not shrink rapidly even though object is uniform in color. This is an improvement over the FragTrack algorithm[7], as explained in Section I.

In Figure 3, the performance of our algorithm in case of scale variation is evaluated using a tracking sequence from the CAVIAR test dataset. Tracking for all CAVIAR sequences was carried out at 25 fps.



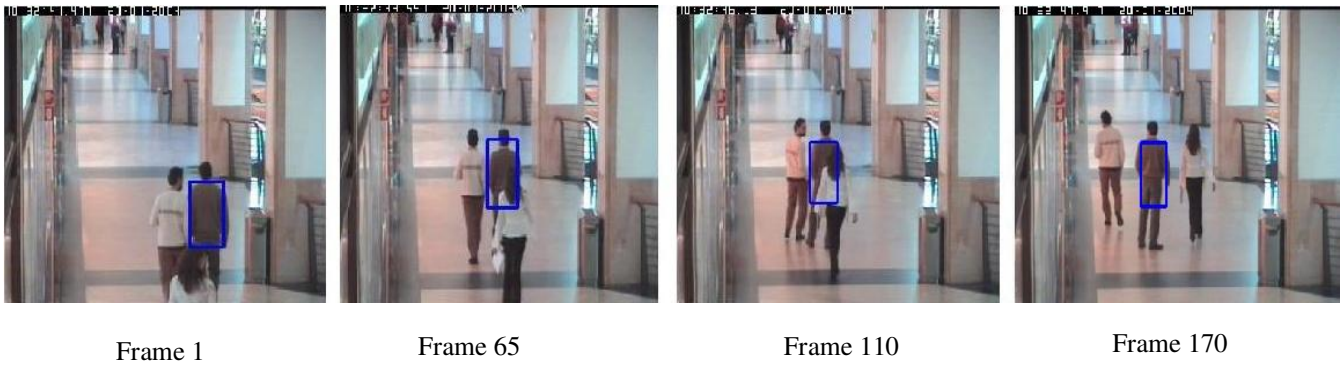


Figure 4. Example of robustness of the algorithm to partial occlusion using the CAVIAR dataset test sequence OneShopOneWait1cor.mpg. It can be seen in Frame 65 and Frame 110 that tracking accuracy is maintained even when target is partially blocked. The frame 1 is offset from the CAVIAR video by 225 frames.

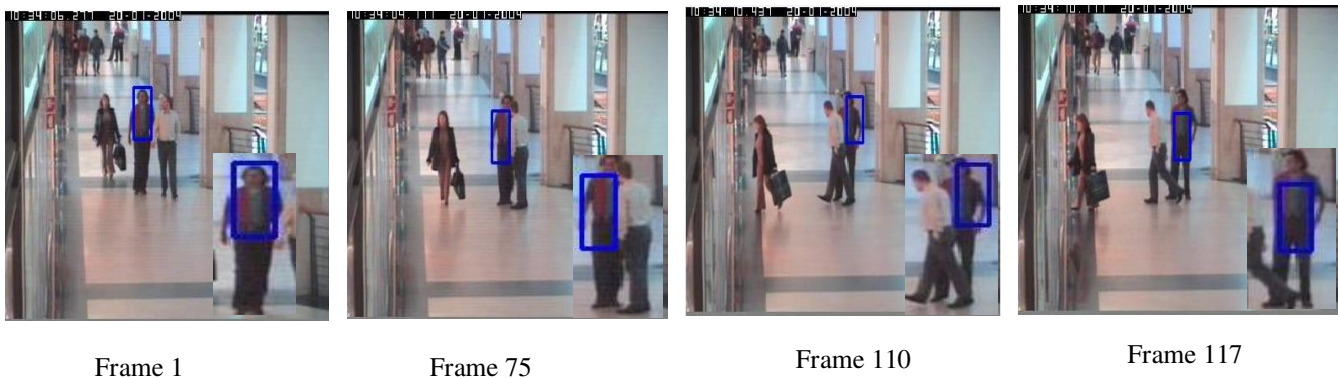


Figure 5. Another example of partial occlusion handling from the CAVIAR sequence 'OneShopOneWait2cor.mpg'. The frame 1 is offset from the CAVIAR video by 195 frames. A magnified view in inset

Our algorithm is also robust to partial occlusion in contrast to some other fragments based approaches including [3, 6]. In our target template, we fix the positions of fragments within the target. So when it gets partially occluded, the fragments which are not occluded contribute towards the selection of best match. Hence our tracker can handle partial occlusions successfully. We also opt for a full explanation scheme during occlusion as explained earlier, which improves the performance especially in case of uniform colored targets.

Figures 4 and 5 demonstrate the performance during partial occlusion using test sequences from the CAVIAR dataset. The tracker remains on the target in these examples even though it is partially occluded.

## VI. CONCLUSION

In this paper, we present an effective multiple fragment based algorithm for real time object tracking. We propose a systematic scheme for scale adjustment. We also demonstrate the robust performance of our tracker in presence of partial occlusion. In contrast, none of the

previously proposed multiple part or fragment based trackers are both scale adaptive and robust to partial occlusion. We also provide conclusive experimental proof using standard test sequences as well as out-of-the lab real life sequences.

In terms of future work, an important improvement can be incorporating a method for selection of only 'informative features' to track. These fragments would be able to provide better localization properties as compared to a rigid selection mechanism of fragments.

## ACKNOWLEDGMENTS

We gratefully acknowledge the valuable comments provided by Prof. A. S. Abhyankar, VIIT, Pune and Dr. Sumantra Dutta Roy, Dept. of EE, IIT Delhi. We would also like to thank Mr. V. Srikrishnan from IIT Bombay for his timely help

## REFERENCES

- [1] W.M. Hu, T.N. Tan and L. Wang, "A survey on visual surveillance of object motion and behaviors," IEEE Transactions on Systems, Man, and Cybernetics Part C, vol. 34, pp. 334-512, March 2004.

- [2] K. Okuma, A. Taleghani, J.F.G. de Freitas, J.J. Little, and D.G.Lowe. "A Boosted Particle Filter: Multitarget Detection and Tracking." In European Conference on Computer Vision (ECCV'04), volume I, pages 28-39, 2004
- [3] E. Maggio and A. Cavallaro. "Multi-part target representation for color tracking." IEEE International Conference on Image Processing, (ICIP '05), vol. 1, pp. 729-732, 2005.
- [4] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking". IEEE Trans. PAMI, 25(5):564-575, 2003
- [5] J. Jeyakar, R.V. Babu, K.R. Ramakrishnan, "Robust object tracking using local kernels and background information" in: IEEE International Conference on Image Processing (ICIP '07), vol. 5, pp. 49-52, 2007.
- [6] V. Srikrishnan, T. Nagaraj, S. Chaudhuri, "Fragment Based Tracking for Scale and Orientation Adaptation" in Sixth Indian Conference on Computer Vision, Graphics & Image Processing, (ICVGIP '08) pp 328-335, 2008.
- [7] A. Adam, E. Rivlin and I. Shimshoni, "Robust Fragments-based Tracking using the Integral Histogram" IEEE Conference on Computer Vision and Pattern Recognition (CVPR'06), pp.798-805, 2006.
- [8] J. Satake and T. Shakunaga, "Multiple target tracking by appearance-based condensation tracker using structure information," the 17th International Conference on Pattern Recognition (ICPR'04), Vol. 1, pp. 537-540, August 2004.
- [9] EC Funded CAVIAR project/IST 2001 37540, found at URL: <http://homepages.inf.ed.ac.uk/rbf/CAVIAR/>
- [10] F. Aherne, N.Thacker, P. Rockett, "The Bhattacharyya metric as an absolute similarity measure for frequency coded data". Kybernetika, 32(4), pp 1-7, 1997
- [11] F. Wang, S. Yu, J. Yang, "A Novel Fragments-based Tracking Algorithm using Mean Shift", 10th Intl. Conf. on Control, Automation, Robotics and Vision (ICARCV'08), pp.694-698, Dec. 2008