ROBUST OBJECT TRACKING USING LOCAL KERNELS AND BACKGROUND INFORMATION

Jaideep Jeyakar, R. Venkatesh Babu, K. R. Ramakrishnan

Computer Vision Lab
Dept. of Electrical Engineering
Indian Institute of Science, Bangalore, India

ABSTRACT

The mean shift algorithm has been proved to be efficient for tracking 2D blobs through a video sequence. Even so, this algorithm has certain inherent disadvantages. In this paper, we propose a robust tracking algorithm which overcomes the drawbacks of global color histogram based tracking. We incorporate tracking based only on reliable colors by separating the object from its background. A fast yet robust model updation is employed to overcome illumination changes. This algorithm is computationally simple enough to be executed real time and was tested on several complex video sequences. The proposed technique could be easily extended to other tracking algorithms too.

***Index Terms*—** Object tracking, Mean Shift tracking, fragment based tracking, foreground separation, adaptive tracking

1. INTRODUCTION

The objective of target localization or object tracking is to faithfully locate a previously specified target in subsequent video frames. Object tracking is one of the most important tasks in computer vision and finds applications in surveillance, activity analysis, video understanding etc. Though several tracking algorithms have been suggested, challenges such as occlusion, changes in illumination, cluttered background and low contrast are still under investigation.

The mean shift tracking algorithm [1] is a well known and efficient technique for blob tracking where the target and candidate models are represented by their color histograms. The algorithm is used to determine the position of the candidate blob which minimizes the (usually Bhattacharya) distance between the candidate model and target model. Since it is a gradient ascent algorithm, it is computationally very simple. Even though this algorithm is fast and performs satisfactorily for various sequences, it has certain drawbacks. It is not robust to extremely fast moving objects and illumination changes. Since the object model is based on the global color histogram without spatial information, a drift is observed when the object undergoes partial occlusion. There is also no inherent mechanism to track objects of varying sizes. Multi-part tracking proposed in [2] does not describe a voting strategy and is not resilient to occlusion. Adam *et al.* [3] use the resource intensive integral histogram.

Several improvements to the mean shift tracking algorithm have been proposed. Collins [4] suggested an elegant method to integrate search in the scale space within the mean shift frame work. A new similarity measure has been suggested in [5] as an alternative to the Bhattacharya distance to include spatial information but is computationally expensive. Birchfield et. al. [6] also talks about how to incorporate some spatial information in the model description. Mean shift tracking has also been combined with the SSD tracker [7], particle filter [8] and Kalman filter.

In this paper, we describe a fragment based representation of the target blob. Tracking is done based on the fragment which is most reliable. A voting strategy is suggested to select the best fragment. By using information about the object and its background, we assign a confidence measure to each histogram bin. Only bins with high confidence is relied on for tracking. Finally, a robust, yet fast target updating is done by replacing selected target fragments with the most recently observed fragment.

This paper is organized as follows. Section 2 explains the suggested fragment based tracker. The problem of histogram weighing and model updation are discussed in Sections 4 and 5 respectively. Results illustrating the performance of the tracker is discussed in Section 6. Finally, concluding remarks are given in section 7.

2. MULTIPLE FRAGMENT-BASED OBJECT REPRESENTATION

Many region based tracking algorithms rely on histograms for blob tracking. This retains only the chromatic composition of the object while discarding all spatial information. This induces attractions to false targets and local minimas. On the other hand, algorithms which use spatial information are computationally intensive.

Initially, the target to be tracked is divided into fragments as shown in Fig. 1. These fragments may or may not overlap with each other. A target model for each fragment is built by taking its color histogram after centering an Epanechnikov kernel in it. We use the joint color histogram in the RGB space. Each dimension in the RGB space is divided into 10 bins. Once the next frame arrives, a candidate model is generated for each fragment at the same location. The mean shift tracking algorithm is applied to find which regions in the new frame are most similar to the target model of each fragment.

Each fragment also casts a vote for the position of the whole object. The Bhattacharya coefficient ρ calculated using the candidate model at y, $\widehat{\mathbf{p}}(y)$, and the target model $\widehat{\mathbf{q}}$ is a good measure of the confidence of each fragment's trajectory and can be used to resolve the voting. Using the most confident estimate of the object location, the coordinates of the winning fragment can be used to reset the positions of all other fragments.

$$x_o = x_J - \Delta_J \tag{1}$$

$$J = \arg\max_f \rho[\widehat{\mathbf{p}}_{f,t}(y), \widehat{\mathbf{q}}_{f,t}] \tag{2}$$

Where the object position x_o is obtained from the fragment with most confidence. Δ_f is the difference vector of the $f-th$ fragment and the blob position in the target. The positions of all other

Fig. 1. Fragments

fragments are reset using

$$x_f = x_o + \Delta_f \quad (f = 1...F,\ f \neq J) \tag{3}$$

The number of fragments and their demarcation plays an important role in the robustness to occlusion. Though the robustness increases with the number of fragments, too many fragments would mean increased processing time for each frame. Since the computation required for each frame depends greatly on the size of each fragment, we need to restrict that as well. Further, selecting very small fragments would result in a tracking drift or discarding some information about the target. Hence a tradeoff is required. This prompts us to use a few overlapping fragments. Contrary to [2, 3], we found that using four overlapping fragments as shown in Fig. 2 gives us satisfactory results.

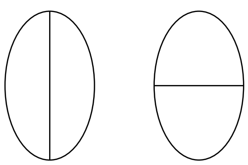


Fig. 2. A good choice of fragments

3. FOREGROUND SEPARATION

An object to be tracked must be visually separable from its background. Similar to [7], our algorithm calculates the log likelihood of the probability of a color being found in the foreground of the region of interest. It is done as follows.

The blob being tracked is taken as the region of interest (*object window*). Another window called the *background window* of equal area is defined around the object window as shown in Fig. 3a. The joint color histograms h_{ob} and h_{bg} in the RGB space are calculated over the object and background windows. $L(x_i)$ gives an indication of the probability that a particular pixel x_i belongs to the foreground.

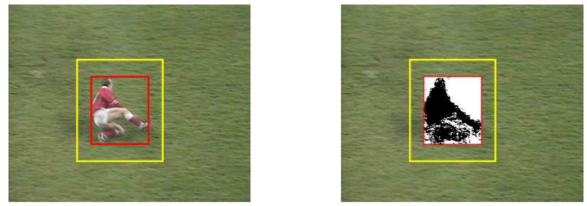
$$L(x_i) = \log \frac{\max(h_{ob}[b(x_i)], \epsilon)}{\max(h_{bg}[b(x_i)], \epsilon)} \tag{4}$$

$b(x_i)$ is a function that maps the pixel at x_i to its histogram bin. A small constant ϵ is included in the equation to avoid numerical instability.

The likelihood is thresholded to discard all background pixels (White pixels in Fig. 3b)

$$T(x_u) = \begin{cases} L(x_i) & L(x_i) > th_o \\ 0 & otherwise \end{cases} \tag{5}$$

The advantages of calculating the likelihood are two fold. In addition to giving measure of confidence for each bin in the histogram, $T(x_i)$ is used to selectively update fragments. In our experiments the threshold th_o is set at 0.8 in order to choose object pixels with higher confidence.



(a) Background & Foreground (b) Thresholded Likelihood

Fig. 3. Foreground / Background separation

4. FOREGROUND BASED TRACKING

We would like to develop an algorithm which takes into account the fact that the object to be tracked has a different color distribution when compared to its background. Background weighed histograms were introduced in [1], but is not very robust.

We are interested in designing a target model which takes the information of the background/foreground into account. $L(x_i)$ defined in (4) can take on a large range of both positive and negative values. We use a sigmoid function to map the likelihood to weights that can take values between $(0, 1)$. For each histogram bin u, the weight λ_u is calculated as

$$\lambda_u = \max\left(1 - \frac{1}{1 + exp\left\{-\frac{(L_u - a)}{b}\right\}}\ ,\ 0.1\right) \tag{6}$$

Thresholding is done to reduce jitter. a is chosen based on our relative confidence we have in the foreground region. b controls the slope of mapping function. Typical values $(a, b) = (1, 1)$ work well.

Like the modified mean shift tracking algorithm described in [1], we define the target and candidate models centered at y as

$$q_u = \frac{1}{C} \sum_{i=1}^{n} k\left(x_i\right) \lambda_u \delta\left[b(x_i) - u\right] \tag{7}$$

and

$$p_u(y) = \frac{1}{\tilde{C}} \sum_{i=1}^{n} k\left(\left|\left|\frac{y - x_i}{h}\right|\right|^2\right) \tilde{\lambda}_u \delta\left[b(x_i) - u\right] \tag{8}$$

where k is either the Epanechnikov or gaussian kernel with bandwidth parameter h (See [1]). λ_u is calculated using the target and its surrounding. Similarly, $\tilde{\lambda}_u$ is calculated using the candidate location centered at y and its neighborhood. C and $\tilde{C}$ are normalization constants which make $\sum q_u = 1$ and $\sum p_u = 1$.

Since the vectors $p(y)$ and q are of unit length, the Bhattacharya distance is still a valid metric. Hence we use the Bhattacharya coefficient (ρ)

$$\rho = \sum_{u=1}^{M} \sqrt{p_u(y) q_u} \tag{9}$$

Maximizing the Bhattacharya coeff. leads to the following mean shift iterations. The calculation of the mean shift vector and tracking is done as in [1]. The new estimate of the target position $\widehat{y}_1$ is calculated to be a weighted sum of pixels contributing to the model.

$$\widehat{y}_1 = \frac{\sum_{i=1}^{n} x_i w_i g\left(||\frac{y-x_i}{h}||^2\right)}{\sum_{i=1}^{n_h} w_i g\left(||\frac{y-x_i}{h}||^2\right)} \tag{10}$$

$$w_i = \sum_{u=1}^{m} \delta[b(x_i - u)] \sqrt{\frac{q_u}{p_u(y)}} \tag{11}$$

5. TARGET ADAPTATION SCHEME

In most object tracking problems, we cannot be sure that the object would remain exactly the same as it was when the target model was defined. It may undergo scale changes, appearance changes and illumination changes. Hence we must adapt the target model as the object undergoes a change. If the updation is done slowly, targets which change quickly cannot be tracked faithfully. But if the updation is very fast, the algorithm may learn the wrong model and we will be tracking an unpurposed target. This calls for making a trade off.

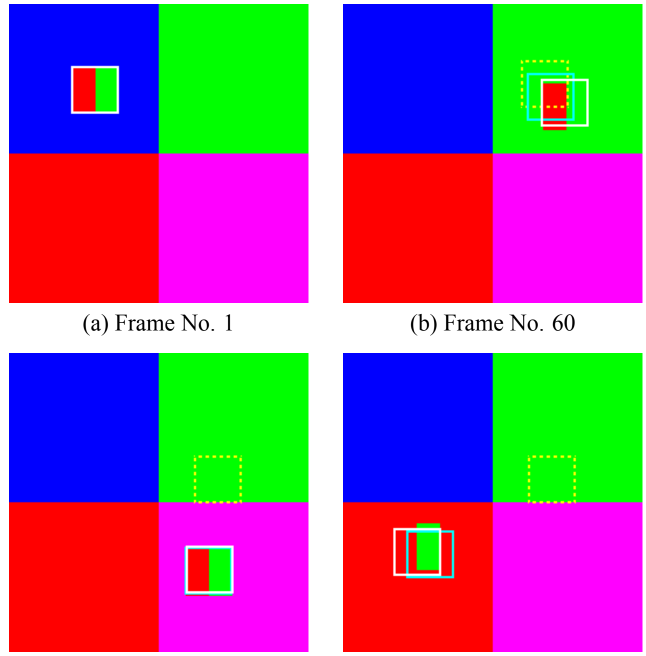
Fragment based tracking allows us to have reliable tracking, as well as update the model quickly. Most objects are of an irregular shape. While adapting the target, we only want to learn pixels which fall in the object rather than background regions. Once we know if a pixel belongs to the foreground or not, we can decide whether or not to update a fragment depending on the number of foreground pixels it contains. A threshold of 80% works well. The fragments that have many foreground pixels and high Bhattacharya coefficient are directly replaced by the current candidate model.

$$\widehat{\mathbf{q}}_{f,t} = \begin{cases} \widehat{\mathbf{p}}_{f,t} & \sum_{i=0}^{n_f} T(x_i) \geq th_1 \ \& \ \rho[\widehat{\mathbf{p}}_{f,t}(y), \mathbf{q}_{f,t}] \geq th_2 \\ \widehat{\mathbf{q}}_{f,t-1} & otherwise \end{cases} \tag{12}$$

This updation scheme is much faster than others which involve taking a linear combination of the current and previous models. In addition, we ensure that we do not cause any tracking drift caused by learning the background into the target model. This improves the robustness of the tracker since we do not depend on any one particular fragment for tracking.

6. RESULTS

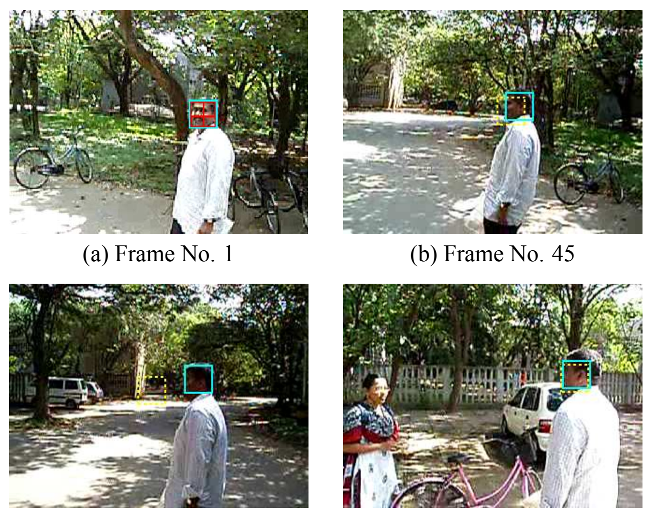
We present some of our results in this section. The yellow dashed rectangle shows the output of a mean shift tracker. The cyan solid rectangle shows the result of using our improved tracking algorithm using background information. Fig. 4 illustrates, using a synthetic video, how the histogram weighing works. When the object (small square containing green and red colors) is surrounded by blue (Fig. 4a), both red and green colors contribute to the tracking. When the object goes into a green background, green is given a low importance and the tracking is based on the red object region (hence the red region is centered). Similarly, both colors are given a equal weightage in Fig. 4c and only green is given importance in Fig. 4d where



(a) Frame No. 1 (b) Frame No. 60

(c) Frame No. 110 (d) Frame No. 164

Fig. 4. Synthetic example - Tracking based on selected colors



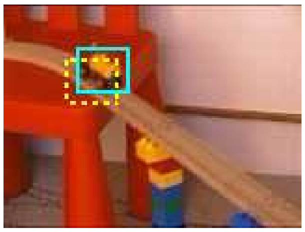
(a) Frame No. 1 (b) Frame No. 45

(c) Frame No. 88 (d) Frame No. 200

Fig. 5. Robustness to illumination change. Initial fragments are shown in (a)

the background is red. The mean shift tracker loses the object during transition from green to the magenta background, whereas, when background information is used, the object could be tracked throughout the sequence with a drift. The fragment based tracker with two vertical fragments (white box) also improves the performance of the tracker and we observe less drift (Fig. 4d).

Figure 5 shows how the adapting scheme works when a person walks from a bright area into a shaded area. Even though there is a sharp change in illumination, the face is tracked well. The mean shift algorithm fails to track the face when it moves from bright to dark region since the target model is fixed, but localizes the face correctly when it moves back into the bright region. In Fig. 6, the object passes through some clutter having the same color of the object (toy train) which is mainly composed of colors red and yellow. Tracking is good because in Fig. 6a, yellow is given high priority (red clutter). In Fig. 6b, c, d, the red region of the object is given a high priority (yellow clutter). Fig. 7 (obtained from authors of [3]) shows the robustness to partial occlusion. For example, in Fig. 7b, when the left half of the face is occluded, tracking is done based on the fragments occupying the right half.



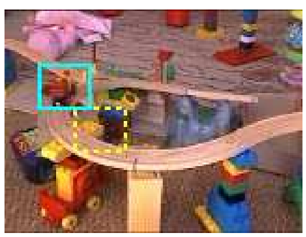
(a) Frame No. 57

(c) Frame No. 526

(b) Frame No. 529

(c) Frame No. 809

(d) Frame No. 885



(d) Frame No. 939

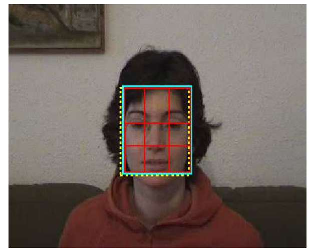
Fig. 6. Robustness to clutter

7. CONCLUSION

We have presented a multi-fragment representation of the target and candidate models to improve the robustness of tracking especially to partial occlusion by including spatial information of the object. We also describe a scheme by which the tracking relies on distinct colors of the object against the background clutter. The use of fragments also enables us to rapidly update the target to be tracked without learning the background into the target model. Techniques presented here can be easily applied for real-time tracking and can be extended to other tracking algorithms as well.

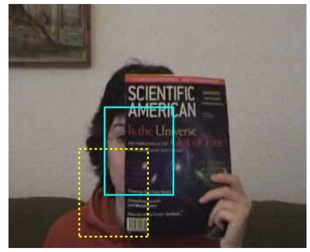
8. REFERENCES

[1] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 564–577, 2003.

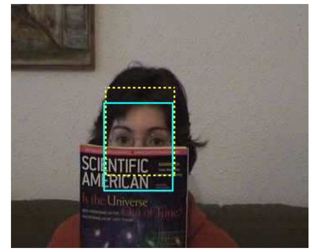


(a) Frame No. 1

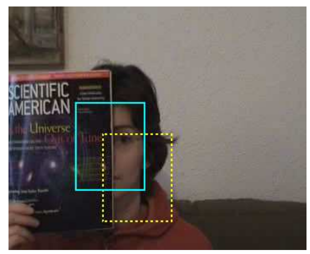
(b) Frame No. 200



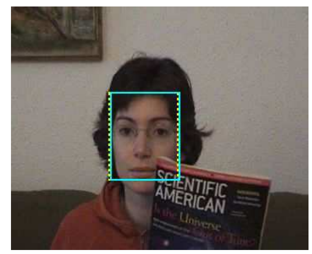
(c) Frame No. 527



(d) Frame No. 635



(c) Frame No. 704



(d) Frame No. 896

Fig. 7. Fragments increase robustness to occlusion. The six fragments are shown in (a)

[2] E. Maggio and A. Cavallaro, "Multi-part target representation for color tracking," *IEEE International Conference on Image Processing*, vol. 1, pp. 729–732, 2005.

[3] Amit Adam, Ehud Rivlin, and Ilan Shimshoni, "Robust fragments-based tracking using the integral histogram," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 798–805, 2006.

[4] R. T. Collins, "Mean-shift blob tracking through scale space," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 234–240, 2003.

[5] Changjiang Yang, Ramani Duraiswami, and Larry Davis, "Efficient mean-shift tracking via a new similarity measure," *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

[6] S. T. Birchfield and S. Rangarajan, "Spatiograms versus histograms for region-based tracking," *IEEE Conf. on Computer Vision & Pattern Recognition*, vol. 2, pp. 1158–1163, 2005.

[7] R. Venkatesh Babu, P. Perez, and P. Bouthemy, "Robust tracking with motion estimation and local kernel-based color modeling," *Image and Vision Computing*, vol. 25, pp. 1205–1216, 2007.

[8] S. T. Birchfield and S. Rangarajan, "Hybrid particle filter and mean shift tracker with adaptive transition model," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 221–224, 2005.