IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 10, October 2017

Robust Non-rigid Object Tracking using Patch Distribution

N. Kiran Babu[1], P. Anil Kumar[2], G. Ezra Sastry[3]

Assistant Professor, Computer Science and Engineering, Narasaraopeta Engineering College, Guntur, India[1,2,3]

Abstract: We propose a novel tracking algorithm for targets with drastically changing geometric appearance over time. To track such objects we use online update of appearance and shape. The problem here is to model foreground appearance of target with histograms in such a way that is accurate and efficient. In this paper, the constantly changing target foreground shape is modelled as a small number of patches or rectangular blocks, whose position within the tracing window are adaptively determined. We show that the robust tracking is possible by adaptively adjusting these patches. Moreover, object segmentation result is integrated into the proposed model to further enhance it. Experimental results show that the algorithm is able to track non-rigid objects undergoing large variation is appearance and shape.

Keywords: Non-rigid Object, Object Tracking, Histograms, Segmentation.

I. INTRODUCTION

Developing an accurate, efficient visual tracker is challenging task, and the task becomes even more tough when the target is expected to undergo rapid variation in shape and appearance. It is one of the most important problems in computer vision. It can be used in a large number of different applications like intelligent robots, surveillance, medical imaging, and so on. In real world environment, objects are typically complex and difficult to track. Example of visual tracker is shown in Figure 1. In this example, the appearance changes are mainly due to change in shape while the intensity distributions remain roughly stationary. To track such object tracking algorithm need to consider the object's appearance changes adaptively.

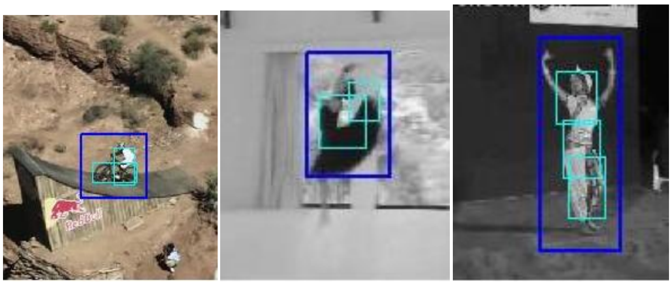

Figure 1: Example of tracking results. The proposed method successfully tracks a target even the target's geometric appearance changes.

Perhaps intensity histograms are the easiest way to represent target appearance, and tracking methods based on this idea are available in the paper (e.g., [1, 2]). Computing intensity histograms from a region bounded by irregular shapes can't be done efficiently. To handle shape variation in histogram based tracking, one idea is to use kernel [3, 4] to define a region around target.

Another way to deal with irregular shapes is to enclose the target in a rectangular shape and compute the histogram for that enclose region. Here the problem is background pixels are included and the foreground shape cannot be closely approximated. As a result of noise, histogram can be corrupted by background pixels, and the results are not efficient.
Each of the above mentioned problems has been solved to some extent (e.g., [5] encoding spatial information in histograms). However, most of them require substantial increase of computation time and such methods are not able to track targets undergoing rapid motions. Here, we propose a tracking algorithm to solve the above mentioned problems. The propose method consists of global scanning for target, and refinement, and update step. Target object can be located by scanning the entire image. Here For shape updating we use small rectangular blocks called patches within the tracked window. The tracking windows are small enough to extract the target contour using segmentation algorithms without increasing complexity of algorithm. Then we update the shape of target by adjusting patches locally, they provide full coverage of the foreground target.

 DOI10.17148/IJARCCE.2017.61013

II. RELATED WORK

Tracking algorithms for articulated objects: In paper [6] Schindler represents an object as the constellations of parts to track bee with the Particle Filter. Active contours using parametric models [7, 8] require offline training. With all the offline training, still it is difficult to predict the tracker's behavior when unseen target is encountered.
Sampling based tracking algorithms: The particle filter [9] has shown efficiency in handling non-gaussianity and multi-modality. The MCMC (Markov Chain Monte Carlo) method is well applied to multi object tracking [10, 11]. However, these methods still suffer from the problem of being trapped in deep local optima and handling vast number of samples.
Kernel-based algorithms: kernel-based methods represent target's appearance with intensity, gradients, and color [1, 2], but they are not effective for representing non-rigid objects. Although methods using multiple kernels are not able to track articulated objects whose shapes vary significantly.
Tracking algorithms based on segmentation results: Lu and Hager in [12] take the tracking problem as binary classification by using dynamic foreground and background appearance models. In [13] the representation of target model as Gaussian mixture model with multiple fragments and target accurate boundaries using level sets. These are used to acquire the dynamic shape of the object over time. These two methods did not consider the geometric structure of the targets. Compared with above mentioned methods, the proposed method covers the geometric structure of the targets by using the rectangular blocks. Hence, the method tracks the targets robustly for which geometric appearance changes over time.

III. TRACKING METHOD

The details of the proposed tracking method are presented here. The output of proposed method shows the rectangular window enclosing the target in each frame. The representation of target is crucial to the success of tracking approach. Originally, the representation is taken as single rectangular block which encloses the target. However, with conventional representation, the tracking methods may fail. To overcome this problem, the tracking method requires advanced target representation as shown in Fig 2.

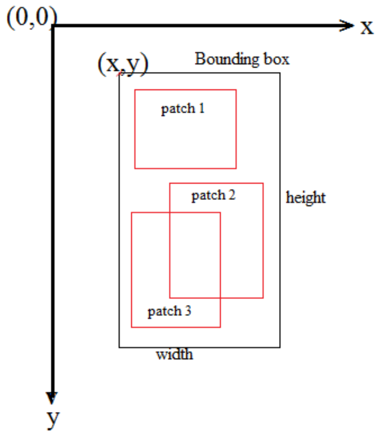

Figure 2: Example of Patch-based model.

The outline of the proposed method is shown in Figure 3, and the proposed algorithm is shown in Figure 4. It has three sequential steps: detection of windows, refinement, and update. First the tracker is initialized with the contour of target, and then it automatically determines the initial tracking window T and R rectangular blocks Bi as well as their weights λi. Here we maintain the foreground intensity histogram H_0^f throughout the sequence. The shape of foreground target is approximated by R rectangular blocks with in the tracking window T as shown in Figure 2. The blocks position within the widow T is adaptively adjusted throughout the tracking sequence. At each frame i, the tracker maintain the followings:

1. A template window T_i with block configuration.
2. A foreground histogram H_i^f represented by local foreground histograms, $H_i^{B_i^f}$ and their associated weights.
3. Background histogram H_i^b.

The tracker detects the likely location of the target by scanning the entire image. In the refinement step the target is segmented from the background using the current foreground intensity. The result is then used to update the block positions in the template window T and then weights associated with each block are recomputed.

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 10, October 2017

Target Detection

The tracker first scans the entire image to locate the target. The target window T* selected in this step has the maximum similarity measure S with reverence to the current template window T,

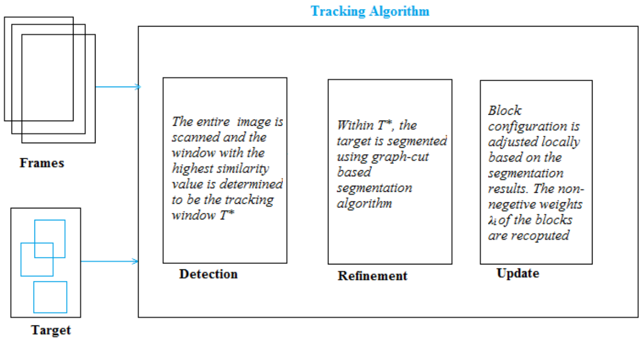

Figure 3: Tracking method

$$T^* = \max S(T', T), \qquad (1)$$

as T' ranging over all scanned windows. The similarity measure can be computed as follows. The local foreground histogram $H_i^{B_t^f}$ for the block Bt is the intersection of raw histogram $H_i^{B_t}$ with the initial foreground histogram of the respected block:

$$H_i^{B_t^f}(b) = \min\left(H_i^{B_t}(b), H_0^{B_t^f}(b)\right),$$

where b indexes the bins. The value of S can be computed by

$$S(T', T) = \sum_{t=1}^{R} \lambda_t d\left(H_0^{B_t^f}, H_i^{B_t^f}\right) \qquad (2)$$

where λt is the weight associated to block Bt and d is the Bhattacharyya distance between two intensity values,

$$d\left(H_0^{B_t^f}, H_i^{B_t^f}\right) = \sum_{b=1}^{N} \sqrt{H_i^{B_t}(b) H_0^{B_t^f}(b)},$$

where N is the number of bins. By using the λt, S will down weights the blocks having the more background pixels, and it provides some measure against clutters and background noise.

Refinement Process

After finding the window T* in which the target is located, the next step is to find an approximate boundary contour so that shape variation can be maintained. Graph-cut segmentation algorithm can be used to segment out the foreground object in T*. In previous work on this type of segmentation process of visual tracking (e.g., [14, 15]) define the cost function in the form of

$E = E_A + \mu E_S,$

where E_A and E_S are related to appearance and shape, respectively. Our solution will be to use only the appearance term E_A.

Let p denote a pixel and $\mathbf{p}$ denote the set of all pixels in T*. Let D_B denote the background density that was estimated in the previous frame, and D_i, $1 \le i \le R$ the foreground intensity from B_i (by normalizing the histogram $H_i^{B_i^f}$). We also denote D_f the foreground intensity obtained by normalizing the current foreground histogram H_i^f. The graph-cut algorithm [15, 16] will minimize the cost function

$$E(C_p) = \mu \sum_{p \in \mathbf{p}} K_p(C_p) + \sum_{(p,q) \in N : C_p \neq C_q} B_{p,q}, \quad (3)$$

where $C_p : \mathbf{p} \rightarrow \{0, 1\}$ is a binary assignment function on $\mathbf{p}$ such that for a given pixel p, C(p) =1 if p is foreground pixel and 0 otherwise.μ is a weighting factor and N denotes the set of neighboring pixels. We define

$$B_{p,q} \propto \frac{\exp\left(-(I(p)-I(q))^2 / 2\sigma^2\right)}{||p-q||},$$

where I(p) is the intensity value at pixel p and σ is the kernel width. The term $K_p(C_p)$ is given as

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified

Vol. 6, Issue 10, October 2017

$$K_p (C_p = 0) = -\log D_F (I(p), p)$$
$$K_p (C_p = 1) = -\log D_B (I(p)),$$

where $D_F (I(p), p) = D_i(I(p))$ if p is contained in B_i, and $D_F (I(p), p) = D_f(I(p))$ if p is not contained in B_i. Now the shape information is implicitly encode through D_F. We only perform the graph-cut in relatively small window, this can be done quickly and does not increase the computational complexity.

Algorithm 1:

for each frame
- scan entire image to find **T***.
- extract the target contour from T* using graph-cut segmentation algorithm.
- Adjust the blocks based on target contour.
- Update weights λ_i.

end

Updating Process

We can update the positions of blocks Bi with in T* after getting the segmentation results. We locally adjust these blocks so that they can cover maximal area of the segmented foreground. We use a greedy policy to cover the entire segmented foreground by arranging each block locally using a priority based on their size. Figure 4 show one result of this block adjustment (shown in blue).

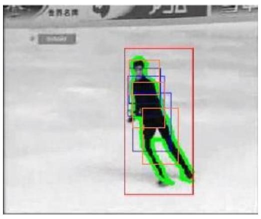

Figure 4: Block Adjustment

We can compute the foreground histogram from each block Bi. Furthermore the percentage of pixels £i in Bi that are foreground pixels can be computed quickly. To maintain the total foreground density, we solve a non-negative least square problem

$$H_0^f = \sum_{t=1}^{R} \alpha i \, H_i^{B_t^f}$$

with $\alpha i > 0$, and H_0^f the foreground density of the initial frame. The weights are updated according to $\lambda i = £i\ \alpha i$ followed by normalization.

IV. EXPERIMENTAL RESULTS

The proposed method has been implemented in MATLAB with some optimized MEX C++ subroutines. In this process we use 16 bin intensity histograms for gray scale video, and for color sequence, each color channel is binned separately with 16 bins. Here the number of blocks R is set to 2 or 3. We tested the tracker on various video sequences, and some of them are reported in this paper. On a Dell 2.4GHz machine, our tracker runs at 3 frames per second while the integral histogram tracker runs at 4 frames per second. The additional time incurred in our method comes from the updating process of patch configurations. However experimental comparisons show that this negligible over head in runtime complexity allow our tracker to produce much more satisfactory results.

In our experiments, for initialization, we manually locate the tracking window in the first frame, and for the experiments, both trackers start with the same tracking window. The sequences shown below are all collected from the web. The foreground targets undergo significant appearance changes, mainly due to the shape variation. As per the results our method can handle small amount of intensity variations also. The first female dancer sequence contains over 225 frames, and it has an incredible pose variation. As show in Figure 5, out tracker is able to track the dancer well and provides the results which are accurate and consistent. The second sequence contains 800 frames with face with partial occlusions. Our method is able to track the face throughout the whole sequence as shown in Figure 6.

DOI10.17148/IJARCCE.2017.61013

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 10, October 2017

The third and fifth sequence contains adverse conditions such as cluttered backgrounds, scale changes and rapid movements. In both shown in Figure 7, 9 our tracker is able to track and produce consistent results. Figure 8 presents the color video sequence of mountain rider.

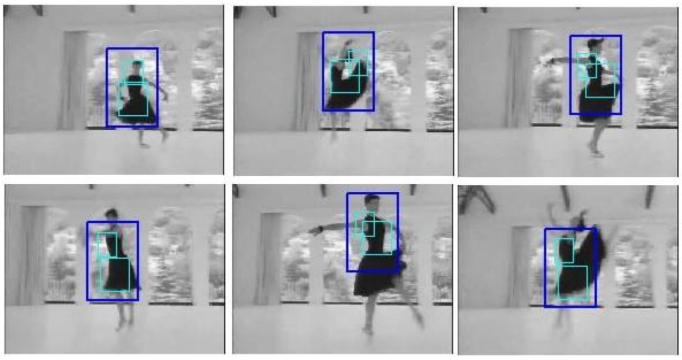

Figure 5: Dancer Performance

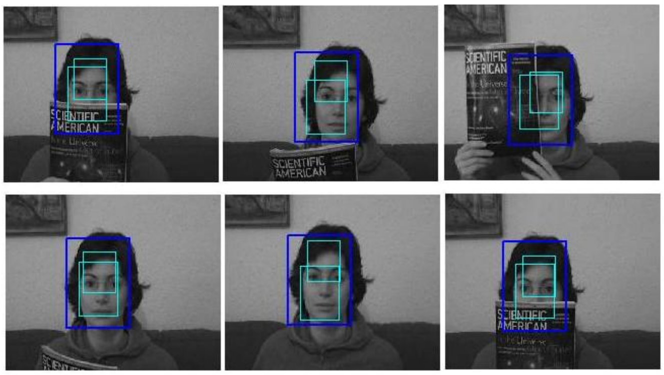

Figure 6: Face Track

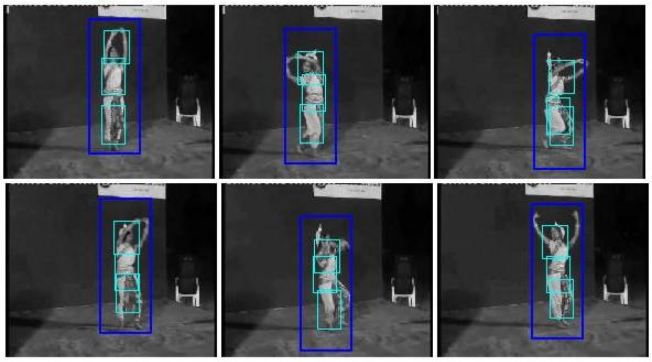

Figure 7: Indian Dancer

Figure 8: Mountain Rider

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 10, October 2017

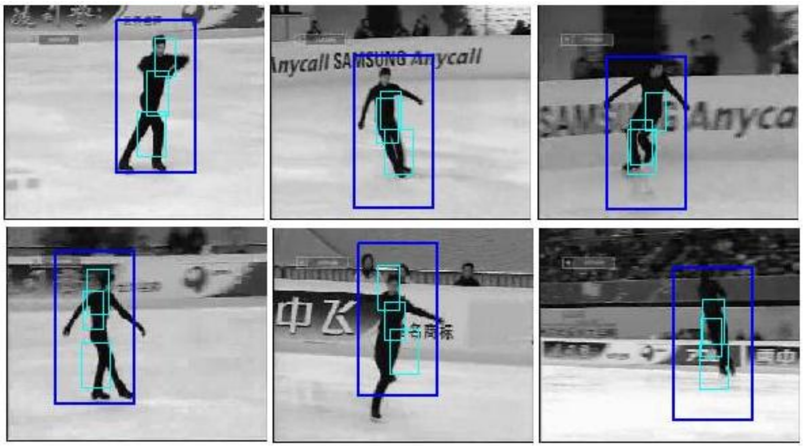

Figure 9: Male Scatter

V. CONCLUSION AND FUTURE WORK

In this paper, we developed a method for accurate tracking of objects undergoing shape variations (e.g., non-rigid objects). With the general assumption that the foreground intensity distribution is nearly stationary, we show that is possible to efficiently estimate shape changes using a collection of rectangular blocks. The method first locates the target by scanning the entire image using the foreground intensity histograms. The refinement step estimates the object contour from which the blocks can be repositioned and weighted.

This method is simple to implement and efficient in tracking. Experimental results have demonstrated that the proposed method consistently provides more precise tracking results when compared with integral histograms based tracker.

The current algorithm we have is not optimal but it is easy to implement with good empirical results. With mechanism of occlusion handling we can improve the tracker robustness. Future work aims at extending the proposed method to deal with occlusions and multi-objects.

REFERENCES

[1] S. Birchfield. Elliptical head tracking using intensity gradients and color histograms. IEEE Conf. on Computer Vision and Pattern Recognition, 1988.
[2] D.Comaniciu, V. Ramesh, and P.Meer. Real-time tracking of non-rigid objects using mean shift. IEEE Conf. on Computer Vision and Pattern Recognition, 2000.
[3] Z.Fan, M.Yang, Y.Wu, and G.Hua. Efficient optimal kernel placement for reliable visual tracking. IEEE Conf. on Computer Vision and Pattern Recognition, volume 1, 2006.
[4] V. Parameswaran, V. Ramesh, and I. Zoghlami. Tunable kernels for tracking. IEEE Conf. on Comp.Vision and Patt. Recog., volume 2, 2006.
[5] S. Birchfield and S. Rangarajan. Spatiograms versus histograms for region-based tracking. IEEE Conf. on Computer Vision and Pattern Recognition, volume 1, 2005.
[6] G. Schindler and F. Dellaert, A Rao-Blackwellized Parts-Constellation Tracker. IEEE Int'l Conf. Computer Vision Workshop, 2005.
[7] A. Blake and M. Isard. Active Contours. Springer, 2000.
[8] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. Int. J. Computer Vision, 1998.
[9] M. Isard and A. Blake, Icondensation: Unifying Low-Level and High-Level tracking in a Stochastic Framework, Proc. Fifth European Conf. Computer Vision, 1998.
[10] Z. Khan, T.Balch, and F. Dellaert, MCMC-Based Particle Filtering for Tracking a Variable Number of Interacting Targets, IEEE Trans. Pattern Analysis and Machine Inteligence, volume 27, no. 11, Nov. 2005.
[11] K. Smith, D.G. Perez, and J. Odobez, Using Particles to Track Varying Numbers of Interacting People, Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2005.
[12] L. Lu and G.G. Hager, A Non-parametric Treatment for Location/ Segmentation Based Visual Tracking. Proc. IEEE Conf. Computer Vision and Pattern Recongnition, 2007.
[13] P. Chickalingam, N. Pradeep, and S. Birchfield, Adaprive Fragments-Based Tracking of Non-Rigid Objects using Level Sets, Proc. 12th IEEE Int'l Conf. Computer Vision, 2009.
[14] D. Cremers. Nonlinear dynamical shape priors for level ser segmentation. IEEE Conf. on Computer Vision and Pattern Recognition, 2007.
[15] D. Freedman and T.Zhang. Interactive graph cut based segmentation with shape priors. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 2005.
[16] Y. Boykov and M.P. Jolly. Interactive graph cits for optimal boundary and region segmentation of objects in n-d images. Int. Conf. on Computer Vision, volume 1, 2001.
[17] F. Porikli. Integral Histogram: A Fast way to extract histograms in Cartesian space. IEEE Conf. on Computer Vision and Pattern Recognition, volume 1, 2005.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 10, October 2017

BIOGRAPHIES

N. Kiran Babu working as an Assistant Professor in the department of Computer Science and Engineering in Narasaraopeta Engineering College (Autonomous) at Narasaraopet. M. Tech from JNTUK, Kakinada. B. Tech with discipline CSE from Bapatla Engineering College, Bapatla. Qualified in GATE twice in the years 2011 and 2012 with the discipline Computer Science and Engineering (CS). Qualified in UGC-NET as JRF in JUNE-2017 and as Lectureship JUNE-2014 with the discipline Computer Science and Applications. His Research areas of interest are Image Processing, and Data Mining. Totally he had more than two years of teaching Experience.

Anil Kumar Prathipati working as an Assistant Professor in the department of Computer Science and Engineering in Narasaraopeta Engineering College (Autonomous) at Narasaraopet. M. Tech from Hyderabad Central University, Hyderabad. B. Tech with discipline CSE from Bapatla Engineering College, Bapatla. Qualified in GATE twice in the years 2006 and 2007 with the discipline Computer Science and Engineering (CS). Qualified in UGC-NET Lectureship twice in JUNE-2012 and Dec-2015 with the discipline Computer Science and Applications. His Research areas of interest are Speech Processing, Image Processing and Data Mining. Totally he had more than nine years of teaching Experience.

G. Ezra Sastry He is received the B. TECH(CSE) degree from Bapatla Engineering college, Nagarjuna University, Guntur, Andhra Pradesh, India, M.TECH (IT) degree from Vignan University, Guntur, Andhra pradesh, India, Currently working as Assistant Professor in Computer Science and Engineering, NEC, Narasaraopet, A.P.
Specialization: Mobile computing, Computer Networks, Database Management Systems, Compiler Design.
Research Area: His interesting research field is Image Processing, Computer Networks, Wireless communications and networks and Mobile Computing.