2014 22nd International Conference on Pattern Recognition

Real-Time Tracking Combined with Object Segmentation

Hongzhi Wang, Nong Sang, Yi Yan
Science and Technology on Multi-spectral Information Processing Laboratory
School of Automation, Huazhong University of Science and Technology
Wuhan, China
{hzwang, nsang, yanyi}@hust.edu.cn

Abstract—**We propose a new approach that integrates object tracking with object segmentation in a closed loop. The EM-like algorithm for color-histogram-based object tracking is modified to deal with the appearance models of the object and background represented by the Gaussian mixture models which are more efficient in RGB color space. It provides a rough object spatial model to guide segmentation. A five-layer region based graph cuts algorithm is developed to extract the accurate object region based on the object spatial model. It is effective even in cluttered background and runs more than 10 times as fast as GrabCut. Then we can establish the appearance models of the object and background avoiding introducing errors and update them frame by frame without the problem of drift. The refined and adaptive models lead to robust tracking in return. Moreover, the motion of the object is estimated to produce a predicted object location in the new frame for tracking. A real-time robust tracking system is built based on the proposed approach and validated on a variety of challenging sequences.**

Keywords—tracking; object segmentation; EM-like; graph cuts; real-time

I. Introduction

Object tracking is an important subtask in many computer vision applications, such as surveillance, video indexing, and human computer interaction [1]. A large number of excellent algorithms have been presented in the past decades [2, 3, 4, 5, 6, 7, 8, 9]. Comaniciu et al. [2, 3] adopt the color histogram in an ellipse as the object appearance model and use the mean shift iterations to find the most probable object position in the new frame. Zivkovic et al. [4] present a natural extension of the mean shift tracking algorithm, which simultaneously estimates the object position and the covariance matrix that describes the approximate shape of the object. Although this new algorithm can adapt to the changes in shape and scale of the object, the problem of model update remains unsolved. In [5] the object and background are respectively divided into multiple regions and represented by a Gaussian mixture model (GMM) in a joint feature-spatial space. The object boundary is tracked by a level set method and the appearance model of each region is updated using a weighted average of the old values and the new values. Kalal et al. [9] propose a novel tracking framework (TLD) that combines tracking, learning, and detection. A learning method is developed to update the detector in runtime, which avoids drifting and reinitializes the tracking when the object reappears in the view of camera. In a recent review on the online object tracking [10], TLD is considered to be one of the top 10 algorithms. Although much progress has been made, object tracking in unconstrained videos remains a very challenging problem due to numerous factors, such as

- The object appearance changes sharply during tracking due to illumination variation. The methods [2, 3, 4, 5] without an effective strategy for model updating could easily lose the object.
- The tracked objects (e.g. hand and pedestrian) undergo out-of-plane rotation and significant deformation. The methods [6, 9] coarsely representing the object shape by a bounding box do not perform well in these scenarios.

In this paper, we propose a new approach that integrates object tracking with object segmentation in a closed loop. The tracking module provides a rough object spatial model to guide segmentation. The segmentation module extracts the accurate object region based on the object spatial model. Then we can establish the appearance models of the object and background avoiding introducing errors and update them frame by frame without the problem of drift. The refined and adaptive models lead to robust tracking in return. More specifically, the object and background appearance models are represented by the GMMs. We adopt the EM-like algorithm for object tracking as in [4], but the weight image in our work is computed using the GMM instead of the color histogram. Then a five-layer region model is established based on the object spatial model from the EM-like algorithm. And we develop a five-layer region based graph cuts algorithm for object segmentation. Since the EM-like algorithm estimates the object shape on the weight image which is pixel-wise and the segmentation algorithm extracts the accurate object region during tracking, the proposed approach is able to adapt to scale variation, rotation, and deformation. In summary, there are three contributions of this paper:

- We combine the tracking and segmentation algorithms mentioned above to build a real-time robust tracking system. Experimental results on a variety of challenging sequences demonstrate that this system could not only track previously unseen objects in unconstrained videos but also obtain the accurate object region.

1051-4651/14 $31.00 © 2014 IEEE
DOI 10.1109/ICPR.2014.702

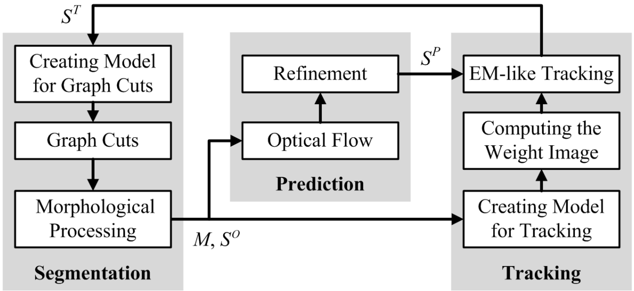


Fig. 1. The block diagram of the proposed approach.

- The EM-like algorithm for color-histogram-based object tracking [4] is modified to deal with the appearance models of the object and background represented by the GMMs which are more efficient in RGB color space.
- We develop a five-layer region based graph cuts algorithm to extract the accurate object region during tracking. It is effective in cluttered background and runs more 10 times as fast as GrabCut [11].

The rest of this paper is organized as follows. In Section II we introduce the proposed approach. In Section III we present the qualitative and quantitative experiment results. In Section IV we report some conclusions.

II. Approach

We propose a new approach that integrates object tracking with object segmentation in a closed loop. Its block diagram is shown in Fig. 1. Based on the object spatial model S^T from the tracking module, the segmentation module extracts the accurate object region M and then computes the corrected object spatial model S^O. In the new frame, the prediction module estimates the motion of the object to produce the predicted object spatial model S^P for tracking. S^P is used as the initial model for the EM-like algorithm. Essentially, S^P, S^T and S^O is respectively represented by an elliptic region, modeled by its center m and covariance matrix C. M is described by a binary image, where the object/background pixels have a value of 1/0. In the first frame, the object region is selected manually or detected using some other algorithms. We obtain S^T used to start the loop by computing the spatial mean and covariance matrix of the pixels in the object region.

A. Segmentation

We develop a five-layer region based graph cuts algorithm to extract the accurate object region. There is a key difference between the widely-used graph cuts based methods [11, 12, 13] and our work. The algorithm in this paper does not need some seeds or an object rectangle marked by the user. A five-layer region model is established based on the object spatial model from the tracking module. We obtain the appearance models of the object and background used for graph cuts based on the five-layer region model.

The five-layer region model $R = \{R_N, R_B, R_U, R_{UO}, R_O\}$ is defined based on the object spatial model S^T, as shown in Fig. 2. S^T is represented by an elliptic region as mentioned above. The four elliptic boundaries between the adjacent regions are determined by Mahalanobis distance, as

$$d(x|S^T) = [(x-m)^T C^{-1}(x-m)]^{\frac{1}{2}} \tag{1}$$

where x is the spatial coordinate of the pixel, m is the center of S^T, and C is the covariance matrix of S^T. The distance thresholds corresponding to each boundary are set to 1, 1.5, 2.5, and 3 respectively. They are kept constant in all of our experiments.

The region R_N is not considered in the following processing. R_O and R_B are fixed as the object and background region respectively. Only the remaining two regions R_{UO} and R_U need to be segmented by graph cuts. We use the GMMs in RGB color space as the appearance models of the object and background. The GMM for the object is established based on the pixels in R_O and R_{UO}. Similarly, the GMM for the background is established based on the pixels in R_B. Since the majority of the pixels in R_{UO} belong to the object, this region is also used to establish the object GMM. In our experiments, this detail helps a lot to extract a more complete object region especially when the shape of the object is irregular. From now on, the steps of constructing the graph and computing the min-cut are similar to [11]. However, the algorithm in this paper needs no iteration. The object region M is obtained as a result, described by a binary image as mentioned above. And some morphological processing, e.g., opening, extraction of connected component, and hole filling, are used to refine the segmentation result. In the end, we compute the spatial mean and covariance matrix of the pixels in the object region to construct the corrected object spatial model S^O.

The comparison of GrabCut [11] and our work in a hand image is shown in Fig. 3. From the results we can see that there are three key benefits of our work:

- It is more effective than GrabCut when the background near the object has the similar color as the object since the appearance models of the object and background established based on the five-layer region model are more precise than those in [11].

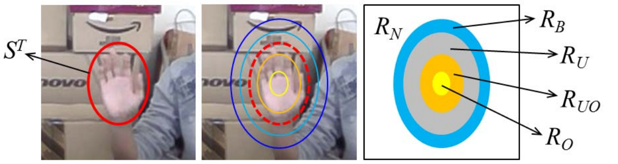


Fig. 2. The five-layer region model.

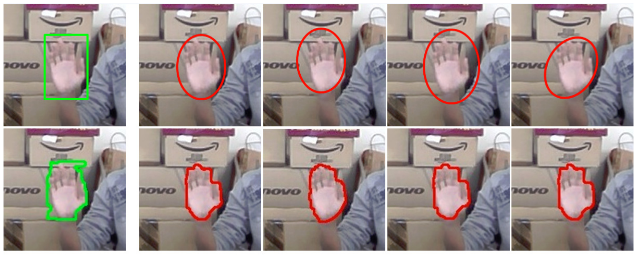

(a) (b)

Fig. 3. The comparison of GrabCut and our work in a hand image. (a) Result of GrabCut. (b) Results of our work given various object spatial models.

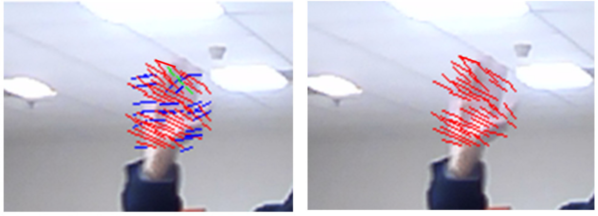
- It runs more than 10 times as fast as GrabCut since only a small number of pixels in R_{UO} and R_U are segmented by non-iterative graph cuts in our work. It takes about 0.019s to extract the hand in Fig. 3, however GrabCut requires 0.221s.
- It is not sensitive to the object spatial model from the tracking module. The shift, scale variation, and rotation of the model have little impact on the performance, as show in Fig. 3b. It just needs to make sure that the real object contour is located in R_{UO} or R_U.

B. Prediction

The prediction module estimates the motion of the object to produce a predicted object spatial model which is used as the initial model for tracking. It is important especially when the motion of the object between two frames is large. First, it could reduce the number of iterations in the EM-like algorithm and thus improve the efficiency. Second, the EM-like algorithm is unable to track the fast moving object without prediction since this algorithm searches for a local maximum mode. For each new frame, the motion of the object is estimated using optical flow [14, 15] as follows.

We compute the optical flow velocities at each pixel in M using the Farnebäck algorithm [15]. This algorithm is found to be a good compromise between accuracy and speed. M is the accurate object region in the previous frame, provided by the segmentation module as mentioned above. If the proportion of the optical flow velocities which have a value of 0 is more than 50%, the object is considered to be motionless. Otherwise, we construct a histogram of the optical flow velocities which are nonzero. These velocities are respectively distributed into 16 orientation bins in 0°–360°. Each bin has a width of 45° and there is an overlap of 22.5° between the adjacent bins. The optical flow velocities voting into the maximum peak of the histogram are considered to be the reliable ones and thus the displacement of the object is estimated using an average of them. We add this displacement to the center of S^O and obtain the predicted object spatial model S^P in the new frame.

The optical flow velocities in the object region computed based on the Farnebäck algorithm are depicted as short lines, as shown in Fig. 4a. It does not perform well in this scenario since the object region is blurred due to the lighting condition and the object motion. We can see that there are a lot of outliers in the results. The reliable ones picked out by our work are shown in Fig. 4b. They are much closer to the real displacement of the object.



(a) (b)

Fig. 4. (a) The optical flow velocities computed by the Farnebäck algorithm. (b) The reliable ones picked out by our work.

C. Tracking

For each new frame, the prediction module estimates the displacement of the object to obtain a predicted object spatial model before tracking. In some scenarios, the predicted object spatial model might already match the object region well. We could remove the tracking module and use S^P instead of S^T in (1). However in most cases, the predicted object spatial model is not precise enough to ensure the effectiveness of segmentation. First, the prediction module does not consider the changes in shape and scale of the object. Second, there are errors in the computation of optical flow especially when the object is non-rigid. Therefore we need to introduce an additional tracking module. The EM-like algorithm for color-histogram-based object tracking [4] is modified to deal with the appearance models of the object and background represented by the GMMs since it is difficult to construct adequate color space histograms with a balance between accuracy and efficiency in practice.

The appearance models of the object and background are established based on the accurate object region M from the segmentation module in the previous frame. First, we compute the upright minimum bounding rectangle enclosing the object region and then a concentric rectangle region expanded by α pixels in width and height is defined as the search range. α is set to 60 in all of our experiments. Second, the GMM for the object F_O is established based on the pixels in the object region and the GMM for the background F_B is established based on the pixels in the background region within the search range.

The weight image, indicating the probability of each pixel belonging to the object, is computed using the log ratio as

$$w(x) = \log\left(\frac{p(y|F_O,x)}{p(y|F_B,x)} + 1\right) \tag{2}$$

where y is the RGB value at the pixel x in the new frame. $p(\cdot)$ is the Gaussian mixture distribution as

$$p(y|F) = \sum_{i=1}^{k} \frac{\pi_i}{\sqrt{|\Sigma_i|}} exp\left\{-\frac{1}{2}(y-\mu_i)^T {\Sigma_i}^{-1}(y-\mu_i)\right\} \tag{3}$$

where k is the number of the components, π_i is the weighting coefficient of the ith component, μ_i is the mean, and Σ_i is the covariance matrix. The pixel with a larger value in the weight image is more likely to belong to the object.

The predicted object spatial model S^P is used as the initial model. We run the EM-like iterations on the weight image to estimate the center and shape of the object in the new frame which determine the object spatial model S^T, as

$$m^{k+1} = \frac{\sum_i x_i w(x_i)\ g(x_i|m^k, C^k)}{\sum_i w(x_i)\ g(x_i|m^k, C^k)} \tag{4}$$

$$C^{k+1} = \beta \frac{\sum_i (x_i - m^k)^T (x_i - m^k) w(x_i)\ g(x_i|m^k, C^k)}{\sum_i w(x_i) g(x_i|m^k, C^k)} \tag{5}$$

where x_i is the spatial coordinate of the pixel within the search range, m is the center of the model, C is the covariance matrix of the model, $g(\cdot)$ is the Gaussian probability distribution, and β is set to 1.5 in all of our experiments.

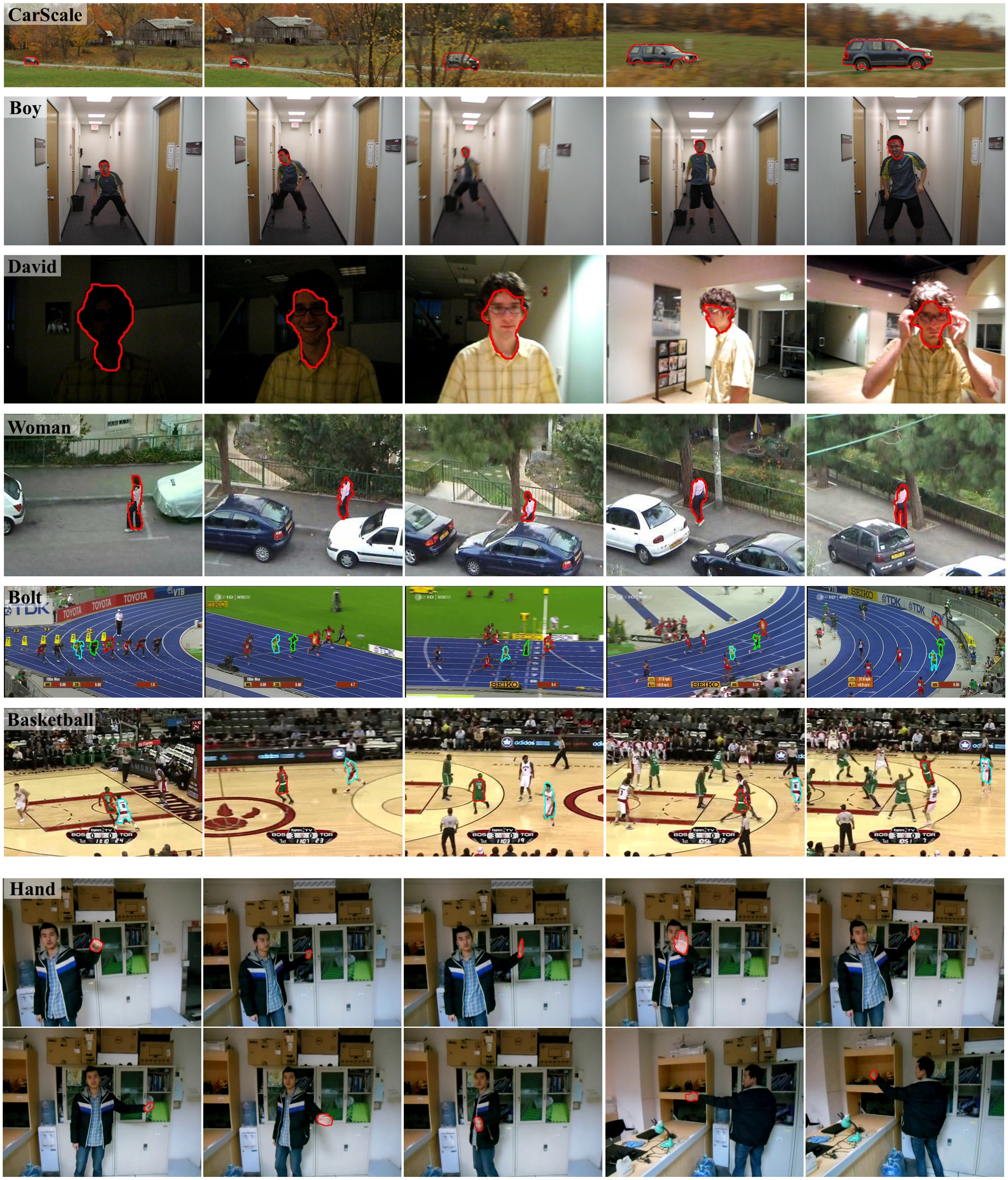


Fig. 5. The test sequences: Carscale, Boy, David, Woman, Bolt, Basketball, and Hand. The contour of the tracked object region in each frame obtained by our approach is drawn on the image. There are three runners being tracked in Sequence Bolt and there two players being tracked in Sequence Basketball.

III. Experiments

The proposed approach was implemented in Visual C++ and validated on a variety of challenging sequences. Due to space limitations, experimental results on seven representative sequences are presented in this section. No parameters were changed from one experiment to the next. The GMM for the object has three components and the GMM for the background has five components. The other parameters were also set and kept constant as mentioned in Section II. A small number of frames from each sequence are shown in Fig. 5. The first six sequences are already available on the Internet and commonly used to evaluate the tracking algorithms. The last one was captured by a moving camera viewing a hand in our laboratory. Note that there are three runners being tracked in sequence Bolt and two players being tracked in sequence Basketball. Table 1 illustrates the main challenges in each sequence.

In all of our experiments, we adopt the following protocol for initialization. In the first frame of each sequence, the object region should be selected manually. As mentioned in Section II, we obtain S^T used to start the loop by computing the spatial mean and covariance matrix of the pixels in the object region. A tracker is initialized in the first frame and tracks the object up to the end except for sequence Basketball. At the beginning of this sequence, the tracked player in green is fully occluded. Therefore we initialize the tracker in the 25th frame.

An almost saturated performance is achieved by our approach on all of these sequences. The contour of the tracked object region in each frame is drawn on the image, as show in Fig. 5. We compared our approach with two relevant algorithms. One is the EM-like tracking algorithm based on the color histogram [4] and the other is the TLD algorithm [9]. Their source codes are available on the Internet and the default parameters were used in our experiments. We have tried to adjust the parameters of the two algorithms to achieve a better performance, but there was no significant improvement. To evaluate the performance of each algorithm, we count the number of successfully tracked frames where the overlap score is larger than 50%. Given the tracked object region and the ground truth object region, the overlap score is defined as in [10]: the ratio of the number of pixels in the intersection to the number of pixels in the union. Table 2 illustrates the comparative results of the two algorithms and our approach. As there are multiple objects being tracked in sequence Bolt and Basketball, the number of their frames is respectively multiplied by the number of the object as shown in the second column of Table 2. From the results we can see that our approach achieved the best performance on all of these sequences, outperforming the two relevant algorithms obviously.

In sequence CarScale, we track the black car running on a country road which undergoes significant scale variation. The algorithm in [4] succeeded in locating the object through the entire sequence, but it failed to exactly estimate the scale of the object in a number of frames. TLD lost the object when it was partially occluded by a tree. This experiment demonstrates that our approach can adapt to scale variation and partial occlusion. And the weight image for the EM-like iterations computed by our approach is more precise than that in [4].

In sequence Boy, we track the face of an energetic boy who is moving quickly and abruptly. The algorithm in [4] is unable to track the fast moving object since the EM-like iteration only searches for a local maximum mode. We introduce a prediction module to estimate the displacement of the object and provide a predicted object spatial model for tracking. It could not only reduce the number of iterations but also overcome the problem of fast motion.

Table 1. The main challenges in each sequence.

Name	MC	SV	R	D	IV	BC	FM	PO
CarScale	yes	yes						yes
Boy	yes	yes	yes				yes	
David	yes	yes	yes		yes			yes
Woman	yes	yes	yes	yes				yes
Bolt	yes	yes	yes	yes		yes	yes	
Basketball	yes	yes	yes	yes	yes	yes		yes
Hand	yes	yes	yes	yes	yes	yes	yes	

MC: moving camera, SV: scale variation, R: rotation, D: deformation, IV: illumination variation, BC: background clutters, FM: fast motion, PO: partial occlusion.

Table 2. Our approach in comparison to two relevant algorithms in [4] and [9] using the number of successfully tracked frames.

Sequence	Frames	No	[4]	[9]	Our
CarScale	252	0	197	154	**252**
Boy	602	0	264	**602**	**602**
David	770	0	33	**770**	**770**
Woman	597	0	65	538	559
Bolt	350×3	0	732	75	**1050**
Basketball	701×2	76	809	366	**1326**
Hand	2280	0	316	1031	**2280**

No: the tracked object disappears from the view of camera.

In sequence David, the appearance of the tracked object changes dramatically due to illumination variation. In [4] the object appearance model is established in the first frame and never updated in runtime. Therefore it easily lost the object. In our approach, the segmentation module extracts the accurate object region during tracking. We can establish the object and background appearance models avoiding introducing errors and update them frame by frame to handle the changes. Without the segmentation module, updating the object appearance model based on the tracking result soon causes drift. This experiment demonstrates that our approach is able to adapt to illumination variation. And object tracking combined with segmentation can significantly enhance its performance.

In sequence Woman, Bolt, and Basketball, we track several people respectively walking on the street, running the race, or playing basketball. The rotation and deformation of the object occur frequently in these scenarios. TLD does not perform well when the object undergoes out-of-plane rotation or significant deformation. In our approach, the EM-like algorithm estimates the object shape on the weight image which is pixel-wise and the segmentation module extracts the accurate object region during tracking. Therefore it is able to handle these difficult scenarios as well. However at the end of sequence Woman, our approach lost the object due to the zoom-in and shake of the camera. The camera shake caused a large displacement of the object. Unfortunately, the prediction module failed to produce an appropriate result.

In sequence Hand, the background near the object (e.g. face and yellow bookcase) has the similar color as the hand being tracked. As the object and background appearance models used for graph cuts are established based on the five-layer region model, they are more precise than those in [11]. Therefore the proposed segmentation algorithm can extract the object region exactly even in cluttered background. A detailed comparison of this algorithm and GrabCut [11] is presented in Section II.

Experimental results demonstrate that our approach is able to track previously unseen objects in unconstrained videos captured by a possibly moving camera. It runs at about 14 frames per second on a PC with Inter Core2 CPU (2.66GHz) without optimization. Moreover, the accurate object region is extracted by the segmentation module during tracking. In a large number of applications (e.g. gesture recognition), the accurate object region rather than a bounding box is crucial for the following processing.

IV. Conclusion

In this paper, we proposed a new approach that integrates object tracking with object segmentation in a closed loop. And we demonstrated that it could not only track previously unseen objects in unconstrained videos captured by a possibly moving camera but also extract the accurate object region during tracking. The appearance models of the object and background are represented by the GMMs. In the tracking module, a weight image indicating the probability of each pixel belonging to the object is computed based on the GMMs. We run the EM-like iterations on the weight image to estimate the object position and shape in the new frame which determine the object spatial model. In the segmentation module, a five-layer region model is established based on the object spatial model from tracking. We obtain the object and background GMMs used for graph cuts based on the five-layer region model. Since the models established in this way are precise, the segmentation algorithm is effective even in cluttered background. The accurate object region is extracted by the segmentation module. Then we can establish the appearance models of the object and background avoiding introducing errors and update them frame by frame without the problem of drift. The refined and adaptive models lead to robust tracking in return. The proposed approach is also able to adapt to scale variation, rotation, and deformation since the EM-like iteration estimates the object shape on the weight image which is pixel-wise and the segmentation module extracts the accurate object region during tracking. Moreover, the motion of the object is estimated to produce a predicted object spatial model for tracking. The EM-like iteration only searches for a local maximum mode. Therefore it is unable to track the fast moving object without the prediction module.

Acknowledgment

We thank the reviewers for valuable comments. This work is supported by the National Natural Science Foundation of China under Grant Nos. 61271328 and 61105014.

References

[1] A. Yilmaz, O. Javed, and M. Shah, “Object tracking: a survey,” ACM Computing Surveys, vol. 38, no. 4, pp. 1-45, 2006.

[2] D. Comaniciu, V. Ramesh, and P. Meer, “Real-time tracking of non-rigid objects using mean shift,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 142-149, 2000.

[3] D. Comaniciu, V. Ramesh, and P. Meer, “Kernel-based object tracking,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 25, no. 5, pp. 564-577, 2003.

[4] Z. Zivkovic and B. Kröse, “An EM-like algorithm for color-histogram-based object tracking,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, vo1. 1, pp. 798-803, 2004.

[5] P. Chockalingam, N. Pradeep, S. Birchfield, “Adaptive fragments-based tracking of non-rigid objects using level sets,” Proc. IEEE Conf. Computer Vision, pp. 1530-1537, 2009.

[6] S. Avidan, “Ensemble tracking,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 29, no. 2, 2007.

[7] Z. Kalal, K. Mikolajczyk, and J. Matas, “Forward-backward error: automatic detection of tracking failures,” Proc. IEEE Conf. Pattern Recognition, pp. 23-26, 2010.

[8] Z. Kalal, J. Matas, and K. Mikolajczyk, “P-N learning: bootstrapping binary classifiers by structural constraints,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 49-56, 2010.

[9] Z. Kalal, K. Mikolajczyk, and J. Matas, “Tracking-learning-detection,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 34, no. 7, pp. 1409-1422, 2012.

[10] Y. Wu, J. Lim, M.H. Yang, “Online object tracking: a benchmark,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 2411-2418, 2013.

[11] C. Rother, V. Kolmogorov, and A. Blake, “GrabCut: interactive foreground extraction using iterated graph cuts,” ACM Trans. Graph., vol. 23, no. 3, pp. 309-314, 2004.

[12] Y. Boykov and M.-P. Jolly, “Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images,” Proc. IEEE Conf. Computer Vision, vol. I, pp. 105-112, 2001.

[13] Y. Boykov and G. Funka-lea, “Graph cuts and efficient N-D image segmentation,” International Journal of Computer Vision, vol. 70, no. 2, pp. 109-131, 2006.

[14] J.L. Barron, D.J. Fleet, and S.S. Beauchemin, “System and experiment performance of optical flow techniques,” International Journal of Computer Vision, vol. 12, no. 1, pp. 43-77, 1994.

[15] G. Farnebäck, “Two-frame motion estimation based on polynomial expansion,” Lecture Notes in Computer Science, vol. 2749, pp. 363-370, 2003.