Robust and Adaptive Object Tracking via Correspondence Clustering

Bo WU†,††a, Member, Yurui XIE†, and Wang LUO†††, Nonmembers

SUMMARY We propose a new visual tracking method, where the target appearance is represented by combining color distribution and keypoints. Firstly, the object is localized via a keypoint-based tracking and matching strategy, where a new clustering method is presented to remove outliers. Secondly, the tracking confidence is evaluated by the color template. According to the tracking confidence, the local and global keypoints matching can be performed adaptively. Finally, we propose a target appearance update method in which the new appearance can be learned and added to the target model. The proposed tracker is compared with five state-of-the-art tracking methods on a recent benchmark dataset. Both qualitative and quantitative evaluations show that our method has favorable performance.

key words: appearance model, correspondence clustering, model update, visual tracking

1. Introduction

Visual object tracking is a hard problem due to the complex object appearance changes caused by factors such as scale variations, occlusion, deformation, and motion blur [1],[2]. During the last decade, many trackers have been proposed, which can be roughly categorized into holistic template-based trackers and local feature-based trackers. The holistic approaches build a template of the target’s appearance globally. Due to the advantages of invariance to rotation and scaling, a color histogram is widely used for visual appearance. In [3], a color histogram was embedded into a particle filter for object tracking. Zhao et al. [4] converted the tracking problem into that of comparing the color distributions across frames. However, the global visual representation has difficulty dealing with occlusion. Different from the global visual representation, local feature-based approaches mainly utilize a set of good-to-track keypoints to encode the object appearance [1]. By estimating a similarity transform, the keypoint-based tracking method can effectively deal with the problem of the scale change and rotation [5]. Nebehay et al. [6] proposed a method in which the object is tracked via consensus-based voting of keypoints. The tracker works better in most of the cases. But due to the keypoint model’s inability to learn new object appearance changes, the drift problem arises under the large object appearance change and low image quality.

This paper proposes a tracking method which joints keypoints and a color histogram to model object appearance. The object is localized by tracking and matching keypoints, where a new clustering method is presented to remove the outliers automatically. The color template is used to evaluate the tracking confidence. According to the tracking confidence, the local and global keypoints matching is performed adaptively, which can reduce the computation cost effectively. Finally, we propose a target model update method. The new appearance can be learned to improve the performance of tracking. The block diagram of the tracker is illustrated in Fig. 1 and detailed in the next section.

2. Tracking Algorithm

Given a bounding box defining an object of interest in the...
To estimate the object's state, we maintain an active set of keypoints: \( \mathcal{P}_A^t = \{(p_{i}^t, \bar{d}_i)\}_{i=1}^{N_t} \) in each frame \( I_t \), where \( p_{i}^t \) is the corresponding point of \( p_i \) in \( I_1 \).

To obtain \( p_i^t \) in \( I_t \), we compute the displacement of each keypoint in \( \mathcal{P}_A^{t-1} \) from \( I_{t-1} \) to \( I_t \) by employing the Lucas-Kanade (LK) method [8]. For \( t = 2 \), \( \mathcal{P}_A^1 \) is initialized by \( \mathcal{T} \). Additionally, we employ a forward-backward error measure [9] to improve robustness.

2.2 Outliers Filtering via Clustering

In active set \( \mathcal{P}_A^t \), the majority of correspondence can be trusted in spite of a fraction of outlier existing. Based on this assumption, the scale \( s_t \) and rotation angle \( \theta_t \) can be easily estimated [6]. The scale \( s_t \) is obtained by

\[
s_t = \text{med} \left( \frac{\|p_i^t - p_j^t\|}{\|p_i^t - p_j^t\|} \right),
\]

\( i \neq j \) for the \( i \)th point in \( I_t \), \( p_i^t \) is the corresponding point of \( p_i \) in \( I_1 \).

The rotation angle can be estimated as \( \theta_t = \text{med}(\text{atan2}(p_i^t - p_j^t), i \neq j), \)

where \text{med} denotes the median. \( \|\| \) denotes L2 norm. Atan2 is the arctangent function which computes the angle in the appropriate quadrant.

Next, the object center \( c_t \) can be estimated by using a voting scheme on correct correspondence set. In order to identify and remove the erroneous correspondence in \( \mathcal{P}_A^t \), we present a simple density-based clustering method. Firstly, the dissimilarity measure \( D \) between correspondences \( \mathcal{P}_A^t(i) \) and \( \mathcal{P}_A^t(j) \) is defined as:

\[
D(i, j) = \| \mathbf{v}_j - \mathbf{v}_i \|,
\]

with

\[
\mathbf{v}_j = \bar{p}_k^r - H \bar{p}_k^r, k = \{i, j\}
\]

where \( \bar{p}_k^r \) is the relative location of \( p_k^r \) with reference to \( c_1 \), i.e. \( \bar{p}_k^r = p_k^r - c_1 \), \( m = \{1, t\} \). \( H_t \) is a estimated similarity transform which includes parameters \( s_t \) and \( \theta_t \). Note that if we transform \( \bar{p}_1^t \) and \( \bar{p}_j^t \) by a common displacement vector, the values of \( \mathbf{v}_j^t \) and \( \mathbf{v}_i^t \) will change. But the relative distance \( D \) is invariant.

The rotation angle can be estimated as

\[
\theta_t = \text{atan2}(p_i^t - p_j^t)
\]

where \text{med} denotes the standard deviation of Gaussian Kernel.

To compute the quantity \( \delta_t \), we first sort the local density \( \rho_{i} \) in descending order: \( \rho_{k_1} \geq \rho_{k_2} \geq \ldots \geq \rho_{k_{N_t}} \). The indices are denoted as \( \{k_{i} \}_{i=1}^{N_t} \). Then the quantity \( \delta_{k_i} \) can be computed as

\[
\delta_{k_i} = \left\{ \begin{array}{ll}
\min_{j\neq i} |D(k_i, k_j)|, & i \geq 2 \\
\max_{j}|D(k_i, k_j)|, & i = 1,
\end{array} \right.
\]

In some cases, several points near one cluster center have the same highest density \( \rho_{i} \). According to [10], these points might be assigned large distance \( \delta_{i} \) and then are all selected as cluster centers. As a result, one cluster is wrongly divided into multiple clusters. Sorting \( \rho_{i} \) prior to compute \( \delta_{i} \) can avoid the problem. According to Eq. (6), when the \( k \)th point is local or global maxima in the density, \( \delta_{k} \) will much larger than the typical nearest neighbor distance.

In [10], the cluster centers need to be selected by user subjectivity. As in our problem, there is exactly one object. So we can automatically select the point with highest density \( \rho_{i} \) as cluster center. According to the prior sorting of local density \( \{\rho_{i}\}_{i=1}^{N_t} \), we select the point with maximum \( \rho_{i} \) as single initial cluster center and set cluster label \( C_i = i \). Then by the descending order of \( \rho \), the remaining data points are put into the nearest cluster with higher local density. The cluster propagation is performed as

\[
C_j = \left\{ \begin{array}{ll}
C_i, & \rho_j \geq \rho_i, \delta_j < \delta_T \\
\text{outlier}, & \text{otherwise},
\end{array} \right.
\]

where \( \delta_T \) is a predefined threshold. As for the problem of single object tracking, only the cluster with highest density is associated to object. Sparse outliers with low density \( \rho \) and high distance \( \delta \) will be filtered by the procedure of clustering effectively.

Once the correspondence clustering is finished, we update the active set \( \mathcal{P}_A^t \) from which the error correspondences are removed. The object center can be estimated by averaging

\[
\rho_{i} = \sum_j \exp \left( \frac{-D^2(i, j)}{d_c^2} \right),
\]

where parameter \( d_c \) denotes the standard deviation of Gaussian Kernel.
where \( \text{histogram} \) is given the estimated object bounding box, the target color is excluded. We use Bhattacharyya coefficient to calculate similarity between \( \mathbf{f}_i \) and \( \mathbf{f}_j \):

\[
Bc(\mathbf{f}_i, \mathbf{f}_j) = \sum_{a=1}^{m} \sqrt{f_{ia} \cdot f_{ja}},
\]

A larger \( Bc \) illustrates that the tracking is successful and without occlusion. Thus, we use a binary variable \( F \) to represent the tracking confidence:

\[
F = \begin{cases} 
1, & Bc(\mathbf{f}_i, \mathbf{f}_j) > \alpha \\
0, & \text{otherwise},
\end{cases}
\]

where \( \alpha \) is a predefined threshold.

In addition to keypoints tracking, we use keypoints matching to establish correspondences. If the tracking drift arise or the object is occluded \( (F = 0) \), the global keypoints matching is performed. We detect keypoints in current frame and match with keypoints from \( I_1 \). Those keypoints matched with \( T \) are denoted as \( \mathcal{P}_A^t \).

If the tracking result is confident \( (F = 1) \), the local keypoints matching is performed. We detect keypoints in a search region which is the double object box in the current frame. For each detected keypoint \( \mathbf{p}_j^t \), we perform match on the subset

\[
\{ \mathbf{p}_j^t \} | (H_1^{-1}(\mathbf{p}_j^t - \mathbf{c}_i) + \mathbf{c}_1) | < \beta,
\]

where \( (H_1^{-1}(\mathbf{p}_j^t - \mathbf{c}_i) + \mathbf{c}_1) \) is projection point of \( \mathbf{p}_j^t \) in \( I_1 \).

Keypoints that match to background descriptors in \( I_1 \) are excluded. We use \( \mathcal{P}_ML \) to denote the matched keypoints. Those unmatched keypoints but within the estimated object bounding box are denoted as \( \mathcal{P}_N \), which will be used to update the target model. Comparing with global matching, local matching can reduce the computation cost effectively.

2.4 Object Appearance Update

Object appearance maybe change drastically due to the illumination variance, severe scale changes, motion blur, and so on. To track the object robustly, it is crucial to learn the new appearance model on the fly.

If \( F = 1 \), the color appearance is updated as

\[
\mathbf{f}_1 = (1 - \gamma)\mathbf{f}_1 + \gamma \mathbf{f}_i,
\]

where \( \gamma \) is learning rate.

As for keypoints, we update the active set \( \mathcal{P}_A^t \) on the fly. Since outliers are filtered via clustering, the number of keypoints in \( \mathcal{P}_A^t \) is reduced progressively. This will decrease the robustness of vote. Therefore, we need to add new valid keypoints into \( \mathcal{P}_A^t \). On the other hand, the active set \( \mathcal{P}_A^t \) can not be updated frequently considering the computation cost. Therefore, the active set can be updated as

\[
\mathcal{P}_A^{t+1} = \begin{cases}
\mathcal{P}_A^t \cup \mathcal{P}_ML^t \cup \mathcal{P}_N^t, & F = 1, N_A < N_L \\
\mathcal{P}_A^t \cup \mathcal{P}_ML^t, & F = 1, N_A \geq N_L \\
\mathcal{P}_A^{t+1} \cup \mathcal{P}_MG^t, & F = 0,
\end{cases}
\]

where \( N_L \) is the predefined threshold. The new keypoints \( \mathcal{P}_A^t \) are added to active set only when \( F = 1 \) and the cardinality of \( \mathcal{P}_A^t \) is lower than \( N_L \). Given a point \( \mathbf{p}_j^t \) in \( \mathcal{P}_A^t \), its correspondence point in \( I_1 \) can be computed as:

\[
\mathbf{p}_j^t = H_1^{-1}(\mathbf{p}_j^t - \mathbf{c}_i) + \mathbf{c}_1.
\]

3. Experiments

We evaluate the proposed tracking method with five state-of-the-art methods on online tracking benchmark dataset [11]. The selected trackers contain the top four ranking trackers [9], [12]–[14] and a keypoint-based tracker [6]. The dataset comprises 50 videos with many challenging aspects such as illumination variation, scale variation, occlusion, deformation and so on.

For our experiments, the parameters mentioned in Sect. 2 are empirically set as \( \delta_T = 6, \alpha = 0.9, \beta = 20, N_L = 100, \) learning rate \( \gamma = 0.05. \) In order to set \( d_c \), we first sort all pairs of the dissimilarities \( D(i, j) \). Then, the \( (N + 0.02) \)-th smallest element is selected as \( d_c \), where \( N = N_A(N_A - 1)/2 \).

3.1 Qualitative Evaluation

As space is limited, we only select six sequences from the benchmark dataset for visualization and analysis. These sequences have scale variances, illumination change, and partial or full occlusion. Some sampled tracking results of different trackers are showed in Fig. 2. Note that our tracking...
result is rotation box. For comparing with other methods consistently, we transform the rotation box into axis-aligned bounding box.

Figure 2 (a), (b) and (c) illustrate three sequences with large scale variations. Tracking results demonstrate that our tracker can well handle the scale variation. This is due to the fact that our tracker can effectively filter out outliers via clustering, which lead to a accurate estimation of similarity transform. In Fig. 2 (a) car4 and (e) skating1, the object undergo significant illumination variation. Our tracker can track the objects successfully, whereas most of the other trackers are suffer from tracking drift. This can be mainly attributed to the robust model update strategy of our tracker. In Fig. 2 (d) liquor and (f) woman, the objects undergo partial or full occlusion. Tracking results show that the proposed tracker can also perform well in these cases.

3.2 Quantitative Evaluation

We perform one-pass evaluation (OPE) on the benchmark to evaluate the proposed method quantitatively. The precision rate (PR) plot and success rate (SR) plot are showed in Fig. 3. PR plot shows the percentage of frames where the distance between the tracked locations and the ground truth locations are within a given threshold. SR plot shows the ratios of successful frames whose intersection over union overlap with the ground truth annotation is over a thresholds varied from 0 to 1 [11]. To rank the performance, the precision obtained at 20 pixels is used as PR. For the success plot, the area under curve is used as SR. Ours-NU denotes our method without appearance updating. CMT. After adding the module of appearance updating, our method outperforms the second best tracker CMT by 8% in PR and SCM by 7% in SR.

We assess the speed of the considered trackers on a 3.2 GHz DualCore PC. The average FPS of [6], [9], [12]–[14] and ours are 11, 26, 15, 14, 0.5 and 9 respectively. Note that our method and SCM are implemented in Matlab while others are implemented in C++.

4. Conclusion

This paper presents a new tracking method in which keypoints and a color histogram are jointed to represent the object appearance. The object is localized by tracking and matching keypoints. The outliers are filtered by a clustering method by which the keypoints relevant to object are selected automatically. The color template is used to evaluate the tracking confidence which is then used to select the local or global keypoints matching adaptively. Finally, we propose a model update method. The new appearance can be learned to improve the performance of tracking.

References

[1] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A.V. D. Hengel, “A survey of appearance models in visual object tracking,” ACM transactions on Intelligent Systems and Technology (TIST), vol.4, no.4, pp.58:1–58:48, 2013.
[2] H. Li and K.N. Ngan, “Saliency model-based face segmentation and tracking in head-and-shoulder video sequences,” J. Vis. Comm. and Image Represent., vol.19, no.5, pp.320–333, 2008.
[3] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, “Color-based probabilistic tracking,” in ECCV, vol.2350, pp.661–675, Springer, 2002.
[4] Q. Zhao, Z. Yang, and H. Tao, “Differential earth mover’s distance with its applications to visual tracking,” IEEE Trans. Patt. Anal. Mach. Intell., vol.32, no.2, pp.274–287, 2010.
[5] F. Pernici and A.D. Bimbo, “Object tracking by oversampling local features,” IEEE Trans. Patt. Anal. Mach. Intell., vol.36, no.12, pp.2538–2551, 2014.
[6] G. Nebehay and R. Pflugfelder, “Clustering of static-adaptive correspondences for deformable object tracking,” CVPR, pp.2784–2791, 2015.
[7] S. Leutenegger, M. Chli, and R.Y. Siegwart, “Brisk: Binary robust invariant scalable keypoints,” ICCV, pp.2548–2555, IEEE, 2011.
[8] B.D. Lucas, T. Kanade, et al., “An iterative image registration technique with an application to stereo vision...,” IJCAI, pp.674–679, 1981.
[9] Z. Kalal, K. Mikolajczyk, and J. Matas, “Tracking-learning-detection,” IEEE Trans. Patt. Anal. Mach. Intell., vol.34, no.7, pp.1409–1422, 2012.
[10] A. Rodriguez and A. Laio, “Clustering by fast search and find of density peaks,” Science, vol.340, no.6191, pp.1492–1496, 2014.
[11] Y. Wu, J. Lim, and M.-H. Yang, “Online object tracking: A benchmark,” CVPR, pp.2411–2418, IEEE, 2013.
[12] S. Hare, A. Saffari, and P.H.S. Torr, “Struck: Structured output tracking with kernels,” ICCV, pp.263–270, IEEE, 2011.
[13] T.B. Dinh, N. Vo, and G. Medioni, “Context tracker: Exploring supporters and distractors in unconstrained environments,” CVPR, pp.1177–1184, IEEE, 2011.
[14] W. Zhong, H. Lu, and M.-H. Yang, “Robust object tracking via sparse collaborative appearance model,” IEEE Trans. Image Process., vol.23, no.5, pp.2356–2368, 2014.