Kernel-Based On-Line Object Tracking Combining both Local Description and Global Representation*

Quan MIAO\(^{(a)}\), Student Member, Guijin WANG\(^{(b)}\), Member, and Xinggang LIN\(^{(c)}\), Nonmember

SUMMARY This paper proposes a novel method for object tracking by combining local feature and global template-based methods. The proposed algorithm consists of two stages from coarse to fine. The first stage applies on-line classifiers to match the corresponding keypoints between the input frame and the reference frame. Thus a rough motion parameter can be estimated using RANSAC. The second stage employs kernel-based global representation in successive frames to refine the motion parameter. In addition, we use the kernel weight obtained during the second stage to guide the on-line learning process of the keypoints’ description. Experimental results demonstrate the effectiveness of the proposed technique.

key words: object tracking, keypoint matching, kernel function, classifier updating

1. Introduction

Object tracking is a crucial module in many image processing and computer vision applications, ranging from visual surveillance to human-computer interfaces. The goal of tracking is to automatically estimate the motion parameter of the same object through a video sequence. Although many methods have been proposed for different applications, developing a robust tracking approach under real-world conditions is still a challenging task. The difficulties of object tracking include complicated appearance variations, illumination change, partial occlusions and cluttered scenes.

Recent tracking algorithms can be classified into two categories: global template-based tracker and local feature-based trackers. In the case of global trackers [1]–[3], the basic idea is to build or learn a template of the target object first and then find the matching region as closely as possible in the remaining frames. Despite the high matching accuracy, these approaches tend to fail once the target undergoes large appearance changes or complex geometric transformations. The reason is that such approaches are based on an underlying assumption that the shape of the target in consecutive frames should be similar. As for feature-based trackers, the strategy is to extract reliable local features, build invariant descriptions and establish feature correspondences to estimate the motion parameters [4], [5]. The main advantage of the feature-based method is its relative robustness against changes in the object.

In [4], randomized trees are used to discriminate keypoints. The disadvantage is that the object must be trained a priori in the off-line phase, which will take a considerable amount of time to collect sufficient image samples. Tracking will fail if a certain appearance change of the object is not covered in the training phase. An extension of this approach is presented in [5] which employs the on-line boosting technique [6]. However, the resulted tracker typically operates on the premise that the appearance of the target should not change drastically over successive frames. The reason is that the keypoints are detected using Harris corner, making the classifiers sensitive to affine transformation and viewpoint change. To overcome the problem, we propose a new tracking scheme by employing the scale and rotation information of local feature to guide the on-line classifier learning [7], [8]. The tracker achieves adaptive tracking under aforementioned complex appearance changes. Despite the satisfying tracking performance, the tracker occasionally fails to observe the object motion, because slight target locating errors accumulated over consecutive frames will cause the appearance model to be updated with a sub-optimal positive sample. Overtime the discriminative power of each classifier will be degraded, causing drift. To address the drift problem, Morency et al. proposed keyframe tracking [9] for pose estimation which belongs to non-rigid object tracking.

This paper proposes a kernel-based coarse-to-fine method combining feature and template matching. In the first stage, instead of consecutive frame-to-frame tracking, we perform classifier-based keypoint matching between the current object and the original one in the reference frame. Thus the previous accumulation of tracking error becomes a nonentity. With the help of RANSAC, a rough motion parameter can be estimated. The second stage applies the detected feature points between consecutive frames in the global template-based algorithm, to refine the current object region. Furthermore, we introduce the kernel function which penalizes the error between the coordinate of each feature point in the input frame and the sub-pixel location of the transformed correspondence, to give those reliable pairs more weight. Finally, the kernel is used to guide the on-line learning process to make the updated classifiers more adaptive. Experimental results show the satisfying tracking performance of the proposed algorithm on several challenging video sequences.
2. Motion Model

In our scheme, tracking is performed by transforming the object region using estimated motion parameters, based on the matching candidates (see Fig. 1). Suppose \( \mathbf{x} \) and \( \tilde{\mathbf{x}} \) to be corresponding object keypoint in two images, we have the following equation:

\[
\tilde{\mathbf{x}} = W(\mathbf{x}; \mathbf{h}),
\]

where \( W(\mathbf{x}; \mathbf{h}) = (W_1(\mathbf{x}; \mathbf{h}) W_2(\mathbf{x}; \mathbf{h}))^T \) is the mapping function and \( \mathbf{h} = (h_1, \ldots, h_6)^T \) is the motion parameter. In this paper, we employ the homography as the motion model, which is shown as follows:

\[
W(\mathbf{x}; \mathbf{h}) = \begin{pmatrix}
(1+h_1)x+h_2y+h_3 \\
h_4x+(1+h_5)y+h_6
\end{pmatrix}
\]

(2)

3. Proposed Algorithm

Given location of the object in the first frame (i.e., reference frame or keyframe), we detect keypoints within the object region. When a new image enters, we extract its keypoints and perform classifier-based keypoint matching between this input frame and the object in keyframe. After homography estimation using RANSAC, we introduce two step motion estimation similar to [10] used for pose-invariant facial expression recognition. First, a rough object region can be obtained for the input frame. Then we refine the region by dealing with the detected feature points between consecutive frames. Finally, we update the on-line classifiers for further tracking. In the remainder of this section we will describe the algorithm shown in Fig. 2.

3.1 First Stage

During initialization in the first frame, we build M classifiers \( \{C_1, C_2, \ldots, C_M\} \), each corresponding to an object keypoint \( \{b_1, b_2, \ldots, b_M\} \). The strong boosted classifier \( C \) is composed of J selectors \( h^{rel}_j \) and holds a binary weak classifier pool from which the training procedure selects the ones with the minimal estimated error. It will predict the matching confidence measure of an unknown point \( \mathbf{x} \) by:

\[
C(\mathbf{x}) = \sum_{j=1}^{J} \alpha_j \cdot h^{rel}_j(\mathbf{x}) \sum_{j=1}^{J} \alpha_j,
\]

(3)

where the value \( \text{conf}(\bullet) \) denotes the confidence measure. Given the keypoints set \( \Upsilon = \{\gamma_1, \gamma_2, \ldots, \gamma_N\} \) detected in the new frame, we employ the classifier \( C_m \) to compute the confidence measure of each \( \gamma_n \), and record point \( \varphi_m \) by:

\[
\varphi_m = \arg \max_{\gamma \in \Upsilon} C_m(\gamma_n).
\]

(4)

Similarly, the set of matching candidates \( \Phi = \{\varphi_1, \varphi_2, \ldots, \varphi_M\} \) is established by using the same operation on \( \Upsilon \).

In this paper, we use the SURF detector [11] to extract the feature points because of its high detection accuracy and invariance to rotation and scale changes. Furthermore, we employ the scale information and the dominant orientation of the SURF feature to guide the classifier learning process. This leads to a series of scale- and rotation-invariant classifiers that are able to handle large appearance changes between frames. For more details on how to construct the classifiers, refer to [8].

Assuming the target object has been successfully identified in frame \( t - 1 \), we perform RANSAC on \( \Phi \) to estimate the homography \( h_{t-1,ref} \) in frame \( t \). Thus a rough object region is obtained. Finally, we solve \( W(\mathbf{x}; \mathbf{h}_{t-1,ref}) = W(W(\mathbf{x}; \mathbf{h}_{t-1,ref}); \mathbf{h}_{t-1}) \) to estimate \( \mathbf{h}_{t-1} \), which is considered as the initial motion parameter for the second step. By performing keypoint matching between original object and the current one instead of consecutive frames, this first step effectively eliminates the aforementioned accumulated tracking error to attain stable tracking.

3.2 Second Stage

Once the initial motion information between views is available, we apply the feature points in global representation to refine the motion parameter. For a certain point \( \mathbf{x} \), the relationship is defined as:

\[
I_{t-1}(\mathbf{x}) \approx I_t(W(\mathbf{x}; \mathbf{h}_{t-1} + \Delta \mathbf{h}_{t-1})),
\]

(5)

where \( I_t(\mathbf{x}) \) and \( I_{t-1}(\mathbf{x}) \) respectively denote the image in frame \( t \) and \( t - 1 \); \( \Delta \mathbf{h}_{t-1} \) denotes increment to \( \mathbf{h}_{t-1} \). Then the expression in Eq. (5) is linearized by a first order Taylor expansion on \( I_t(W(\mathbf{x}; \mathbf{h}_{t-1} + \Delta \mathbf{h}_{t-1})) \):

\[
I_{t-1}(\mathbf{x}) \approx I_t(W(\mathbf{x}; \mathbf{h}_{t-1})) + \nabla I_t \frac{\partial W}{\partial \mathbf{h}_{t-1}} \Delta \mathbf{h}_{t-1}.
\]

(6)

Generally, the repeatability of different SURF features often changes under different geometric and photometric transformations. For example, some points may be more adaptive
Capturing this idea, we focus on minimizing the following to indicate how adequate each feature point is for matching. Thus we introduce a kernel function to scale changes while others are more repeatable during illumination variations. Therefore we introduce a kernel function to scale changes while others are more repeatable during illumination variations. Thus we introduce a kernel function to scale changes while others are more repeatable during illumination variations. Therefore we introduce a kernel function to scale changes while others are more repeatable during illumination variations.

\[
\min_{\Delta \mathbf{h}_{t-1}, \gamma \in \text{feature}} \sum_{x \in P_t} \left[ I_{t-1}(x) - I_t(W(x; \mathbf{h}_{t-1})) \right] - \nabla I_t \frac{\partial W}{\partial \mathbf{h}_{t-1}} \Delta \mathbf{h}_{t-1}^2 K(\gamma - W(\gamma; \mathbf{h}_{t-1})),
\]

where \( \gamma \) is defined in frame \( t \), denoting the corresponding point of object keypoint \( \gamma \) in frame \( t-1 \). \( P_t \) denotes the influence region of \( \gamma \), which is a round area with radius \( 2.5 s \). \( s \) indicates the scale of \( \gamma \). The keypoint with large \( s \) is detected as the local extremum in large scale space, so it has a relatively large influence region. \( K(\bullet) \) is the 2-D kernel function, which gives those feature points that are well matched in luminance and location more weight. The motion parameter and the weight are estimated simultaneously and iteratively. First the motion parameter is solved using a particular kernel weight; then the kernel weight is recalculated using the motion parameter in the prior step. Specifically, we employ the gradient descent method [12] to estimate \( \Delta \mathbf{h}_{t-1} \) in Eq. (7). Hence the motion parameter is refined to attain accurate object tracking.

3.3 Updating

Once tracking is completed, the classifiers describing the tracked object will be updated by taking matches as positive samples and other keypoints as negative. As is shown in Fig. 3, each selector \( h^{rel}_j \) re-selects the best weak classifier and the corresponding voting weight \( \alpha_j \) is updated. For more details, refer to [6].

During updating, how to assign each sample the importance weight is a crucial problem. This paper non-uniformly updates the classifiers according to their kernel weight obtained in the second stage. When a classifier has a high kernel weight, the importance weight of its samples for updating is likewise high. This way, we emphasize the feature points that are more repeatable in the current successive frames, making the updated classifier more reliable.

4. Experimental Results

We now present the experimental results of applying our algorithm on two image sequences with the size of \( 640 \times 480 \). Both experiments are implemented in C++ code on a PC with 2.2 GHz CPU and 2 GB RAM. Each classifier consists of 20 selectors selected from a global weak classifier pool holding 250 weak hypotheses. The processing speed is about 8 fps in average.

As a first attempt, we concern the tracking stability of the proposed method. For comparison, we implemented Grabner’s tracker [5]. Meanwhile, our previous feature-based tracker proposed in [8] is used. Associated parameters and thresholds of these two methods are fixed to be the same as we use in this paper.

Figure 4 shows the performance of tracking a bottle under complex changes. It can be observed that Grabner’s tracker fails once appearance change becomes large, because the keypoints are detected using Harris corner very sensitive to scale and rotation changes. Afterwards Grabner’s tracker re-tracks the object, whose region is distorted because of the tracking errors accumulated over frames. In addition, drifting has seriously influenced the previous feature-based tracker. Although the tracker can handle the complex changes (e.g., the 70th frame) in the beginning, the identified region in the 315th frame becomes completely out of shape. In contrast, the proposed method successfully tracks the object, experiencing 45-degree rotation changes (the 315th frame) and large scale changes (the 70th frame). Although the object changes between keyframe and the current frame may be complex over frames, the subsequent second stage makes the tracker adaptive to the latest appearance changes.

Furthermore, we focus on validating the advantage of employing kernel function. For comparison, we implemented another tracking method in which the first frame is considered as the reference frame and correspondences are established between the keypoints in the defined object region of the reference frame and those in the input frame. The best candidate match is defined as the one with the minimum Euclidean distance for the SURF descriptor vector. We call this second approach the SURF-based tracker.

Figure 5 shows the performance of tracking a book cover we open and close to generate a viewpoint change. In addition, some partial occlusion is mixed. It can be observed that the SURF-based method fails to track the object when the partial occlusion or viewpoint change becomes severe, because the detected features’ repeatability limited. As
for the proposed method, the combination of local feature and global representation using kernel helps to improve the matching repeatability and generate accurate object region. Meanwhile, the corresponding kernel-based on-line learning mechanism weighted by the second stage emphasizes those points more repeatable and reliable with respect to the current occlusion or viewpoint change. All these ingredients contribute to the satisfying tracking performance.

Figure 6 shows the number of correct matches certified by RANSAC over time. Tracking loss (the percentage is below 25%) occurs constantly using SURF-based method because of significant appearance change. In contrast, the proposed tracker always achieves successful tracking. Moreover, the proposed method keeps establishing more correct matches than the SURF-based method, validating the superiority of our kernel-based combined scheme.

5. Conclusion

This paper presents a kernel-based object tracking scheme by matching detected SURF features, which is composed of two stages, from coarse to fine. The first stage estimates a motion parameter by establishing feature correspondences between the original object and the input frame, using adaptive classifiers. The second stage applies the feature points in global representation to refine the object region. Moreover, we introduce kernel function to make sure the more repeatable keypoints are given larger weights for classifier learning. Experimental results using real image sequences demonstrate the stability and accuracy of the proposed technique.

References

[1] I. Matthews, T. Ishikawa, and S. Baker, “The template update problem,” IEEE Trans. Pattern Anal. Mach. Intell., vol.26, pp.810–815, 2004.
[2] D. Ross, J. Lim et al., “Incremental learning for robust visual tracking,” IJCV, vol.77, no.1, pp.125–141, 2008.
[3] H. Grabner, C. Leistner, and H. Bischof, “Semi-supervised on-line boosting for robust tracking,” Proc. European Conference on Computer Vision, 2008.
[4] V. Lepetit, P. Lagger, and P. Fua, “Randomized trees for real-time keypoint recognition,” CVPR, pp.775–781, 2005.
[5] M. Grabner, H. Grabner, and H. Bischof, Learning features for tracking, CVPR, 2007.
[6] H. Grabner and H. Bischof, “On-line boosting and vision,” CVPR, 2006.
[7] Q. Miao, G. Wang, X. Lin et al., “Scale and rotation invariant feature-based object tracking via modified on-line boosting,” ICIP, pp.3929–3932, 2010.
[8] Q. Miao, G. Wang, C. Shi, X. Lin et al., “A new framework for online object tracking based on SURF,” Pattern Recognit. Lett., vol.32, no.13, pp.1564–1571, 2011.
[9] L. Morency, J. Whitehill, and J. Movellan, “Monocular head pose estimation using generalized adaptive view-based appearance model,” Image and Vision Computing, vol.28, no.5, pp.754–761, May 2010.
[10] S. Kumano, K. Otsuka et al., “Combining stochastic and deterministic search for pose-invariant facial expression recognition,” British Machine Vision Conference, 2008.
[11] H. Bay, T. Tuytelaars and L. Van Gool, “SURF: Speed up robust features,” Proc. European Conf. of Computer Vision (ECCV), pp.404–417, 2006.
[12] S. Baker and I. Matthews, “Lucas-Kanade 20 Years on: A unifying framework,” IJCV, vol.56, no.3, pp.221–255, 2004.