IEICE TRANS. INF. & SYST., VOL.E99–D, NO.11 NOVEMBER 2016

LETTER

On-Line Rigid Object Tracking via Discriminative Feature Classification

Quan MIAO[†a)], ***Nonmember***, **Chenbo SHI**[††b)], ***Student Member***, **Long MENG**[†††], ***and*** **Guang CHENG**[†], ***Nonmembers***

SUMMARY This paper proposes an on-line rigid object tracking framework via discriminative object appearance modeling and learning. Strong classifiers are combined with 2D scale-rotation invariant local features to treat tracking as a keypoint matching problem. For on-line boosting, we correspond a Gaussian mixture model (GMM) to each weak classifier and propose a GMM-based classifying mechanism. Meanwhile, self-organizing theory is applied to perform automatic clustering for sequential updating. Benefiting from the invariance of the SURF feature and the proposed on-line classifying technique, we can easily find reliable matching pairs and thus perform accurate and stable tracking. Experiments show that the proposed method achieves better performance than previously reported trackers.

key words: *object tracking, on-line boosting, Gaussian mixture model, self-organizing clustering*

1. Introduction

Recently, on-line classification has received a lot of attention and is widely used in non-rigid object (e.g., human body and faces) tracking [1], [2]. The object region is localized using a bounding box. Tracking is formulated by classifying foreground from background.

For a rigid object (e.g., cup and notebook), the movement of each point within the region can denote the whole change. In contrast, on-line classification has been little used in rigid object tracking. Grabner et al. used the on-line boosting scheme [4] to learn classifier-based keypoint descriptions [3]. However, the keypoints are detected using the Harris corner, which is sensitive to scale changes. Thus the corresponding tracker cannot handle complex variations. Afterwards Miao et al. [5], [6] employed the scale and rotation information of SURF features [7] to guide the on-line classifier learning process. Although the resulting tracking is robust to significant changes like scale, rotation and viewpoint, it cannot ensure long-term accurate tracking. The critical issue is the model degradation caused by inaccurate identification of object region. Target locating errors will accumulate during long-term tracking and cause the appearance model to be updated with a sub-optimal positive sample. Over time this can cause drift and degrade performance.

The proposed method is classified as a rigid object tracker. In contrast to the previous methods [5], [6], the novelty and contributions of this paper are twofold. First, we propose a new discriminative feature modeling and classification mechanism instead of the previously used boosting [4]. The Gaussian mixture model is employed to describe weak classifiers. Meanwhile, we apply self-organizing theory to perform automatic clustering for updating. During updating, each correct match is used as the positive sample and no updates are applied to the false negative samples. The combination of SURF features and the proposed mechanism makes the resulting tracker more accurate against model degradation. Experiments show better tracking performance on challenging video sequences.

2. System Overview

The basic flow is illustrated in Fig. 1. In the first frame, we extract local features within the object region and initialize classifiers. When a new frame $t+1$ arrives, we first detect its keypoints, and then use classifiers to establish matching candidates with frame t. The homography is estimated using RANSAC over the set of matching candidates. The current object region is tracked by geometric transformation on the previous one. Finally, we update the classifiers to make the tracker adaptive to subsequent changes. Both matching and updating are closely related to the proposed GMM-based classifying mechanism.

Each strong classifier C corresponds to one SURF feature within object region. C is composed of J selectors h_j^{sel} and holds a weak classifier pool. It predicts the matching

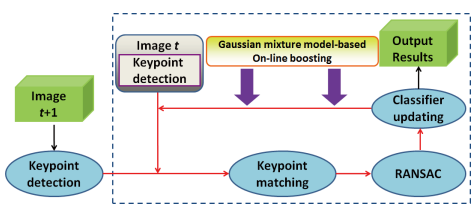

Fig. 1 Framework of the proposed algorithm.

Manuscript received May 9, 2016.
Manuscript revised July 7, 2016.
Manuscript publicized August 3, 2016.

[†]The authors are with the National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing 100029, China.
[††]The author is with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China.
[†††]The author is with Shandong Mingjia Technology Limited Company, Beijing 100084, China.
a) E-mail: miaoq@tsinghua.org.cn
b) E-mail: scb@mails.tsinghua.edu.cn

DOI: 10.1587/transinf.2016EDL8098

confidence measure of a point **x** by:

$$C(\mathbf{x}) = conf(\mathbf{x}) = \sum_{j=1}^{J} \alpha_j \cdot h_j^{sel}(\mathbf{x}) \Big/ \sum_{j=1}^{J} \alpha_j, \tag{1}$$

where $conf(\bullet)$ denotes the confidence measure. The usage of on-line classifiers allows boosting learning by collecting samples over time. As new samples arrive sequentially, each h_j^{sel} re-selects the best weak classifier and updates corresponding α_j.

3. Proposed Algorithm

3.1 GMM-Based On-Line Boosting

In the classifier-based matching, each weak classifier judges whether its corresponding Haar feature $f_j(\mathbf{x})$ is positive or negative. Grabner builds distribution models related to positive samples and negative samples, respectively [4]. The classifying problem is ultimately attributed to a simple binary decision criterion by comparing the probability $P(1 \mid f_j(\mathbf{x}))$ and $P(-1 \mid f_j(x))$. However, such modeling assumes that positive samples should lie on one side of the decision surface while negative samples should appear on the other, which does not match the reality. Indeed the distribution of positive samples tends to be centralized; the emergence of negative samples may be not sufficient to maintain such regularity, which will confuse the classifier.

In Fig. 2, green star denotes the only positive sample obtained from current correct match and red circles denote negative samples randomly chosen from the detected points. When the number of negative samples is small, the classifier is relatively easy to train and can perhaps distinguish the positive sample from negative samples. As negative samples arrive sequentially, the classifier hyperplane becomes oscillating and totally puzzled. Once positive sample and negative ones are mingled, matching accuracy will be influenced seriously.

To address the problem, we only focus on positive samples. On the one hand, since each weak feature has been scale-rotation normalized before updating, positive samples under different transformations are prone into clusters. On the other hand, the positive samples may belong to different classes under different appearance changes. Therefore we model the positive samples using mixture of **K** Gaussian distributions of the form:

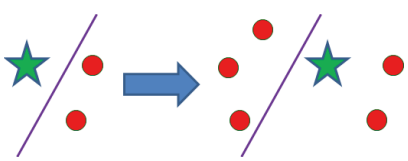

Fig. 2 The drawback of the original classifying mechanism. As the negative samples arrive sequentially, the classifier hyperplane becomes oscillating and totally puzzled.

$$p(x) = \sum_{k=1}^{K} \pi_k N(x \mid \mu_k, \sigma_k). \tag{2}$$

Each Gaussian density $N(x \mid \mu_k, \sigma_k)$ is a component of the mixture and has its own mean μ_k and variance σ_k; π_k is mixing coefficients. If an incoming feature belongs to any interval $(\mu_k - \omega\sigma_k, \mu_k + \omega\sigma_k)$ where ω is a constant, we classify it as positive and assign ω to 1.0.

3.2 Model Updating

First we use the estimated motion parameter to perform a verification procedure over the set of suggested matches, obtaining a subset of correct matches (inliers). Each correct match is used as the positive sample. We apply no updates on false matching candidates.

For each weak classifier, we compute its Haar response x within the positive sample (i.e., the current correct match). Then we find out which mean value of GMM x is closest to. If the response is assigned to the kth component which already has N_k observations, we adjust the relevant parameters as follows:

$$N_k = N_k + 1, \tag{3}$$

$$\mu_k = \mu_k + \frac{1}{N_k}(x - \mu_k), \tag{4}$$

$$\sigma_k^2 = \frac{1}{N_k}(x - \mu_k)^2 + \left(1 - \frac{1}{N_k}\right)\sigma_k^2. \tag{5}$$

Furthermore, we pay close attention to the distribution of each component. Self-organizing clustering is adopted to keep optimizing the **K** mixtures, as is shown in Fig. 3. We use the 1-of-K scheme to assign the nth data point of the kth component as x_{kn}. Then we focus on minimizing the following distortion measure:

$$Q = \sum_{k=1}^{K} \sum_{n=1}^{N_k} (x_{kn} - \mu_k)^2. \tag{6}$$

which depicts sum of squares of the distances between each data point and its center. Our goal is to find for each component its clustered data points. EM algorithm is perform for a two-stage optimization. Current **K** Gaussian mixtures are used as the initial value.

3.3 Discussion about GMM

Recently, GMM has been widely employed in non-rigid

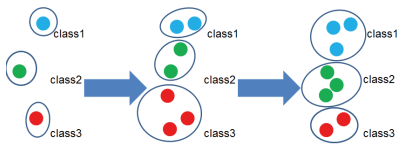

Fig. 3 Self-organizing clustering of GMM is adopted to keep optimizing the **K** mixtures.

object tracking [1]. Each tracker shows its novelty. Take Wang's tracker [8] for example, a spatial-color mixture of Gaussians appearance model was presented to encode both spatial layout and color information. For Kim's tracker [9], GMM was employed to weight the influence of strong classifier which judges whether each input bounding domain belongs to object region.

The usage of GMM in this paper is essentially different from others, including object model, motion model, classifying mechanism, and online learning. The novelty of this paper is to combine the invariance of SURF feature and GMM-based on-line boosting technique, making resulted classifiers accurate to establish reliable matching correspondences. First, the proposed tracker is appropriate for rigid objects and use homography to map consecutive object regions. In addition, after modeling strong classifiers using SURF-based keypoints, we correspond GMM to weak classifiers for discriminative differentiation. Self-organizing automatic clustering is applied. Finally, supervised learning stage is not needed and no updates are applied on false matching candidates, which obviously saves computation without sacrificing performance.

4. Experimental Results

Each strong classifier contains a global weak classifier pool holding 250 weak hypotheses. Each weak classifier is modeled using 3 Gaussian mixtures ($\mathbf{K} = 3$). After keypoint matching, we choose the best 40 matching candidates to perform RANSAC. If the number of inliers exceeds 10 (the percentage is above 25%), the object is tracked based on the estimated homography.

Rigid object trackers are essentially different from non-rigid object trackers in many aspects, such as motion model, classifying mechanism and online learning. The non-rigid object trackers cited in [1], [2] are not appropriate for comparison. Thus we use Grabner's tracker [3] and our original tracker [6]. The speed of Grabner's tracker is 15fps on a PC with 2.93 GHz CPU and 4 GB RAM. The original tracker runs at a speed of about 7fps. The proposed tracker runs at a speed of about 9fps. To further accelerate the proposed tracker, we implement the SURF-based keypoint detection using CUDA [10]. The CPU-GPU cooperative tracking system runs at a real-time speed of 18fps. All these trackers are suitable for planar objects because homography is used as motion model.

4.1 Results on Public Datasets

In this subsection, experiments are carried on published videos for an intuitive comparison. First, Grabner's datasets [3] are downloaded and used. The results are shown in Fig. 4, in which the proposed tracker (in white) overcomes the upcoming object changes (e.g., frame 106 of line 1 and frame 323 of line 2) and is superior to Grabner's tracker (in yellow).

In addition, our previous video [6] is used. Figure 5

Fig. 4 Tracking results using published datasets.

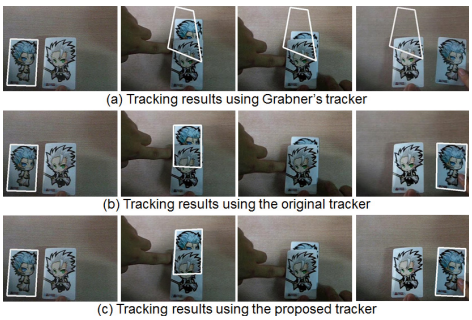

Fig. 5 Tracking results using published datasets. From left to right column, the first, 52nd, 55th and 67th frame.

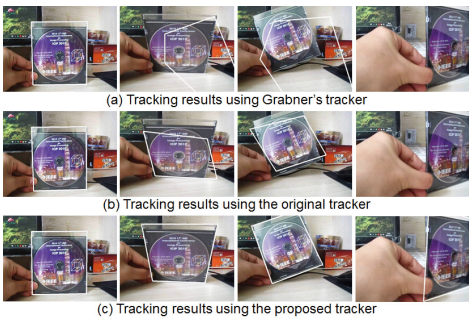

Fig. 6 Tracking a CD under various changes. From left to right column, the first, 563rd, 663rd and 825th frame.

indicates that the proposed tracker is at least comparable to the original tracker and is better than Grabner's tracker. The sequence containing occlusion change is not long, which is not difficult for both trackers.

4.2 Results on Challenging Scene

Figure 6 shows the performance of tracking a CD. The longer sequence is captured by moving the CD back and forth and rotating it, causing rapid scale, rotation, illumination and viewpoint variations. For Grabner's tracker, the identified region in the 663th frame becomes completely out of shape. For the original tracker, distorted object region appeared in the 663th frame, which will influence tracking performance in later frames. Tracking failure occurs in the 825th frame. In contrast, our tracker does not exhibit this error.

To show the strength of GMM-based classifying mechanism, we analyze the distribution of a certain weak Haar re-

sponse of both positive and negative samples used over output frames. First we crop out a short sequence of 50 frames, from Frame 854 to Frame 903. Figure 7 (a) shows the used Haar feature of a strong classifier corresponding to a certain object keypoint. The scale and rotation information is reflected. Figure 7 (b) gives four example frames of this short sequence. All the 50 frames undergo successful tracking, during which viewpoint change and illumination variation occurs continuously. For each frame, the correspondence to the marked object keypoint in Fig. 7 (a) is proven to be correct matches (inliers) by the verification step.

For further validation of using GMM, Fig. 8 depicts the Haar feature distribution over both positive and negative variables, using normalized histogram. Positive samples under different transformations are combined into clusters while distribution of negative samples is totally random. Our approach of using GMM is highly suitable for differentiating positive samples from negative ones, making the formed classifiers preserve sufficient discriminative power.

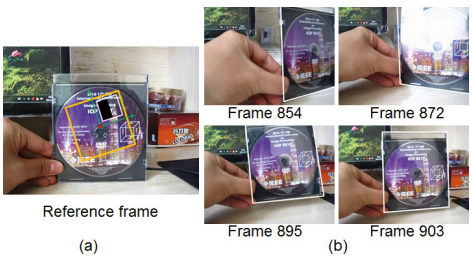

Fig. 7 (a) Used Haar feature of a strong classifier. (b) Four example frames of the output sequence.

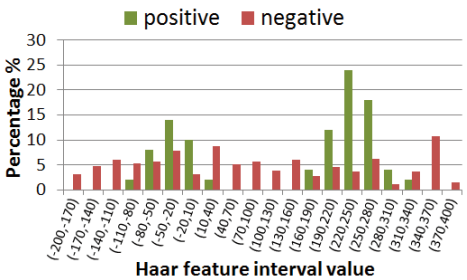

Fig. 8 Histogram distribution of a certain Haar feature over both positive and negative samples, from Frame 854 to Frame 903. Normalization is performed over the 20 bins.

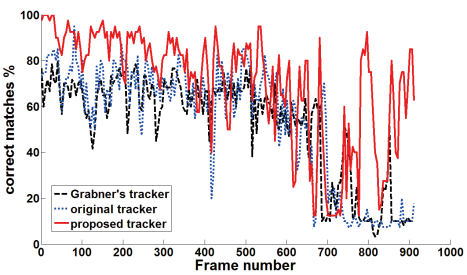

Fig. 9 Number of the matches of the proposed tracker versus Grabner's tracker and the original tracker.

For quantitative comparison, Fig. 9 shows the number of matches over time. When the percentage of correct matches is above 25%, we consider tracking is successful. Tracking loss occurs continuously from frame 700 to frame 900 using Grabner's tracker and the original tracker. However, our tracker can handle the phenomenon, except around some frames where the object drastically changes its appearance. In most cases, the proposed tracker establish more correct matches than those in the other two methods, which demonstrates the matching accuracy against model degradation and the adaptiveness of model matching and updating.

5. Conclusion

This paper treats rigid object tracking as a keypoint matching problem. SURF features are matched by on-line classifiers. During on-line boosting, we propose a Gaussian mixture model based classification mechanism. In the subsequent classifier updating scheme, self-organizing theory is employed to perform automatic clustering. Only the distribution of positive samples is focused on, and no updates are applied to the false matching candidates. Experimental results show the robustness and accuracy of the proposed method.

References

[1] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. van den Hengel, "A survey of appearance models in visual object tracking," ACM Trans. Intelligent Systems and Technology, vol.4, no.4, Article No.58, 2013.

[2] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," 2013 IEEE Conference on Computer Vision and Pattern Recognition, pp.2411–2418, 2013.

[3] M. Grabner, H. Grabner, and H. Bischof, "Learning features for tracking," 2007 IEEE Conference on Computer Vision and Pattern Recognition, 2007.

[4] H. Grabner and H. Bischof, "On-line boosting and vision," 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006.

[5] Q. Miao, G. Wang, X. Lin, Y. Wang, C. Shi, and C. Liao, "Scale and rotation invariant feature-based object tracking via modified on-line boosting," 2010 IEEE International Conference on Image Processing, pp.3929–3932, 2010.

[6] Q. Miao, G. Wang, C. Shi, X. Lin, and Z. Ruan, "A new framework for on-line object tracking based on SURF," Pattern Recognit. Lett., vol.32, no.13, pp.1564–1571, 2011.

[7] H. Bay, A. Ess, T. Tuytelaars, and L. van Gool, "Speeded-up robust features," Computer Vision and Image Understanding, vol.110, no.3, pp.346–359, 2008.

[8] H. Wang, D. Suter, K. Schindler, and C. Shen, "Adaptive object tracking based on an effective appearance filter," IEEE Trans. Pattern Anal. Mach. Intell., vol.29, no.9, pp.1661–1667, 2007.

[9] T.-K. Kim, T. Woodley, B. Stenger, and R. Cipolla, "Online multiple classifier boosting for object tracking," 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp.1–6, 2010.

[10] Q. Miao, G. Wang, and X. Lin, "Kernel-based on-line object tracking combining both local description and global representation," IEICE Trans. Inf. & Syst., vol.E96-D, no.1, pp.159–162, Jan. 2013.