Pattern Recognition Letters 32 (2011) 1564–1571

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

A new framework for on-line object tracking based on SURF

Quan Miao, Guijin Wang *, Chenbo Shi, Xinggang Lin, Zhiwei Ruan

Department of Electronic Engineering, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Article history:
Received 15 October 2010
Available online 30 May 2011
Communicated by Y. Liu

Keywords:
Object tracking
Keypoint matching
On-line boosting
Adaptive classifiers
SURF

ABSTRACT

We present a new object tracking scheme by employing adaptive classifiers to match the corresponding keypoints between consecutive frames. The detection of interest points is a critical step in obtaining robust local descriptions. This paper proposes an efficient feature detector based on SURF, by incrementally predicting the search space, to enhance the repeatability of the tracked interest points. Instead of computing the SURF descriptor, we construct a classifier-based descriptor using on-line boosting. With on-line learning ability based on our sample weighting mechanism, the classifier maintains its discriminative power to establish robust feature description and reliable points matching for subsequent tracking. In addition, matching candidates are validated using improved RANSAC to ensure correct updates and accurate tracking. All of these ingredients contribute measurably to improving overall tracking performance. Experimental results demonstrate the robustness and accuracy of our proposed technique.

1. Introduction

Robust object tracking under real-world conditions is still an open issue and limits the use of state-of-the-art methods in applications ranging from visual surveillance to human–computer interfaces. The difficulties of object tracking include complicated object appearance variations, illumination change, partial occlusions and cluttered scenes.

Recently, tracking formulated as a classification problem has received a lot of attention due to its promising results. The classification-based tracking algorithms can be classified into two categories: region-based methods and feature-based methods. In case of the region-based methods (Avidan, 2004, 2005; Collins and Liu, 2005; Grabner et al., 2008), the basic idea is to learn a binary classifier which distinguishes the object from the background. The main advantage of region-based method is its relative robustness against illumination change, occlusion and cluttered scenes. However, these approaches have problems with complex transformations of the target object. In contrast, feature-based trackers (Grabner et al., 2007; Lepetit et al., 2005; Meltzer et al., 2004) are more adaptive to the object transformations. In (Lepetit et al., 2005; Lepetit and Fua, 2006), a feature-based tracker proposes randomized trees and ferns to discriminate keypoints from each other by classifiers. Although their algorithm demonstrates excellent empirical results, it entails learning a set of object changes before the tracking task begins. To achieve robust tracking with this method, it is imperative to collect a large set of training images covering the range of possible appearance variation, costing a considerable amount of time.

To cope with these problems, Grabner et al. (2007) propose an efficient tracking approach which employs the on-line boosting algorithm (Grabner and Bischof, 2006). However, such approaches typically operate on the premise that the model of the target object does not change drastically over time. The keypoints are detected using Harris corner which is sensitive to scale changes, not to mention more complex transformations. The tracker is prone to failure when significant appearance variations such as affine transformation and viewpoint change arise.

By contrast, in our earlier work (Miao et al., 2010), we propose a rough-but-robust feature-based tracking algorithm which fuses the keypoints' scale and rotation information into the on-line boosting technique. This paper further expands the original idea and thus provides in detail a new framework which fully improves the robustness of object tracking. Our contributions can be summarized as follows:

(1) To exploit the sequential patterns in the data, such as correlations between observations close in the sequence, we efficiently compute the SURF features in each video frame by incrementally predicting the object region.
(2) We employ the scale information and the dominant orientation of SURF feature to guide the discriminative learning process of the keypoints' description. This leads to a series of scale- and rotation-invariant classifiers that are able to cope with significant appearance variations between frames.
(3) Unlike standard RANSAC (Hartley and Zisserman, 2004), we employ a non-uniform sampling strategy according to the matching score of each classifier. That is, we consider the

* Corresponding author. Tel.: +86 10 62781291; fax: +86 10 62780317.
E-mail address: wangguijin@tsinghua.edu.cn (G. Wang).

0167-8655/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2011.05.017

matches with higher matching score more reliable and give them larger weight, to achieve efficient verification and robust estimation of the homography.

(4) We improve the on-line boosting technique by adaptively updating the classifiers. Discriminative samples are selected and assigned higher importance weights.

The remainder of this paper is organized as follows. Section 2 reviews the on-line boosting technique and gives a short survey on existing local feature detectors. A detailed description of the whole tracking framework is presented in Section 3. In Section 4, we illustrate how to implement the proposed algorithm and give a brief analysis of parameters setting. Section 5 is dedicated to experimental validations and Section 6 concludes this paper with remarks on potential extensions for future work.

2. Background and related work

2.1. On-line boosting

The underlying idea of boosting is to combine a set of well selected weak classifiers (Freund and Schapire, 1997) to form a strong classifier. After the seminal work of Viola and Jones (2001), boosting has been successfully used in many computer vision problems, such as human detection (Laptev, 2006), image retrieval (Tieu and Viola, 2004), face detection (Viola and Jones, 2004), etc.

Recently, there has been considerable research interest in on-line vision applications, in which the learning and updating phase are performed on-line as new samples arrive. Oza and Russell (2001) make the primary efforts on studying on-line boosting and demonstrate their equivalence to the off-line counterparts under particular conditions. Based on Oza and Russell (2001), Nair and Clark (2002) employ on-line boosting in a co-training framework for object detection, Collins and Liu (2005) apply on-line discriminative learning in object tracking and Grabner and Bischof (2006) propose a novel on-line boosting for feature selection, etc.

There is a rich literature in on-line boosting and a thorough discussion on this topic is beyond the scope of this paper. Here, we briefly review the most relevant on-line boosting algorithm (Grabner and Bischof, 2006) in which the strong boosted classifier C is composed of J selectors h_j^{sel}. Each classifier holds a binary weak classifier pool X from which the training procedure selects the ones with the minimal estimated error. The strong classifier wishes to predict the matching confidence measure of an unknown point $\boldsymbol{x}$ by:

$$C(\boldsymbol{x}) = conf(\boldsymbol{x}), \tag{1}$$

$$conf(\boldsymbol{x}) = \sum_{j=1}^{J} \alpha_j \cdot h_j^{sel}(\boldsymbol{x}) \Big/ \sum_{j=1}^{J} \alpha_j, \tag{2}$$

where the value $conf(\bullet)$ denotes the confidence measure. As new samples arrive sequentially, each selector h_j^{sel} is responsible for re-selecting the best weak classifier and the corresponding voting weight α_j is updated.

During boosting learning, how to construct a robust weak classifier pool is an important issue. The method described in (Grabner and Bischof, 2006) uses the standard Haar-like features (Viola and Jones, 2001) computed in a fixed bounding patch centered at the corresponding keypoint, which can only deal with pure translations and slight rotations. This paper employs the scheme we proposed in (Miao et al., 2010) where the scale and the dominant orientation of the keypoint are incorporated in the weak classifier pool. In addition, each sample should bear an importance weight to indicate its contribution to the classifier update. Grabner's method gives all the samples equal weight. We emphasize the negative samples that are "similar" to the positive one, to make the updated classifiers more discriminative.

2.2. Feature detectors

Feature detectors, which provide the feature points to be matched (Li and Allinson, 2008), are widely utilized in a large number of applications such as image retrieval (Tuytelaars and Van Gool, 2004), image registration (Brown and Lowe, 2007), and object recognition (Lowe, 2004). Feature detectors can be traced back to the Moravec's corner detector (Moravec, 1977), and improved by Harris and Stephens (1988) to make it more repeatable under small image variations. However, Harris corners are very sensitive to changes in image scale, so it does not provide a good basis for matching images of different sizes. Lindeberg (1998) introduces the concept of automatic scale selection. Based on Lindeberg (1998), several approaches to scale-invariant interest point detection have been proposed, such as the detector based on Harris-Laplace and Hessian-Laplace by Mikolajczyk and Schmid (2001), Difference of Gaussians (DoG) in SIFT by Lowe (2004), and Hessians approximated in SURF by Bay et al. (2006). Matas et al. (2002) have also developed the maximally stable extremal region (MSER) detector, which is a watershed-like method.

In this paper, we use the SURF detector (Bay et al., 2006) to extract keypoints because of its high detection accuracy and full invariance to rotation and scale changes. Furthermore, it can be computed efficiently due to the use of integral images.

3. Proposed algorithm

Feature-based object tracking involves three consecutive steps: feature detection, feature description and feature matching. In feature detection, we incrementally detect keypoints based on SURF. Then we compute the classifier-based descriptions, followed by feature matching in which adaptive classifiers are employed.

The target object region is located in the first frame, either manually or by using an automated detector. When a new frame arrives, we establish matching candidates with the previous frame by means of the feature-based scheme mentioned above. The homography H is estimated using weighted RANSAC over the set of matching candidates. The on-line classifiers are updated to perform further target tracking in the subsequent frame. In the remainder of this section we will describe the algorithm shown in Fig. 1.

3.1. Local feature detection

As is pointed in (Mikolajczyk and Schmid, 2001), the repeatability of the Harris corner detector fails when image resolution changes significantly. In contrast, the SURF detector is more robust to variations. Ta et al. (2009) propose an incremental SURF detection scheme to detect matching candidates of each keypoint in a local neighborhood, aiming to make establishing feature correspondences easier. However, the neighborhood has to be three dimensional (including the scale space), which will take time to search. Moreover, it will be a waste of memory since there are often overlaps between the neighborhoods of different keypoints within the object.

In this subsection, we efficiently detect keypoints in each frame by predicting the object region. As feature matching is performed within the object region in our tracking scheme, predicting the target object means telling the possible range matching candidates are located in. Suppose we are observing a binary variable describing whether on a particular day it rains or not. If we consider the

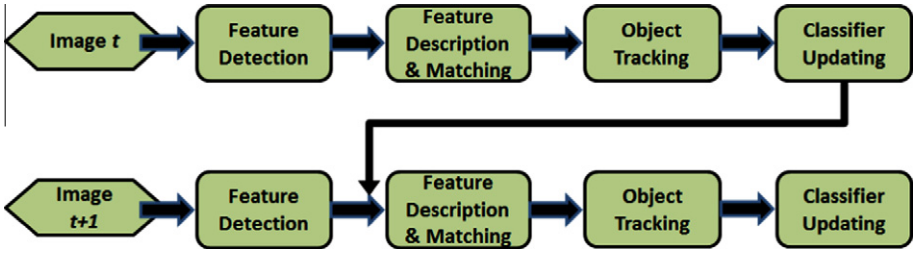

Fig. 1. Flowchart of the proposed on-line feature-based tracking algorithm.

distribution as i.i.d., then the only information we need is the relative frequency of rainy days. However, we know in practice that the weather often exhibits trends that may last for several days. Observing whether or not it rains today is therefore of significant help in predicting if it will rain tomorrow. Similarly, since most of the time consecutive frames are highly correlated, we use the homography $H_{t-2,t-1}$ between the consecutive frames I_{t-2} and I_{t-1} to simply transform the object region $\boldsymbol{bound}_{t-1}$ to get the predicting object region $\boldsymbol{bound}_t^{pre}$ in frame I_t:

$$\boldsymbol{bound}_t^{pre} = \boldsymbol{Map}(\boldsymbol{bound}_{t-1}; H_{t-2,t-1}), \tag{3}$$

where $\boldsymbol{Map}(\boldsymbol{bound}; H)$ is the mapping function based on the homography H.

In practice, the predicted searching space $\boldsymbol{bound}_t^{pre}$ works well in most cases, except when abrupt object movement occurs. To solve this problem, we use a global detection scheme as a backup to ensure the efficiency and accuracy of our tracker. For details about the global detection, refer to Section 4.

3.2. Robust classifier-based matching under the guidance of scale and orientation

In our matching scheme, the classifier-based keypoint description is formed by a linear combination of weak classifiers, chosen by selectors from a weak classifier pool X. For each feature point, Grabner et al. (2007) use fixed neighborhood to construct the weak classifier pool, which cannot cover the same image content as the image region will change with scale and rotation variations. In our scheme, the neighboring region contains scale and dominant orientation information to make the description invariant to complex changes.

Similar to Grabner and Bischof (2006), we also use Haar wavelet representation in weak classifier to capture the structural similarities between object changes. In addition, we create a $Ms \times Ms$ (M is a constant) square region to compute the Haar-like responses. The window size is determined by the point's scale. Moreover, the region is rotated relative to the dominant orientation α. Our approach samples this window with sampling step s and then computes the Haar wavelet responses at each sample point. Furthermore, the responses are distributed in horizontal and vertical direction where "horizontal" and "vertical" is defined in relation to α, followed by normalizing each response within the $s \times s$ region. We use two kinds of features. The value of a two-rectangle feature is the difference between the sum of the pixels within two rectangular regions. The regions have the same size and shape related to scale and are adjacent along the dominant orientation. A three-rectangle feature computes the sum within two outside rectangles subtracted from the sum in a center rectangle. All these features are used to establish a binary weak classifier pool for selectors to choose from. All classifiers are normalized to the point's scale and represented relative to the orientation, thus achieving invariance to image scale and rotation.

As can be seen, our classifier-based keypoint description is quite different from the SURF feature description despite the same keypoints detection. As for the SURF feature descriptor, first the interest region should be split up into 4×4 square sub-regions with 5×5 regularly spaced samples points inside. Then the spatial distribution of intensity changes in each sub-region is computed, weighted and summed, followed by histogram-based calculation. After performing interpolation and normalization on the resulted histogram, the 64D feature vector is thus achieved. The best matching candidate is found by identifying the nearest neighbor in the database of detected keypoints. The nearest neighbor is defined as the keypoint with minimum Euclidean distance for the established descriptor vector, using k–d tree or exhaustive search. In contrast, our work mainly employs the scale and dominant orientation information to guide keypoints matching and classifier learning. Take keypoints matching for example, we treat it as a classifying problem, rather than searching similar feature vector. According to scale and orientation, each selector of the strong classifier seeks its corresponding Haar-like feature within the invariant neighborhood centered around the current keypoint and outputs the classification result. Similarly, we can also apply the scale and orientation of selected samples to steer classifier updating, which is somewhat like an inverse process to matching.

In general, the SURF feature descriptor is more suitable for matching wide-baseline static images, while our on-line classifier-based matching can adapt to the object changes in video sequences better, even though the changes may be complex. As for computing complexity, the Euclidean distance between 64D feature vectors has to be computed for the SURF feature descriptor while in our scheme 20 selectors are sufficient to produce a highly efficient classifier (refer to Section 4.2).

3.3. Object tracking

The object tracking problem is formulated as follows. We build a P-class discriminant by constructing P classifiers $\{C_1, C_2, \ldots, C_P\}$, each corresponding to a keypoint $\{k_1, k_2, \ldots, k_P\}$ lying within the current object region. Given the keypoints set $\Upsilon = \{\gamma_1, \gamma_2, \ldots, \gamma_Q\}$ detected in the new frame, we employ the classifiers to find the point ε_i corresponding to k_i by:

$$\varepsilon_i = \arg\max_{\gamma_q \in \Upsilon} C_i(\gamma_q). \tag{4}$$

Similarly, the set of matching candidates $\Sigma = \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_P\}$is established by evaluating all the classifiers on Υ.

The homography estimation is fairly straightforward: given the set of matching candidates $\Sigma = \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_P\}$, we proceed to estimate the homography H and reject incorrect matches. Classical RANSAC adopts unique uniform sampling (UUS) strategy to select the matching pairs in each cycle. The drawback is that the confidence information each classifier provides is completely ignored. Some of the matching pairs with relatively low confidence are not reliable enough for computing accurate H. Instead of RANSAC,

a non-uniform sampling strategy called weighted sampling (WS) is introduced in (Kalal et al., 2008). In this subsection, we integrate the weighted sampling strategy with the standard RANSAC, which is very similar to PROSAC (Chun and Matas, 2005). The difference is that samples in PROSAC are drawn from progressively larger sets of top-ranked correspondences while in our scheme the frequency of selecting each matching pair is based on its sampling weight. Our approach uses the confidence information as the sampling weight. That is, we consider the matches with higher confidence measure more reliable and select them more frequently to achieve efficient estimation of the homography. Fig. 2 illustrates the process.

Assume the target object has been successfully identified in frame $t-1$. When frame t arrives, we estimate the homography $H_{t,t-1} = (h_1,\ldots,h_8)^{\mathrm{T}}$. If the number of inliers exceeds a threshold, the object can be tracked by transforming $\boldsymbol{bound}_{t-1}$ using $H_{t,t-1}$.

3.4. Adaptive on-line updating

Once the homography H is successfully computed, the strong classifiers describing the tracked object will be updated using matches as positive samples and other keypoints as negative samples. Since only the correct match can be used as the positive sample for each classifier, how to assign each negative sample the importance weight $\lambda$is a crucial problem. In (Miao et al., 2010), we first proposed to improve the on-line boosting algorithm (Grabner and Bischof, 2006) by non-uniformly updating the keypoints. This subsection reviews this idea and presents a more detailed analysis. On the one hand, the importance weight of each negative sample $\boldsymbol{x}$ is related to its confidence measure:

$$\lambda_1(\boldsymbol{x}) = \mu + \sigma \cdot \exp\{conf(\boldsymbol{x}) + \eta\}, \tag{5}$$

where μ, σ and η are constants. When a sample has a high confidence measure, its importance weight is likewise high. This way, we emphasize the negative samples close to the classifier hyperplane such that the classifier will learn to better distinguish these samples (shown in Fig. 3). The updated classifiers convey much more distinguishable information under the circumstance when objects are similar to the background and make detected feature point more distinguishable from each other. As a result, the on-line trained classifier is less likely to have false positives and thus higher matching accuracy.

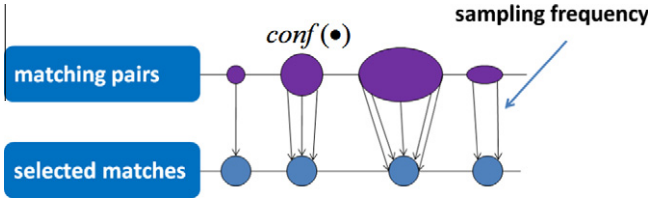

Fig. 2. Weighted sampling for RANSAC. Matches with higher confidence measure are considered more reliable for homography estimation.

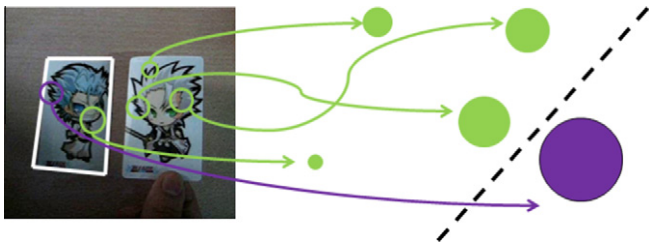

Fig. 3. Negative samples that are closer to the classifier hyperplane are assigned higher importance weight in order to improve the classifier's discriminative power.

On the other hand, we consider the distance between the selected negative sample and the positive one. A kernel function is introduced to give the nearer negative samples more weight:

$$\lambda(\boldsymbol{x}) = \lambda_1(\boldsymbol{x}) \cdot K(\boldsymbol{x} - \hat{\boldsymbol{x}}), \tag{6}$$

where $K(\,)$ is the 2-D realization of the kernel function, which is symmetric and attains its maximum at zero. $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$, respectively denote the selected negative sample and the positive sample. As the proposed method uses sampling strategy in motion estimation, false matching candidates are usually located near the true correspondence in the image domain. The resulted false matching pair is also suitable for the computed motion and mistaken as one of the inliers. Afterwards this will lead to incorrect updating due to the established false positive samples, which seriously influences the accuracy of subsequent tracking. Emphasizing the nearer negative samples can make the updated classifier better distinguish the true correspondence from its confused neighbor. The importance weight of the positive sample $\hat{\boldsymbol{x}}$ (the corresponding keypoint) is the summation of the weights of all the negative samples.

4. Summary of the tracking algorithm

4.1. Implementation details

In the first frame I_1, we need to define the object region. Afterwards, the keypoints set Ψ is detected using the SURF detector and separated into object keypoints Λ and background points O such that $\Lambda \cap \mathrm{O} = \emptyset$, $\Lambda \cup \mathrm{O} = \Psi$. Next we randomly choose P keypoints from Λ to form a subset $K = \{k_1, k_2, \ldots, k_P\}$ ($K \subset \Lambda$), and create the corresponding classifier C_i for k_i. Then we establish weak classifier pool X_i for C_i and employ the boosting algorithm (Grabner and Bischof, 2006) to initialize C_i, and update C_i using k_i as positive sample and other keypoints randomly chosen from Ψ as negative samples.

Now assuming the target object has been successfully tracked in frame I_{t-1}. When frame I_t arrives, we detect the set of keypoints Υ_t using our incremental SURF-based detector. The matching candidates $\Sigma = \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_P\}$ are established by Eq. (4).

The following is a verification step using the weighted RANSAC scheme. If the number of inliers T exceeds a threshold δ, tracking is considered successful and $\boldsymbol{bound}_t$ is computed using $H_{t,t-1}$. We obtain a subset of correct matches $\Sigma^c \subseteq \Sigma$ and then employ our object updating scheme to update the classifiers. Meanwhile, we compute the probability $P_{i,t+1}$ that classifier C_i will still match successfully in the next frame. The method for calculating $P_{i,t+1}$ is the same as (Grabner et al., 2007):

$$P_{i,t+1} = \beta P_{i,t} + (1-\beta)\rho_i, \quad \rho_i = \begin{cases} 1 & \varepsilon_i \in \Sigma^c, \\ 0 & else, \end{cases} \tag{7}$$

where $\beta \in [0,1]$ denotes the weight current probability takes over the matching judgment. That is, $P_{i,t+1}$ is based on $P_{i,t}$ and rise or fall according to the matching judgment. If the current match is correct, we consider the classifier more reliable, and vice versa. If $P_{i,t+1}$ is below a threshold θ, we randomly choose a new keypoint lying within $\boldsymbol{bound}_t$ and initialize a corresponding classifier C_i to better adapt to target changes. On the other hand, if RANSAC fails, we remove the spatial constraint and supplement to detect the keypoints in the whole image as the set Υ_t and repeat the aforementioned steps. If RANSAC still fails, we discard the current frame I_t and wait for I_{t+1}, because I_t might be corrupted by possible motion blur or significant object changes, making the corresponding keypoints difficult to be detected and matched.

4.2. Parameters setting

While details on the whole system are presented in Section 3, several parameters mentioned above merit discussion. In Grabner's tracker which uses the Harris corner detector, 60 classifiers ($P = 60$) are employed in the object. In contrast to the corner detector, the SURF detector detects the local minimal or maximal in a 3D space (including scale space) and also has to eliminate unreliable points by interpolation and computing edge responses. Because of the elimination, the total number of keypoints detected in the target object is reduced. In this paper, we randomly choose 40 keypoints in the object as classifiers and the threshold δ in the verification step is set to 10. Although the quantity decreases, the quality is improved: it outperforms the previously used detector with respect to repeatability and distinctiveness, thus making the tracking performance more reliable.

When computing our classifier-based keypoint description, we calculate the first order Haar wavelet response as weak classifiers in a $Ms \times Ms$ (s denotes scale) neighborhood centered around each keypoint. To make the weak classifier pool richer, we should try to increase M. However, as for the keypoints detected with a very large scale, the length Ms may go beyond the image border. In this paper, we refer to the SURF feature description (Bay et al., 2008) in which the size is $20s$ and set $M = 20$. Hence there will be about 1500 Haar wavelet response, and the weak classifier pool randomly chooses 250 responses as weak classifiers.

Each classifier contains certain number of selectors (J). Classifiers with more selectors will achieve lower false positive rates but also require more time for matching and updating. To find a balance between discriminative power and computational efficiency of classifiers and the number of selectors, we perform tracking on a sequence with about 1000 frames with various object changes using different setting of J. We observe that setting $J = 20$ is appropriate to produce a highly efficient classifier.

5. Experiments

We now present the experimental results of applying our algorithm on several video sequences. All experiments are implemented in C++ code on a PC with 2.2 GHz CPU and 2 GB RAM. The image size is 640×480. For comparison, we implemented the Grabner's tracker described in (Grabner et al., 2007). To better illustrate the soundness of our approach, we have implemented another feature-based tracker 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 for each keypoint is defined as the one with the minimum Euclidean distance for the SURF descriptor vector. We call this second approach the SURF-based method. In experiment section, we will compare the performance of the SURF-based method, Grabner's method and the proposed method, in terms of their ability to handle changes in illumination as well as appearance.

As a first attempt, we focus on tracking the object in the following sequence shown in Fig. 4. The sequence is captured by moving the target up and down and rotating it, causing complex appearance changes including both viewpoint change and the scale and rotation variations. In the beginning, we compare the calculation speed of these methods. For the SURF-based method, it will take more than 200 ms to extract a total number of 900 feature points. In addition to computing the interest point description (64D vector) and the nearest neighbor indexing for matching, the overall tracking speed of the SURF-based method is less than 3 fps. In contrast, our approach of applying the incremental SURF detection saves a lot of time, taking only 50 ms on average. The resulted tracking system runs at a speed of about 7 fps.

The efficiency of our incremental SURF detection can be explained as follows. Specifically, the computation time of SURF detection consists of two parts: point localization and orientation assignment. The computational cost of point localization is directly associated with the image size, in that the calculation of Hessian matrix as well as the scale space representation are built on the image pyramid generated by sub-sampling the original image. Besides, the calculation time of orientation assignment is obviously proportional to the number of detected point. As in most cases the object movement in consecutive frames presents consistent tendency, the incremental detection scheme successfully reduces the search space and the number of detected points, while not sacrificing the performance. As for Grabner's method, the tracker achieves a frame rate of 12 fps (the same as is announced in (Grabner et al., 2007)) due to the fast Harris corner detector. However, we still decide to employ the SURF detection, in that it has

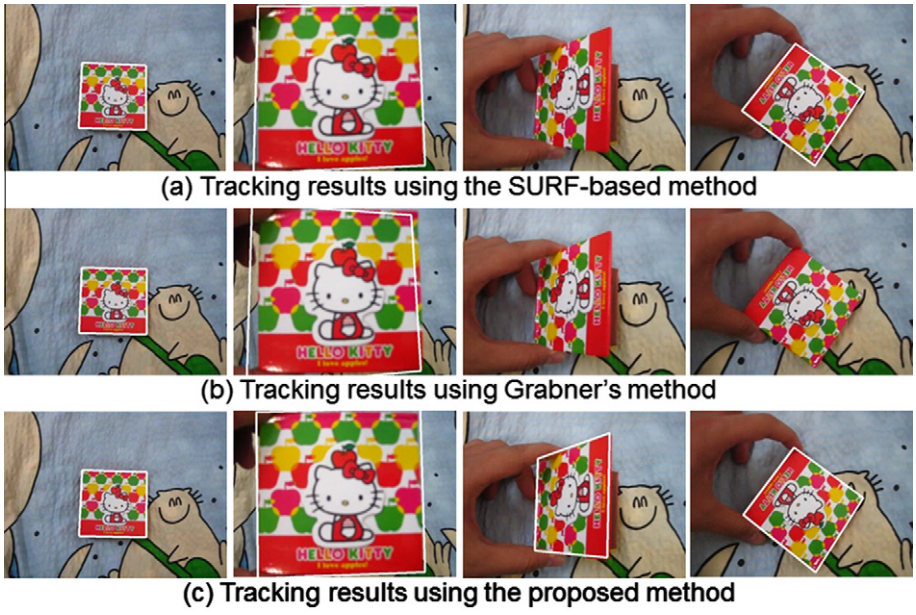

Fig. 4. Tracking a notebook under complex appearance changes. From left to right column, the first, 38th, 142th and 225th frame.

been proven to outperform most previously proposed schemes with respect to robustness and detection repeatability (Tuytelaars and Mikolajczyk, 2008), including Harris corner. The tracking performance below validates the advantages.

As is shown in Fig. 4, once large viewpoint change between the current frame and the first frame occurs (the 142th frame), the SURF-based method could not work because its descriptor is unable to preserve invariance. Grabner's tracker fails to track the object in later frames since it lacks the adaption to complex changes of the object. Our tracker continues to track the object despite the significant appearance change. The performance can be explained by the sample trajectory described in Fig. 5. On-line learning of local features allows us to apply correct matches obtained over time as the positive updates to the corresponding classifier. As for Grabner's tracker, its simple classifiers can only deal with slight changes with the help of on-line learning. Under other more complex changes (e.g. large scale and rotation variation), the corresponding patch for its positive sample is out of square. Thus the classifier updating will fail and the keypoints cannot be classified afterwards. In contrast, the guidance of the scale and rotation information on the on-line classifier learning makes the proposed method ideally suited for such rapid changes. Our positive patches collected during tracking are represented relative to the scale and orientation for the weak classifier pool to find its true counterparts for updating. In addition, the incremental detection scheme helps to further improve the detector's repeatability, especially under the viewpoint change.

Fig. 6 shows the number of matches certificated by RANSAC for each frame. Tracking loss (the percentage is below 25%) occurs frequently using SURF-based method and Grabner's method because

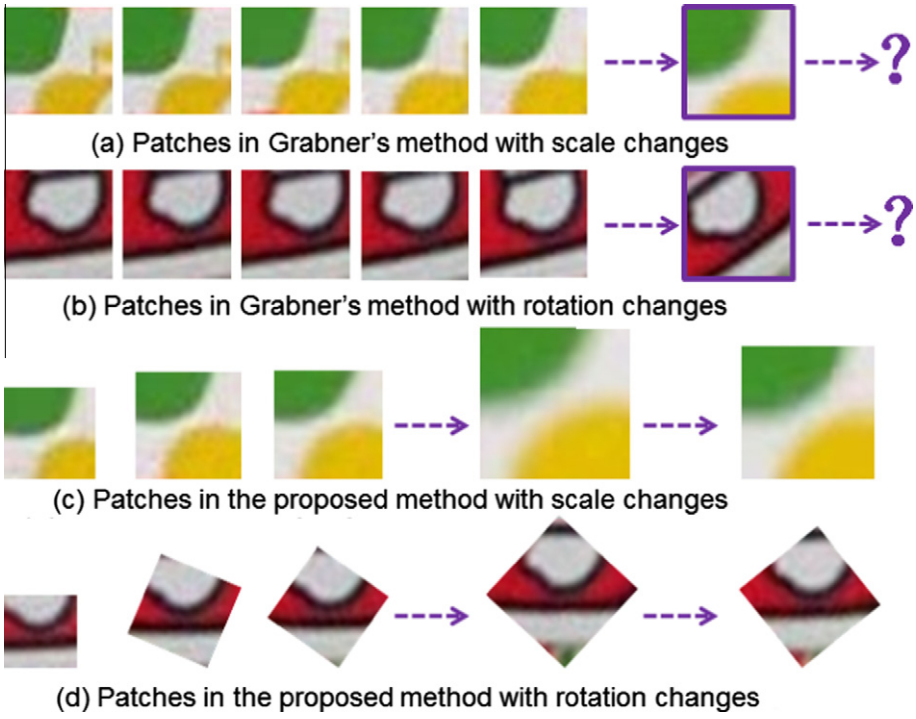

Fig. 5. Patches for positive updating the corresponding classifier are collected during tracking. Patches for negative updates are randomly selected from any other keypoint.

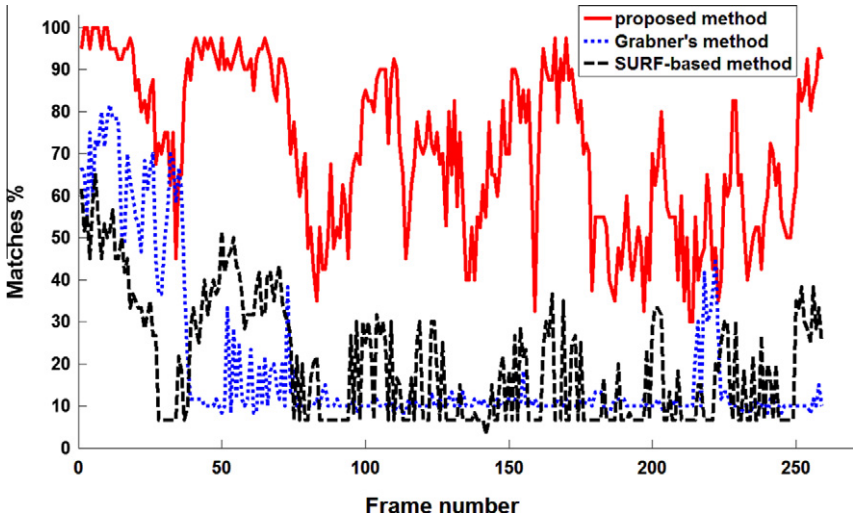

Fig. 6. Number of the matches of the proposed algorithm versus the SURF-based method and Grabner's method.

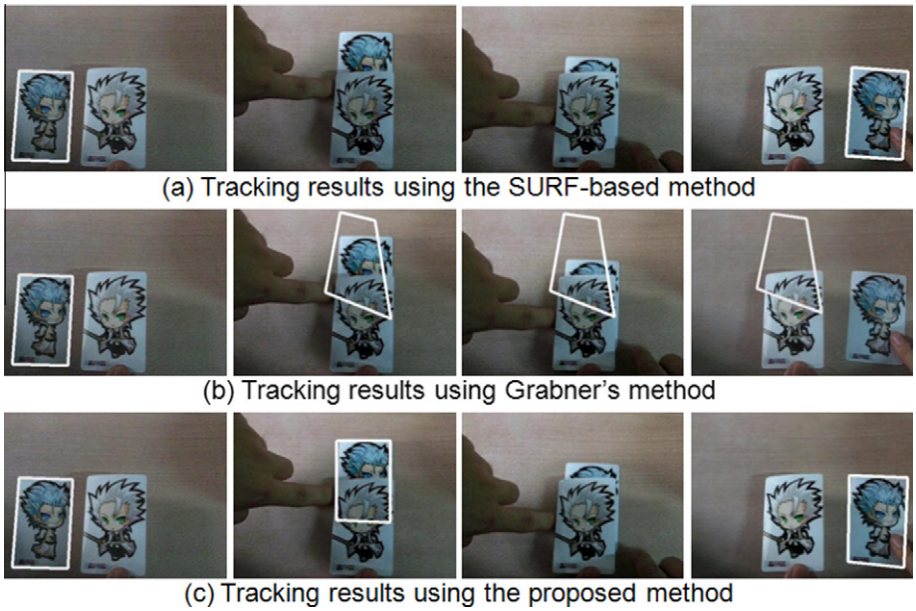

Fig. 7. Tracking a card under image occlusion and scene clutter. From left to right column, the first, 52th, 55th and 67th frame.

of the significant appearance change. However, our proposed tracker can handle the phenomenon. In most general cases, the proposed tracker establishes more correct matches than those in the other two methods, validating the superiority of our approach.

To illustrate the strength of our object updating scheme, we apply the trackers in the following sequence where the target periodically moves behind a similar looking object, causing image occlusion and scene clutter.

The tracking performance is shown in Fig. 7. The SURF-based tracker suffers from image occlusion because some of the corresponding keypoints disappear when occlusion occurs. As for Grabner's tracker, the performance declines around the 52th frame and the tracking region is completely confused at the 55th frame. Some keypoints lying in the occluder are mistaken as the new classifiers, since their descriptions are extremely similar to those in the target object. In contrast, our tracker does not exhibit this error. The reason is that our adaptive object update improves the classifiers' discriminative power and thus rejects the confusing outliers. In addition, the scheme of the weighted sampling for RANSAC makes the keypoint matching more reliable. As a result, the probability $P_{i,t+1}$ for a certain classifier C_i remains high and thus the occluder's keypoints are prevented from being selected as new classifiers.

6. Conclusion and future work

This paper presents a new framework for object tracking based on SURF, in which feature points in the defined object are matched between consecutive frames by adaptive classifiers. The proposed incremental feature detection scheme not only increases the detector's repeatability but also speeds up the whole tracking system. Equipped with the scale and rotation information, the Haar wavelet representation is sufficient to construct reliable classifiers. In addition, the mechanism of weighted sampling for RANSAC and the modified object updating scheme ensure the matching accuracy and the classifier's distinctiveness. Experimental results verify that our approach completely outperforms the state-of-art tracker to achieve robust and accurate tracking.

There are also some interesting ways to extend this work in the future. First, while our current experiments are limited to 2D transformations, future work will attempt to include 3D motions as well. Moreover, while our tracker achieves efficient tracking by employing incremental keypoints detection, future work will involve a more efficient on-line updating process to further improve the tracking speed.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.patrec.2011.05.017.

References

Avidan, S., 2004. Support vector tracking. IEEE Trans. Pattern Anal. Machine Intell. 26, 1064–1072.
Avidan, S., 2005. Ensemble tracking. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 494–501.
Bay, H., Tuytelaars, T., Van Gool, L., 2006. SURF: Speed up robust features. In: Proc. European Conf. of Computer Vision (ECCV), pp. 404–417.
Bay, H., Ess, A., Tuytelaars, T., Van Gool, L., 2008. Speeded-up robust features (SURF). Comput. Vision Image Understanding 110 (3), 346–359.
Brown, M., Lowe, D., 2007. Automatic panoramic image stitching using invariant features. Int. J. Comput. Vision 74, 59–77.
Chun, O., Matas, J., 2005. Matching with PROSAC – progressive sample consensus. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 220–226.
Collins, R., Liu, Y., 2005. Online selection of discriminative tracking features. IEEE Trans. Pattern Anal. Machine Intell. 27 (10), 1631–1643.
Freund, Y., Schapire, R., 1997. A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. System Sci. 55 (1), 119–139.
Grabner, H., Bischof, H., 2006. On-line boosting and vision. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 260–267.
Grabner, M., Grabner, H., Bischof, H., 2007. Learning features for tracking. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR).
Grabner, H., Leistner, C., Bischof, H., 2008. Semi-supervised on-line boosting for robust tracking. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR).
Harris, C., Stephens, M., 1988. A combined corner and edge detector. Alevy Vision Conference, pp. 147–151.
Hartley, R.I., Zisserman, A., 2004. Multiple View Geometry in Computer Vision, second ed. Cambridge University Press.
Kalal, Z., Matas, J., Mikolajczyk, K., 2008. Weighted sampling for large-scale boosting. In: Proc. British Machine Vision Conf. (BMVC).
Laptev, I., 2006. Improvements of object detection using boosted histograms. In: Proc. British Machine Vision Conf. (BMVC), pp. 949–958.
Lepetit, V., Fua, P., 2006. Keypoint recognition using randomized trees. IEEE Trans. Pattern Anal. Machine Intell. 28 (9), 1465–1479.
Lepetit, V., Lagger, P., Fua, P., 2005. Randomized trees for real-time keypoint recognition. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 775–781.

Li, J., Allinson, N.M., 2008. A comprehensive review of current local features for computer vision. Neurocomputing 71 (10–12), 1771–1787.
Lindeberg, T., 1998. Feature detection with automatic scale selection. Int. J. Comput. Vision. 45 (2), 79–116.
Lowe, D., 2004. Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision. 60 (2), 91–110.
Matas, J., Chun, O., Urban, M., Pajdla, T., 2002. Robust wide-baseline stereo from maximally stable extremal regions. In: Proc. British Machine Vision Conf. (BMVC), pp. 384–393.
Meltzer, J., Yang, M.-H., Gupta, R., Soatto, S., 2004. Multiple view feature descriptors from image sequences via kernel principal component analysis. In: Proc. European Conf. of Computer Vision (ECCV), pp. 215–227.
Miao, Q., Wang, G., Lin, X., Wang, Y., Shi, C., Liao C., 2010. Scale and rotation invariant feature-based object tracking via modified on-line boosting. In: Proc. IEEE Conf. on Image Processing (ICIP) 2010, Hong Kong.
Mikolajczyk, K., Schmid, C., 2001. Indexing based on scale invariant interest points. In: Proc. Internat. Conf. on Computer Vision (ICCV), pp. 525–531.
Moravec, H., 1977. Towards automatic visual obstacle avoidance. In: Proc. Internat. Joint Conf. on Artificial Intelligence.
Nair, V., Clark, J., 2002. An unsupervised, online learning framework for moving object detection. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 317–324.
Oza, N., Russell, S., 2001. Experimental comparisons of online and batch versions of bagging and boosting. In: Proc. 7th ACM SIGKDD Internat. Conf. on Knowledge Discovery and Data Mining, pp. 359–364.
Ta, D.-N., Chen, C., Gelfand, N., pulli, K., 2009. SURFTrac: Efficient tracking and continuous object recognition using local feature descriptors. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 2937–2943.
Tieu, K., Viola, P., 2004. Boosting image retrieval. Int. J. Comput. Vision. 56 (1–2), 17–36.
Tuytelaars, T., Mikolajczyk, K., 2008. Local invariant feature detectors: A survey. Found. Trends Comput. Graphics Vision 3 (3), 177–280.
Tuytelaars, T., Van Gool, L., 2004. Matching widely separated views based on affine invariant region. Int. J. Comput. Vision. 59 (1), 61–85.
Viola, P., Jones, M., 2001. Rapid object detection using a boosted cascade of simple features. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 511–518.
Viola, P., Jones, M., 2004. Robust real-time face detection. Int. J. Comput. Vision 57 (2), 137–154.