EURASIP Journal on Advances in Signal Processing
a SpringerOpen Journal

RESEARCH Open Access

Robust online tracking via adaptive samples selection with saliency detection

Jia Yan[1], Xi Chen[1,2*] and QiuPing Zhu[1]

Abstract

Online tracking has shown to be successful in tracking of previously unknown objects. However, there are two important factors which lead to drift problem of online tracking, the one is how to select the exact labeled samples even when the target locations are inaccurate, and the other is how to handle the confusors which have similar features with the target. In this article, we propose a robust online tracking algorithm with adaptive samples selection based on saliency detection to overcome the drift problem. To deal with the problem of degrading the classifiers using mis-aligned samples, we introduce the saliency detection method to our tracking problem. Saliency maps and the strong classifiers are combined to extract the most correct positive samples. Our approach employs a simple yet saliency detection algorithm based on image spectral residual analysis. Furthermore, instead of using the random patches as the negative samples, we propose a reasonable selection criterion, in which both the saliency confidence and similarity are considered with the benefits that confusors in the surrounding background are incorporated into the classifiers update process before the drift occurs. The tracking task is formulated as a binary classification via online boosting framework. Experiment results in several challenging video sequences demonstrate the accuracy and stability of our tracker.

Keywords: Object tracking, Online boosting, Saliency detection, Adaptive samples

1. Introduction

Within the last decades, object tracking has obtained much attention in computer vision. However, for many real-word problems, the ambiguities inherent to the visual data and the tracking process make it difficult to develop accurate, robust, and efficient trackers.

Recently, tracking has been formulated as an online classification problem, where a classifier is trained and updated online to distinguish the object from the background during the tracking [1-6]. This method is also termed as tracking-by-detection, in which a target object identified by the user in the first frame and the surrounding background are described by a group of features, and a binary classifier separates target from background in the successive frames. To handle appearance changes, the classifier is updated incrementally using the new information over time. Avidan [5] used an adaptive ensemble of classifiers for visual tracking. Each weak classifier is a linear hyperplane in an 11D feature space composed of R,G,B color and a histogram of gradient orientations. Grabner et al. [1] have proposed a tracker via online boosting. This work has demonstrated excellent tracking performance in natural scenes such as illumination variations, partial occlusions, and appearance change. This algorithm can be categorized into two steps: the generation of strong classifier and the detection step of the object. The strong classifier used to detect new object location is computed by a linear combination of several selectors which are generated by selecting the weak classifiers with low estimated errors in the feature pool. Saffari et al. [6] proposed the online random forest algorithm based on an online decision tree growing procedure.

The main challenge of online tracking can be attributed to the difficulty in handling the ambiguities: as these classifiers perform self-learning it is difficult to decide where exactly to take the positive and negative updates, respectively, which can lead to slightly wrong updates of the tracker. If these errors accumulate over time and self-reinforce the classifier in its wrong decisions, the tracker can drift easily [2].

* Correspondence: robertcx@whu.edu.cn
[1]DSP Laboratory, Department of Electrical Engineering, School of Electronic Information, Wuhan University, Wuhan 430072, China
[2]Institute of Microelectronics and Information Technology, Wuhan University, Wuhan 430072, China

Many research efforts have been conducted in the literature to deal with this problem. Grabner et al. [7] proposed a semi-supervised approach where labeled examples come from the first frame only, and subsequent training examples are left unlabeled, but it may lose lots of information due to excessively depending on the prior classifier in the first frame. Recently, a semi-supervised learning approach is developed in which positive and negative samples are selected via an online classifier with structural constraints [8], but it is easy to lose the target completely for some frames. Yu et al. [9] propose a co-training-based approach to label incoming data continuously and online update a hybrid discriminative model. Liu et al. [10] deduce a boosting error bound to guide the tracker construction and semi-supervised learning in Adaboost framework. Babenko et al. [2] proposed a novel tracking method based on the online multiple instance learning method, where the current tracking position is considered uncertain and several positive samples are selected close to current object position, arranged in a so-called bag. The classifiers resolve the ambiguities by itself. However, they cannot handle unreliably labeled negative samples [11]. Hence, we would like to point out that these approaches to include new (unreliable labeled) samples are either too firm hindering to acquire new information or too adaptive tending to drift.

Visual saliency is the perceptual quality that makes an object or pixel stand out relative to its neighbors and thus capture our attention. There are many excellent saliency detection methods [12-15]. To the best of the authors' knowledge, the saliency detection method has never been applied in the object tracking tasks, although it has successfully been applied in object-of-interest image segmentation [16], object recognition [17], and content-aware image editing [18].

Inspired by the visual saliency detection approach, we propose a visual saliency detection-based sample selection unifying with online boosting approach for robust object tracking. Our approach solves the problem of inaccurately detected target location. The main components of our tracking algorithm are shown in Figure 1b. The main idea is to combine the current tracking position with the image saliency detection to select the reliable positive and negative samples adaptively to avoid the classifier be harmed by the unreliable sample. Our algorithm can prevent the tracking drift effectively even when the target position is slightly inaccurate. We present the empirical results of our method comparing with several state-of-the-art tracking algorithms on publicly available challenging sequences, experimental results show that our method can lead to a more robust and stable tracker than other methods.

The remainder of this article is organized as follows: Section 2 gives a short review of online boosting algorithm. Section 3 gives a short survey of the existing saliency detection methods and the method we use. A detailed description of the proposed tracking algorithm and a brief analysis are presented in Section 4. In Section 5, we present our experimental results and some discussions. Finally, we conclude the article and outline the future work in Section 6.

2. Online tracking with boosting

2.1. Offline boosting

Boosting was proposed as a classification algorithm in [19]. Any input $\mathbf{X}\in\mathbb{R}^m$ is categorized as one of the classes 1 or −1 using the strong classifier $H(\mathrm{x})$. The classifier: H :

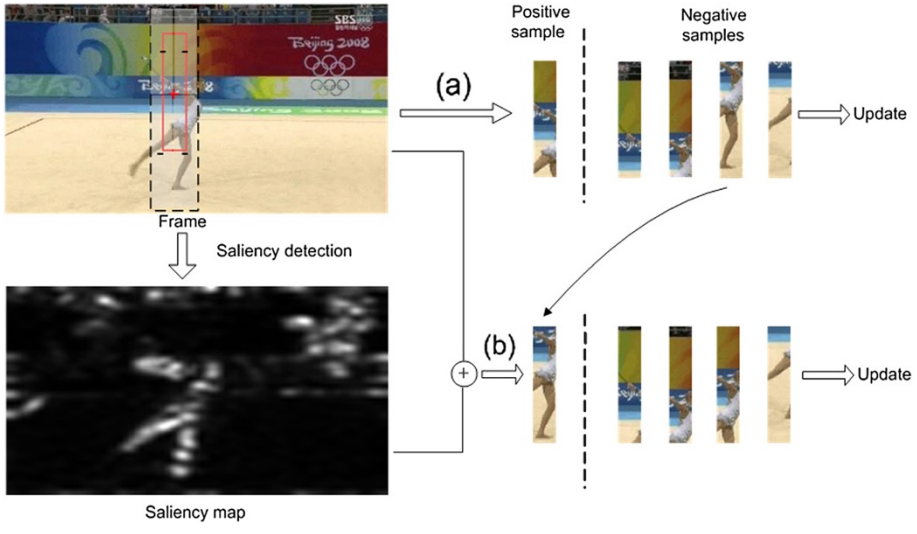

Figure 1 An illustration to explain the main components of online tracking algorithms. (**a**) The original algorithm in [1]. (**b**) Our proposed algorithm.

$\mathbb{R}^m \rightarrow \{1, -1\}$, is a linear combination of N trained weak classifiers $h_n(x)$, $n = 1,\ldots,N$.

$$H(\mathrm{X}) = sign\left(\sum_{n=1}^{N} \alpha_n h_n(\mathrm{X})\right) \tag{1}$$

Given a labeled training set $\widetilde{\mathrm{X}} = \{\langle \mathbf{X}_1, y_1\rangle, \cdots, \langle \mathbf{X}_L, y_L\rangle | \mathbf{X}_l \in \mathbb{R}^m, y_l \in \{1, -1\}\}$ with a set of m-dimensional features $\mathbf{x}_l$, positive or negative labeled samples y_l and an initial uniform distribution over the L samples, the weak classifier h_n is trained in an iterative fashion using $\widetilde{\mathrm{X}} \in \mathbb{R}^{m \times L}$. The weak classifier has to perform only slightly better than random guessing [19]. At nth step the algorithm searches the h_n producing the lowest error e_n. Then the weight α_n for the linear combination given in Equation (1) is calculated by

$$\alpha_n = \frac{1}{2} \ln\left(\frac{1 - e_n}{e_n}\right) \tag{2}$$

In the next iteration, for misclassified samples the corresponding weight is increased while for correctly classified samples the weight is decreased, and all the weights are normalized.

2.2. Online boosting

As opposed to the whole available training set $\widetilde{\mathrm{X}}$ in the offline boosting, the samples arrive one after another in the object tracking tasks. Hence, the weak classifiers have to be updated online every time a new training sample (positive or negative) is available. In online boosting, since no distribution on the samples is available, a distribution on the weak classifiers is maintained in the work by Oza and Russell [20]. The basic idea of online boosting is that the importance λ of a sample can be estimated by propagating it through a fixed set of weak classifiers. The error of the weak classifier e is estimated by the sum of correctly λ^c and incorrectly λ^w samples seen so far: $e = \frac{\lambda^{wrong}}{\lambda^{wrong} + \lambda^{correct}}$ Then the weight of the selected weak classifier is computed in the same way as Equation (2). It is proved that for a number of training samples growing to infinity the online and offline boosting algorithms converge to the same classifier [20].

Grabner et al. [1] introduced online boosting algorithm to the object tracking and demonstrated successful tracking of objects on various sequences. In [1], online boosting is not directly performed on the weak classifiers, but on the "selectors". Let $h_n^{\mathrm{sel}}(n = 1, \ldots, N)$ be the set of N selectors, then the strong classifier is the linear combination of N selectors:

$$H(\mathbf{X}) = sign\left(\sum_{n=1}^{N} \alpha_n h_n^{sel}(\mathbf{X})\right) \tag{3}$$

Selectors share a pool representing the weak classifiers, which are denoted by $h_w(w = 1,\ldots,W)$. When training a selector, weak classifiers are updated with the positive or negative samples and one with the lowest error $e_w e_w$ is selected as h^{sel}.

The strong classifier is served as the target detector. Let $L_p(p = 1,\ldots,P)$ be the candidate positions, and x_{new} be the new location. $x_{\mathrm{new}} = L_{p+}$, where p^+ is the location with highest response of the strong classifier:

$$p^+ = \arg\max_p \left(\sum_{n=1}^{N} \alpha_n h_n^{sel}(L_p)\right) \tag{4}$$

A major challenge in the online boosting algorithms is how to choose the positive and negative samples. In [1,4,5,7,11], the samples centered on the new target location are chosen as the new positive samples and the negative samples are extracted randomly or fixedly by taking regions of the same size as the positive samples from the surrounding background. If the location x_{new} is not precise (i.e., the red rectangle in Figure 1), the classifiers will get updated with sub-optimal samples.

3. Visual saliency detection

Since the classifiers become less distinguishable as the samples become sub-optimal in the update steps, the selection of positive and negative samples needs some guidance even when the target location is not precise. Therefore, we introduce the visual saliency detection to our tracking problem to provide guidance in selecting samples, as shown in Figure 1.

Visual saliency results both from fast, pre-attentive, bottom-up saliency extraction, as well as from slower, task-dependent, top-down saliency extraction [14]. Different to most of the relevant literature, the saliency detection method in our tracking framework must satisfy two requirements: simple and the restrain of the background pixels. Thus, we introduce the method proposed in [13] to produce the saliency map. In [13], the log spectrum of each image is analyzed and the spectral residual is obtained. Then the spectral residual is transformed to spatial domain to obtain the saliency map. This method is independent of features, categories, or other prior knowledge of the image.

Given an image $I(x)$ and its down-sampled image $I'(x)$, we can get the log-magnitude spectrum $L(f)$ and phase magnitude $P(f)$ of the image using the Fourier transform:

$$L(f) = \log(|F[I'(x)]|) \tag{5}$$

$$P(f) = \varphi(F[I'(x)]) \tag{6}$$

Then the spectral residual $R(f)$is found by subtracting a smoothed version of the log-magnitude spectrum from the original log-magnitude spectrum as follows:

$$R(f) = L(f) - L(f) * h_n(f) \tag{7}$$

where the $h_n(f)$ is an$n \times n$ matrix, denotes the average filer. $n = 3$ in [13].

The saliency map$S(x)$ is constructed using the inverse Fourier transform of the spectral residual. $S(x)$is defined as

$$S(x) = F^{-1}[\exp(R(f) + P(f))]^2 \tag{8}$$

In [13], $I'(x)$is an image with the height and width equal 64 pixels. The selection of the scale in our tracking framework will be discussed in Section 4.2.

4. Proposed algorithm

The whole tracking algorithm is summarized in pseudo-code below.

Initialization:

1. Initialize the region of the tracked object in the first frame and the parameters
2. Sampling negative patches
3. Extract Haar-like features for positive and negative samples
4. Train the weak classifiers using the labeled samples
5. Get the strong classifier based on selectors and their weights by (3)
6. Determine the scale of $I'(x)$ by (9)

Online tracking:

For $i = 2$

1. Find the object location in current frame by (4)
2. Obtain the saliency map $S(x)$by (8)
3. Select the positive sample by (14)
4. Extract the negative samples by (15)
5. Update the weak classifiers with the new samples
6. Obtain the selectors and the weights by (2)
7. Get the new strong classifiers by (3)

End

First, we get the tracking window in the first frame, the positive and negative samples are obtained and then used to train the weak classifiers and get the first strong classifier. We propose the use of saliency detection for object tracking. To achieve this goal, we determine the scale of the down-sampled images during the saliency detection process to get better saliency map. To predict the target location in the next frame, the location with highest confidence value by the strong classifier is determined as the new location. Before updating the weak classifiers, we obtain the saliency map and then combine it with the classification confidence map to select the correct positive and negative samples adaptively. This selection mechanism avoids the inaccurate target locating and the influence caused by the confusors in the surrounding background. The classifiers are updated with the new samples and the new strong classifier is obtained finally.

4.1. Initialization

The region of the tracked object is drawn in the first frame manually, and this region (or patch) is the first positive sample. The negative samples are generated by the method in Figure 2a in the first frame. In our approach, we only use Haar-like features, which are randomly generated, to represent each image patch (or samples), similar to [1]. Note that we mainly focus on the investigation of samples selection, we employ a simple feature, RGB color histogram, and HoG histogram also could be included in our tracking framework.

Given the labeled samples, we can train the weak classifiers and obtain the first strong classifier, which to be used to determine the new location of the object in the new frame.

4.2. Scale of the down-sampled image

Visual saliency detection works under certain scales. Changing the scales of the down-sampled image $I'(x)$ leads to different results in saliency maps. When the scale of $I'(x)$ is small, the small objects and detailed features are omitted in the saliency map. But in a large scale, the big objects become less salient to the small but abrupt changes in local details. This property is illustrated in Figure 3, and the values of the scale are chosen as 64 and 128. The tracked object is the motorbike in the "Motorbike" sequence, the size of the target is small compared to the size of the whole frame. The saliency map $S(x)$ of scale is 128, having higher saliency values at the region of the target, which is more accurate to locate the object in our tracking framework. The saliency map of the "Coupon book" sequence is better when the value of the scale is 64.

To measure the property quantitatively, we design a method of measurement. Let $w_I \times h_I$ be the size of the frame image, and $w_I \times h_I$ be the size of the target. The relative size of the target is estimated by

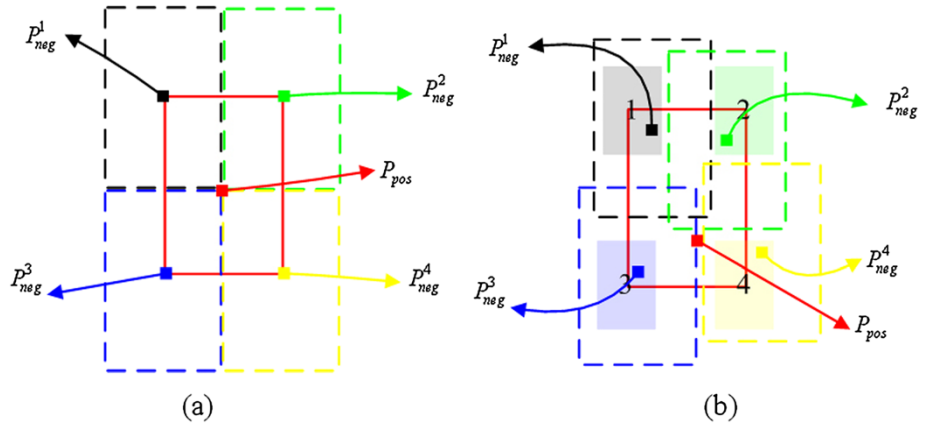

Figure 2 Illustration of accuracy and robustness of our approach in selecting samples compared with the fixed mode. (**a**) The fixed mode in [1]. (**b**) The adaptive mode in the proposed algorithm.

$$\gamma = \frac{w_t \times h_t}{w_I \times h_I} \tag{9}$$

if $\gamma < \theta$, we choose 128 as the scale. Otherwise, the value of the scale is set to 64. Note that, although some pixels inside the region of the target have very low saliency values (nearly zero), it does not matter the utilization of$S(x)$, because it is only used to guide the selection of the positive and negative samples, not to detect the target.

4.3. Samples selection using saliency detection

Given a new frame, the strong classifier is applied to each candidate position $L_p(p = 1,\ldots,P)$, and we get the new location p^+ of the target by (4). Then we need some samples to update the weak classifiers. Recall that the positive sample is chosen as the target patch, and the negative samples are picked up fixedly (left top, right top, left bottom, and right bottom) in the surrounding background of the target center in [1], as shown in Figure 2a. This samples selection mechanism often relies excessively on classification confidence map produced by the strong classifier in (4), which can be seen as a drawback of the method. The negative samples are also selected randomly far away from the target in other literatures [4,5,11], but the randomness cannot deal well with the similar object nearby or background clutter.

We use the saliency map to select more accurate positive sample and also more discriminative negative samples to compensate the uncertainty caused by p^+. Given the saliency map $S(x)$ of the current frame, we can get the saliency confidence map$SC(x)$:

$$SC(x) = S(x)^* f \tag{10}$$

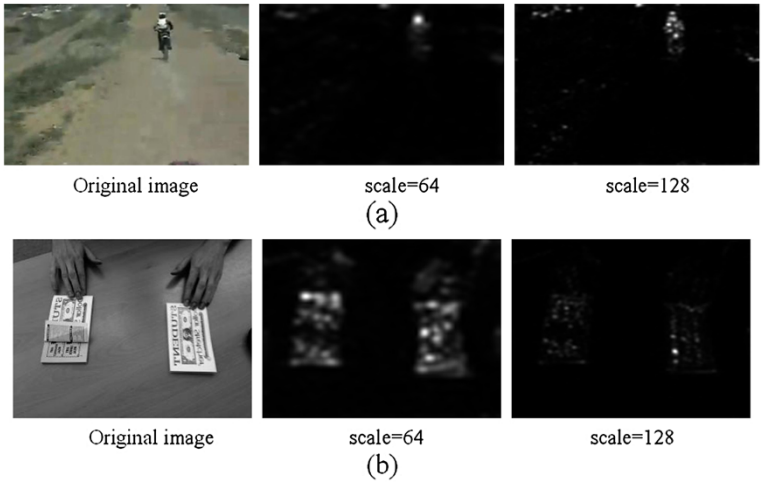

Figure 3 Different saliency maps of two scales. (**a**) The saliency maps of one frame in "Motorbike" sequences. (**b**) The saliency maps of one frame in "Coupon book" sequences.

where f denotes an $h_t \times w_t$matrix defined by

$$f = \frac{1}{h_t \times w_t}\begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix} \tag{11}$$

Then $SC(x)$ is normalized into the range 0 to 1 for each pixel (or location), as can be seen in Figure 4b. Red color pixels denote large saliency confidence values and blue ones denote low values.

Let $HC(x)$ be the confidence map produced by strong classifier in the early detecting period. $HC(x)$ is also normalized into the range of 0 to 1 using $\max\left(\sum_{n=1}^{N}\alpha_n h_n^{sel}(x)\right)$. The positive sample P_{pos} is selected through combining two confidence maps by adaptive weights

$$w^{HC} = \frac{HC(L_p)}{SC(L_p) + HC(L_p)} \tag{12}$$

$$w^{SC} = \frac{SC(L_p)}{SC(L_p) + HC(L_p)} \tag{13}$$

$$P_{pos} = \arg\max_{p}\left(w^{HC} \times HC(L_p) + w^{SC} \times SC(L_p)\right) \tag{14}$$

where w^{HC} and w^{SC} are the weights of HC and SC, respectively, L_p are the pixels inside the samples extracting region. Recall that $w^{HC} = 1$ and $w^{SC} = 0$ in [1], which is the main cause of drift problem when the target appearance changes dramatically. Our approach combines two individual confidence map by (14), which is more robust than only depending on HC. Some positive samples obtained by the above process in "Gymnastics" sequence are shown in Figure 5b. We can see that the samples obtained by us can adapt to the appearance change of the target and always locate on the center of the target. However, the samples obtained in [1] contain too much background information due to the only one cue (as shown in Figure 5a).

As to negative samples, we provide two criteria for the selection such that

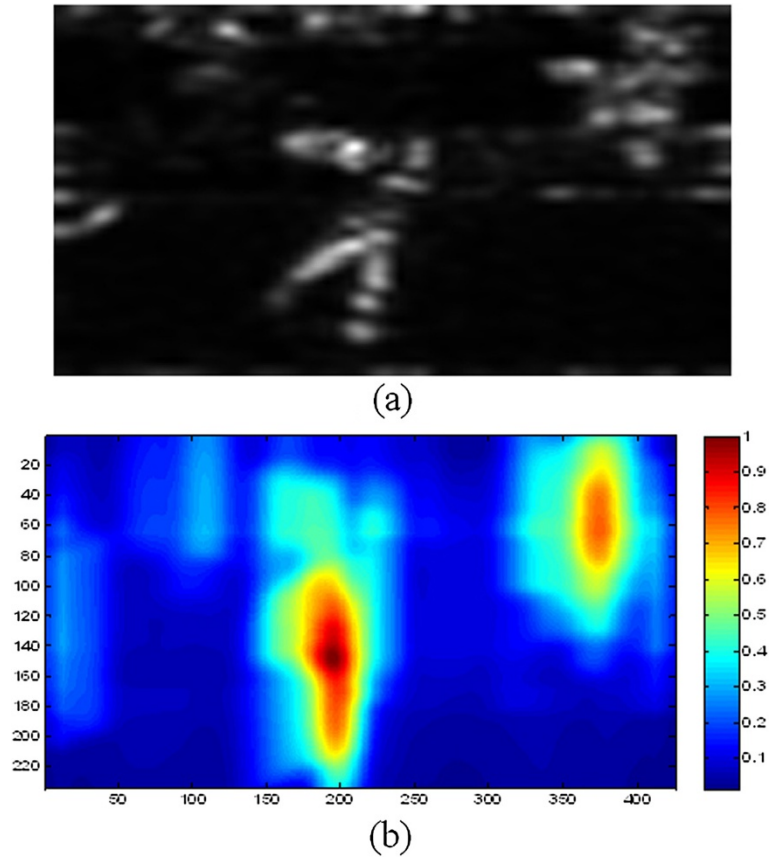

Figure 4 Example of a saliency map and its corresponding saliency confidence map. (**a**) The saliency map produced by the method in Section 3. (**b**) The corresponding saliency confidence map.

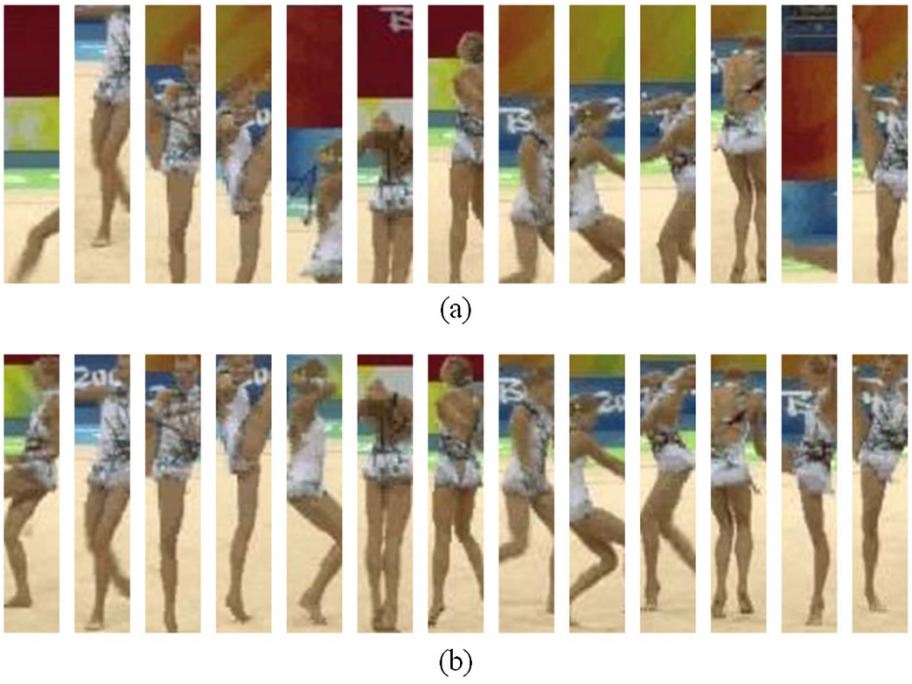

Figure 5 Some positive samples obtained by two algorithms. (**a**) Positive samples obtained by the method in [1]. (**b**) Positive samples obtained by our proposed algorithm.

Criterion 1: Negative samples should have low saliency confidence values.

Criterion 2: Negative samples should contain those similar objects which also have high confidence values computed by the strong classifier around the target.

The above criterions check that the selected negative samples to be used to update the weak classifiers should contain less information of the target. Furthermore, they should also be the indicators of the surrounding confusors. The criterions are formulated by

$$P_{neg}^{i} = \arg\max_{p}\left(\frac{HC\left(L_p^i\right)}{HC\left(L_p^i\right)+1-SC\left(L_p^i\right)}\times HC\left(L_p^i\right) + \frac{1-SC\left(L_p^i\right)}{HC\left(L_p^i\right)+1-SC\left(L_p^i\right)}\times\left(1-SC\left(L_p^i\right)\right)\right) \tag{15}$$

where $i \in \{1, 2, 3, 4\}$ denotes the index of four search regions which are divided near the target area as illustrated in Figure 2b, and L_p^i are the locations (or pixels) inside the ith region. Comparing with the fixed or random mechanism, our approach is more reasonable and adaptive, as shown in dotted rectangles in Figure 2b.

4.4. Discussion

The benefits of our approach are also illustrated in Figure 1. Due to the inaccurate detected target location and the fixed selection positions in [1], the positive sample is suboptimal, but one of the negative samples is the actually the "tracked target". In our adaptive selection mechanism, the "tracked target" is finally selected as the positive sample, which leads to better update of the weak classifiers.

The cooperation of the saliency detection is the prime characteristic of our tracking algorithm. It should be noted that the saliency detection is an improved version from the original method in [13], because we have exactly known the size of the salient target. The scale of the down-sampled image could be selected in a more reasonable way to improve the saliency map.

There are many ways to spatially divide the search region to select the negative samples, and we just use a simple way. The locations for extracting positive sample is set to be inside the target area, and the four search regions for extracting negative samples are set to $(w_t \times h_t)/4$ with the four corners of the target area as the centers, which are demonstrated in black/green/blue/yellow blocks, respectively, in Figure 2b.

5. Experiments

We compared the proposed algorithm with five different online trackers: online AdaBoost(OAB) [1], semi-supervised boosting tracker(SemiB) [7], MIL tracker (MILTrack) [2],

the TLD tracker [8], and the compressive tracker(CT) [4]. For fair comparison, we use the source codes provided by these authors with the original parameters. Since all of the trackers and our approach involve randomness in either features extracting process or samples selection process, we run them five times and obtain the average results for each video. In experiments, we use the most challenging video sequences from the publicly available sequences [2,8,21,22]. Table 1 presents the sequences and their major challenges.

The number of weak classifiers is 300, and the number of selectors is 40 in our tracking algorithm. We set $\theta = 0.03$ in determining the scale of the down-sampled images. Finally, the search region to detect the target in the new frame is set to 1.5 times of the target size and the search region to select the samples is set in the way described in Section 4.4. To prove the validity of our algorithm, these parameters are fixed for all the sequences.

We use the center location errors measured with labeled ground truth data to evaluate our proposed algorithm with the above-mentioned five trackers. For thorough investigation, we draw the error curves for each sequence in Figure 6, and the results are summarized in Table 2. TLD is able to redetect the target during the tracking, but it is easy to lose the target, so we only show the errors for the sequences that TLD keep track the target through the tracking. We note that our proposed algorithm achieves five bests and one second bests in all the sequences. In addition, Figure 7 shows tracking results of the above trackers and some salient maps obtained in our algorithm. More details of experiments will be discussed below.

5.1. Out of plane rotation and pose variation

The sequence "Gymnastics" in Figure 7b shows a gymnast who rotates with 360°out of plane and undergoes drastic geometric appearance change. The CT, TLD, SemiB, and OAB lose the target in the frame #154, #248, #315, and #494, respectively. Note that the TLD relocates the target in the frame #324 (see Figure 6b). Only our approach and MILTrack can keep track all the time as both our approach and MILTrack are designed to handle the target location ambiguity. Moreover, the error plot of our approach is lower than that of MILTrack. The CT extracts many positive samples close to the target center; the OAB and SemiB use samples only depending on the target center, which degrade the classification performance of weak classifiers and lead to drift. The proposed tracker is robust to the out of plane rotation and pose changes as the salient maps are combined with the strong classifiers to select the most correct positive samples even when the previous target locations are not exact and the samples are used to update the weak classifiers to separate the target and background well. Figure 7b illustrates the advantage of our tracker. In addition, our tracker performs well on the "Sylvester" sequence in which the target undergoes significant pose variation (see Figure 7c). The SemiB performs poorly as it relies strongly on the target information in the first frame, and cannot keep up with the variation of the target appearance.

5.2. Background clutter and similar object

For the "Mountain-bike" sequence shown in Figure 7a, the orientation and the appearance of the target changes gradually, and the surrounding background has similar texture. As TLD does not take the background samples into account, it is easy to be distracted by the similar objects in the background (see in Figure 6a). All the other trackers except our approach are also distracted by the background. With the help of the saliency maps in the bottom row of Figure 7a, the distractions are significantly weakened (see frames #67 and #204). The target in "Coupon book" sequence undergoes appearance change suddenly at the beginning, and all the trackers except the CT can keep track of the target correctly. Then a similar target appears near the tracked target in frame #134, and the OAB is influenced by the similar target. At the same time, the SemiB locates on another target completely. Although the TLD keeps up with the right target correctly in the first 194 frames, it locates on the confusor suddenly in frame #194. The reason for their failure is partly that the confusor now has more similar texture with the prior information.

Our algorithm is able to track the right targets perfectly in the above two sequences because it has selected the similar objects as the negative samples online to update more discriminative classifiers.

Table 1 Tracking sequences used in experiments

Sequences	Challenges	Frame size (pixel)	Object (pixel)	γ	Scale
Mountain-bike	Background clutter	640 × 360	66 × 56	0.016	128
Gymnastics	Out of plane rotation	426 × 234	26 × 130	0.033	64
Sylvester	Appearance changes	320 × 240	50 × 52	0.033	64
Coupon book	Similar object	320 × 240	62 × 98	0.079	64
Motorbike	Cameral motion	470 × 310	32 × 64	0.014	128
Car	Occlusions	290 × 217	100 × 32	0.05	64

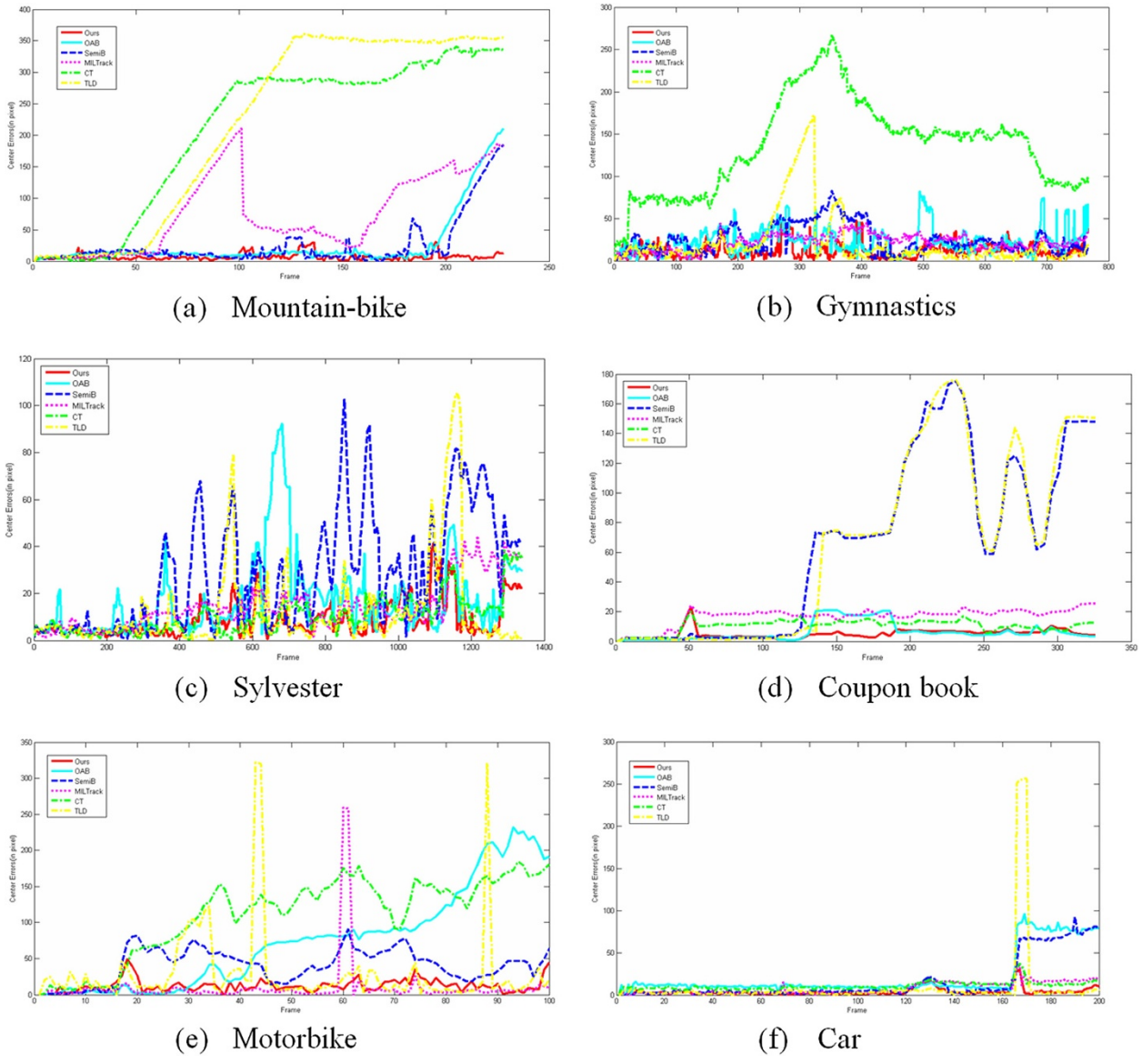

Figure 6 Center location errors of our method and other five different algorithms. (**a**) Mountain-bike. (**b**) Gymnastics. (**c**) Sylvester. (**d**) Coupon book. (**e**) Motorbike. (**f**) Car.

Table 2 Center location errors (in pixel)

Sequence	Ours	OAB	SemiB	MILTrack	CT	TLD
Mountain-bike	**7**	27	*25*	74	210	–
Gymnastics	**12**	23	25	*21*	136	–
Sylvester	**9**	17	28	15	*10*	13
Coupon book	**4**	*6*	66	17	10	65
Motorbike	*12*	74	40	**10**	110	–
Car	**3**	22	16	*8*	10	–

Bold type indicates the best performance, italic type indicates the second best.

5.3. Occlusion and cameral motion

The "Car" sequence in Figure 7f shows a car contains occlusion and cameral motion. The target is almost fully occluded in frame #165. After the occlusion (in frame #169), all the OAB, SemiB, and TLD lose the target completely. Note that SemiB locates on another car again. We can see that only our algorithm catches up the target correctly as our exception. The reason for the good performance of our tracker is that it selects correct positive and negative samples and our tracker can alleviate the influence caused by the similar objects.

Among all the testing sequences, "Motorbike" includes most challenges such as drastic motion blur and cameral

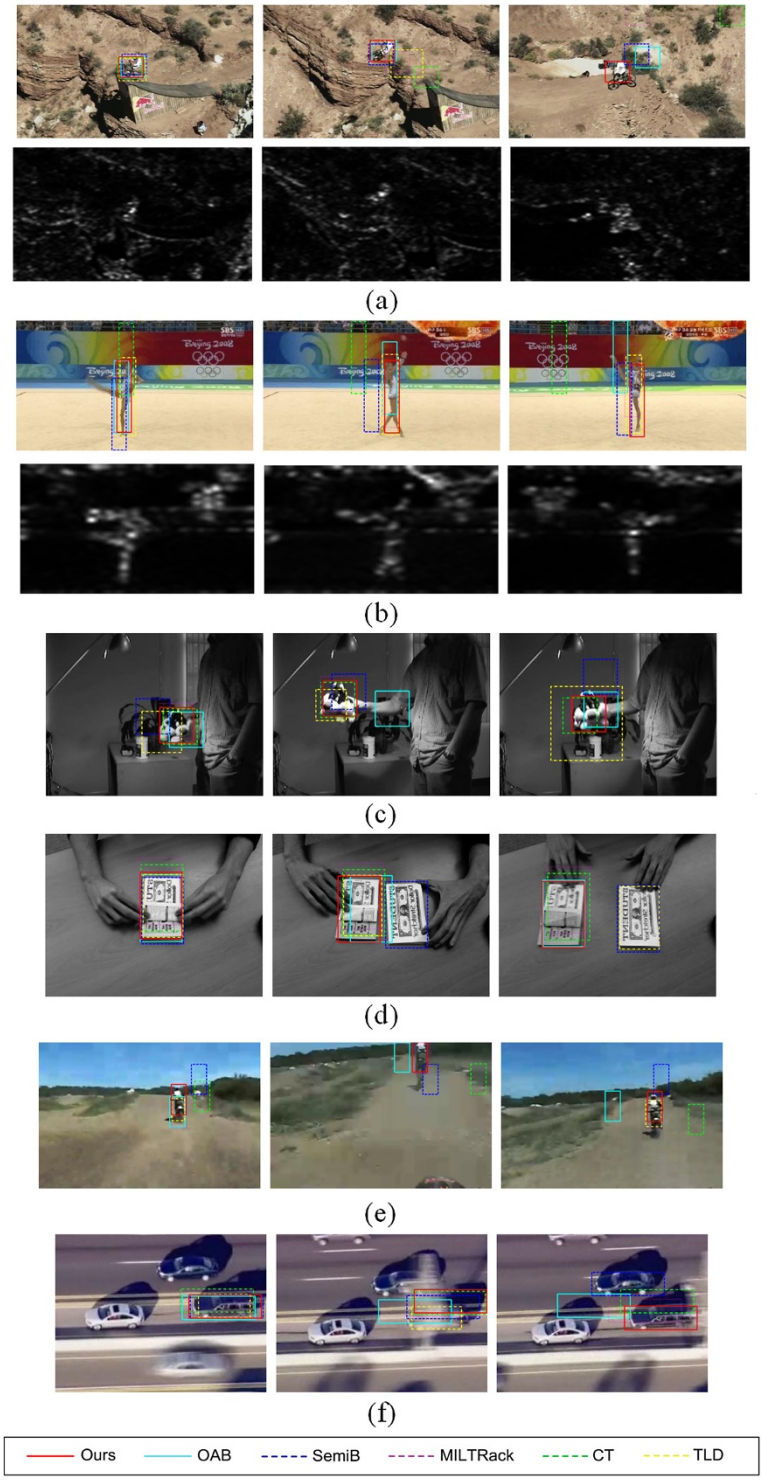

Figure 7 Tracking results of our tracker and other five different trackers. (**a**) Mountain-bike. Frame #32, #67, and #204. (**b**) Gymnastics. Frame #28, #182 and #503. (**c**) Sylvester. Frame #521, #672 and #931. (**d**) Coupon book. Frame #52, #134 and #194. (**e**) Motorbike. Frame #20, #42 and #71. (**f**) Car. Frame #142, #165 and #169.

motion as shown in Figure 7e. Due to the severe motion abruptly, all the online trackers are unable to get the correct target locations during the tracking. Some trackers (OAB, SemiB, and CT) lose the target completely, and some other trackers (MILTrack and TLD) lose the target in some frames but relocate the target again. However, our tracker never loses the target and outperforms the other trackers.

Our proposed tracking algorithm achieves performances more robust than SemiB and TLD in dealing with the background clutter and confusors, and outperforms OAB, MILTrack, and CT in the sequences that contain severe out of plane rotation, pose variation, and cameral motion. The reason is that our tracker is not only able to pick out most correct positive samples in spite of the inaccurate target location, but also can select the adaptive negative samples which contain the potential threats in the surrounding background to update the weak classifiers. Both MILTrack and TLD are also good trackers. However, MILTrack doe not take the negative samples selection into account, which degrades its performance. As to TLD, it is able to learn a robust detector to relocate the target after the drastic appearance change and abrupt cameral motion, but it is easy to lose the target completely in several sequences. In a word, our approach is an accurate and stable online tracker.

6. Conclusion

In this article, we proposed the use of saliency detection method for robust online tracking regarding the drift problem. The proposed scheme employs saliency confidence map and classification confidence map to select the reliable positive samples to adapt to the target's appearance variation, as well as the reasonable negative samples to handle the background clutter and similar objects. The weak classifiers are updated with the obtained samples in the online boosting framework. We employ a simple saliency detection method and analysis the relationship between the scale of the down-weighted images and the size of the tracked target to produce saliency maps. Numerous experiments with state-of-the-art algorithms on challenging sequences demonstrate that the proposed algorithm performs well in terms of accuracy and stability.

Competing interests
The authors declare that they have no competing interests.

Received: 27 September 2012 Accepted: 12 December 2012
Published: 15 January 2013

References

1. H Grabner, M Grabner, H Bischof, Real-time tracking via online boosting, in *Proceedings of British Machine Vision Conference (BMVC)*, ed. by (Edinburgh, 2006), pp. 47–56
2. B Babenko, M-H Yang, S Belongie, Visual tracking with online multiple instance learning, in *IEEE Proceedings of the CVPR*, ed. by (Miami, 2009), pp. 983–990
3. M Godec, Hough-based tracking of non-rigid objects, in *IEEE International Conference on Computer Vision*, ed. by (Barcelona, 2011), pp. 81–88
4. K Zhang, L Zhang, MH Yang, Real-time compressive tracking, in *European Conference on Computer Vision (ECCV)*, ed. by (, Firenze, 2012)
5. S Avidan, Ensemble tracking. IEEE Trans. Pattern Anal. Mach. Intell. **29**(2), 261–271 (2007)
6. A Saffari, C Leistner, J Santner, M Godec, H Bischof, On-line random forests, in *IEEE International Conferenceon Computer Vision Workshops*, ed. by (Xi'an, 2009), pp. 1393–1400
7. H Grabner, C Leistner, H Bischof, Semi-supervised on-line boosting for robust tracking, in *European Conference on Computer Vision (ECCV)*, ed. by (Marseille, 2008), pp. 234–247
8. Z Kalal, J Matas, K Mikolajczyk, P-n learning: boostrapping binary classifier by structural constraints, in *IEEE Proceedings of the CVPR*, ed. by (San Francisco, 2010), pp. 49–56
9. Q Yu, TB Dinh, G Medioni, Online tracking and reacquisition using co-trained generative and discriminative trackers, in *European Conference on Computer Vision (ECCV)*, ed. by (Marseille, 2008), pp. 678–691
10. R Liu, J Cheng, H Lu, A robust boosting tracker with minimum error bound in a co-training framework, in *IEEE International Conference on Computer Vision*, ed. by (Xi'an, 2009), pp. 1459–1466
11. H Lu, Q Zhou, D Wang, X Ruan, A co-training framework for visual tracking with multiple instance, in *Automatic Face and Gesture Recognition*, ed. by (2011), pp. 539–544
12. L Itti, C Koch, E Niebur, A model of saliency-based visual attention for rapid scene analysis. IEEE Trans. Pattern Anal. Mach. Intell. **20**(11), 1254–1259 (1998)
13. X Hou, L Zhang, Saliency detection: a spectral residual approach, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, ed. by (Minnesota, 2007), pp. 1–8
14. MM Cheng, GX Zhang, NJ Mitra, H Xiaolei, SM Hu, *Global contrast based salient region detection, inIEEE Proceedings of the CVPR* (Springs, Colorado, 2011). pp. 409–416
15. R Achanata, S Hemami, F Estrada, S Susstrunk, Frequency-tuned salient region detection, in *IEEE Proceedings of the CVPR*, ed. by (Miami, 2009), pp. 1597–1604
16. B Ko, J Nam, Object-of-interest image segmentation based on human attention and semantic region clustering. J. Opt. Soc. Am. **23**, 2462–2470 (2006)
17. D Walther, L Itti, M Riesenhuber, T Poggio, C Koch, Attentional selection for object recognition—a gentle way. Lect. Notes Comput. Sc. **2525**, 472–479 (2002)
18. H Wu, YS Wang, KC Feng, TT Wong, TY Lee, PA Heng, Resizing by symmetry-summarization. ACM Trans. Graph. **29**, 1–9 (2010)
19. Y Freund, RE Schapire, A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. **55**, 119–139 (1997)
20. N Oza, S Russell, Online bagging and boosting, in *Proceedings of Artificial Intelligence and Statistics*, ed. by (2001), pp. 105–112
21. DA Ross, J Lim, RS Lin, MH Yang, Incremental learning for robust visual tracking. Int. J. Comput. Vis **77**, 125–141 (2008)
22. J Kwon, K Lee, Tracking of a non-rigid object via patch based dynamic appearance modeling and adaptive basin hopping monte carlo sampling, in *IEEE Proceedings of the CVPR*, ed. by (Miami, 2009), pp. 1208–1215

doi:10.1186/1687-6180-2013-6
Cite this article as: Yan *et al.*: **Robust online tracking via adaptive samples selection with saliency detection.** *EURASIP Journal on Advances in Signal Processing* 2013 **2013**:6.