SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

July 2016, Vol. 59 072102:1–072102:15
doi: 10.1007/s11432-015-5424-5

Efficient compressive sensing tracking via mixed classifier decision

Hang SUN, Jing LI*, Jun CHANG*, Bo DU & Zhenyang SU

Computer School, Wuhan University, Wuhan 430072*, China*

Received October 12, 2015; accepted December 29, 2015; published online June 16, 2016

Abstract Recent years have witnessed successful use of tracking-by-detection methods, with a number of promising results being achieved. Most of these algorithms use a sliding window to collect samples and then employ these samples to train and update the classifiers. They also use an updated classifier to establish the appearance model and they take the maximum response value of the classifier as the location of the target within a fixed radius. Compressive Tracking (CT) is a novel tracking-by-detection algorithm that updates the appearance model in a compressed domain. However, the conventional CT algorithm uses a single classifier to detect the target, and if the selected region drifts, the classifier may become inaccurate. Furthermore, the CT algorithm updates the classifier parameters with a constant learning rate. Therefore, if the target is completely occluded for an extended period, the classifier will instead learn the features of the covered object and the target will ultimately be lost. To overcome these problems, we present a compressive sensing tracking algorithm using mixed classifier decision. The main improvements in our algorithm are that it adopts mixed classifiers to locate the target and it applies a dynamic learning rate to update the appearance model. An experimental comparison with state-of-the-art algorithms on eight benchmark video sequences in complicated situations shows that the proposed algorithm achieves the best performance with 12 pixels on the average center location error and 66.82% on the average overlap score.

Keywords compressive sensing, object tracking, mixed classifier decision, dynamic learning rate, appearance model update

Citation Sun H, Li J, Chang J, et al. Efficient compressive sensing tracking via mixed classifier decision. Sci China Inf Sci, 2016, 59(7): 072102, doi: 10.1007/s11432-015-5424-5

1 Introduction

Object tracking is a topic of great current importance in the field of computer vision [1,2] owing to its wide applications in areas such as automated surveillance, video indexing and traffic monitoring, to name just a few. The task of tracking is to estimate the target states in video sequences under the condition that the initial state of the target in the start frame is already known from a sequence of measurements made on the object. Although many detecting and tracking algorithms [3–25] have been proposed, no single algorithm has been able to solve all the scenario problems that arise. This is because there are numerous factors that may lead to changes in appearance and affect the performance of a tracking algorithm, such

* Corresponding author (email: leejingcn@163.com, chang.jun@whu.edu.cn)

© Science China Press and Springer-Verlag Berlin Heidelberg 2016

as pose adjustment, variations in illumination and occlusion. Therefore, the establishment of a new object tracking system remains a challenging task with regard to efficiency, accuracy and robustness.

During the last few years, tracking algorithms based on an effective adaptive appearance model have attracted a great deal of attention. In general, tracking algorithms can be categorized as either generative or discriminative, based on the appearance models that they employ. A generative model first learns an appearance model to represent the target object and then uses it to determine the location of the target with the minimum error in subsequent frames. Hence, the construction of an efficient and robust appearance model is crucial for the performance of a generative model in object tracking. Adam et al. [9] developed an appearance model using some information fragments to deal with pose variation and partial occlusion. The l1-tracker [10] uses a sparse linear combination of the target and trivial templates to establish a target model with the aim of solving the problems caused by pose variation, changes in illumination and partial occlusion. However, the computational complexity of this tracker is rather high, limiting its application in real-time scenarios. Li et al. [11] effectively optimized the performance of the l1-tracker in real-time applications by employing an orthogonal matching pursuit algorithm. To enhance the discriminative power of the appearance model, Liu et al. [12] proposed a model based on sparse coding. Jia et al. [13] made further improvement by using partial sparse coding to express the appearance model with all the target templates being decomposed into a group of smaller image blocks to make full use of local and spatial information on the objects. However, both of these tracking methods ignore the connections among sparse representations. Zhang et al. [14] proved that tracking results can be refined by applying correlations between each sparse representation, and they also demonstrated the superior performance of the multi-task tracker. Although these algorithms have given good results in some circumstances, several problems remain. First, algorithms based on generative models require a variety of training samples for the initial learning of an appearance model. Second, they discard useful background information that could have been exploited to separate the target from the background more efficiently. Discriminative models pose object tracking as a detection problem in which a binary classification is learned and then used to sort the target object from its surrounding background within a local region. Much researches have been carried out in this direction. Collins et al. [15] showed that tracking performance can be effectively improved by selecting discriminative features in an online tracking algorithm. Babenko et al. [16,17] introduced multiple-instance learning into online tracking with samples being considered to lie within positive and negative bags or sets. To correct errors in detection, Kalal et al. [18] proposed the PN-learning algorithm for exploiting the underlying structure of positive and negative samples to learn effective classifiers for object tracking.

Recently, compressive sensing techniques have attracted much attention in many fields. Zhang et al. [26] proposed an effective tracking algorithm in the context of compressive sensing theories, namely, Compressive Tracking (CT). This method considers target tracking as a detection problem based on a binary classification and thus it is a discriminative-based model. It has been proved that discriminative information present in high-dimensional features of a multi-scale image can be preserved in low-dimensional features extracted at random from the image. Thus, real-time tracking can be realized through dimensional reduction. This discovery has greatly accelerated the development of target tracking research.

Despite the fact that the CT algorithm has achieved great success in applications, problems still exist. First, the algorithm uses an already-trained classifier to estimate the position of the target object in the next frame. In this situation, if the object is influenced by occlusion, changes in appearance, variations in illumination or other factors, the maximum response value of the classifier may be less than zero. As a consequence, it is inappropriate to take the maximum response value of the classifier as the target location or to label positive and negative samples around the target. Second, the CT algorithm updates the classifier parameters at a constant learning rate. Under this condition, if the target is occluded for an extended period, the classifier may learn the features of the covering object and the target will ultimately be lost.

In this paper, we propose a CT algorithm based on mixed classifier decision; it is a discriminative model-based method that takes target tracking as a detection problem based on a binary classification. Our proposed algorithm is selective in choosing different classifiers for object tracking, and it adopts

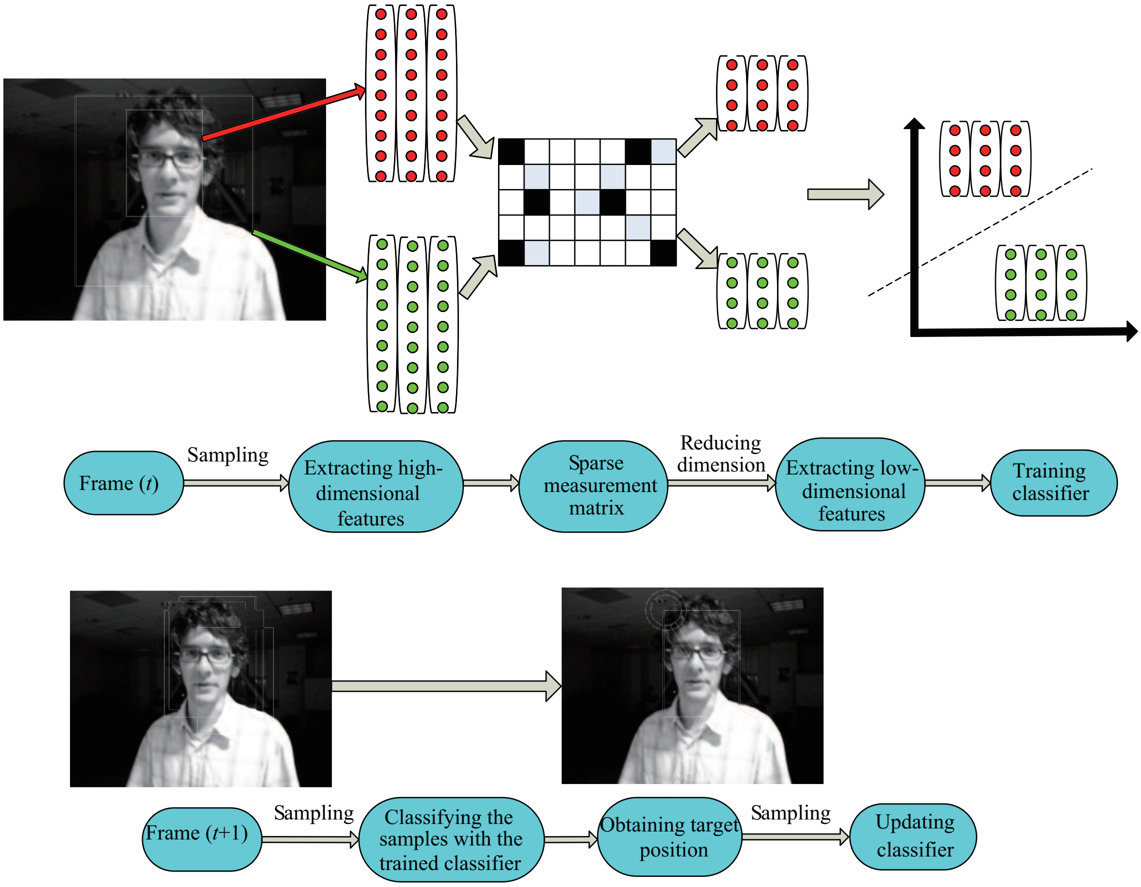
Frame (*t*)
Sampling
Extracting high-dimensional features
Sparse measurement matrix
Reducing dimension
Extracting low-dimensional features
Training classifier
Frame (*t*+1)
Sampling
Classifying the samples with the trained classifier
Obtaining target position
Sampling
Updating classifier

Figure 1 (Color online) Structure of the compressive tracking algorithm.

different strategies to update the classifiers. This means that it is better able to handle problems caused by variations in illumination, by occlusion and by background clutter. The remainder of the paper is organized as follows. Section 2 introduces the CT algorithm. Section 3 presents the selection principle for the classifiers and the updating rules for both the classifiers and the dynamic learning rate in the proposed algorithm. Section 4 provides details of the experiment setup and results. Finally, Section 5 concludes the paper.

2 Compressive tracking

The CT algorithm is a simple but efficient tracking algorithm based on compressive sensing theories. Its main idea relies on dimensional reduction. The algorithm uses a very sparse measurement matrix that satisfies the restricted isometry property (RIP) to project the original image into the feature space to give a compressed low-dimensional subspace that preserves the features of the original high-dimensional space. Thus, we can extract the foreground object as well as the background information via a sparse measurement matrix and use them as the positive and negative samples for online learning. We then classify the test target image in the next frame through a naive Bayes classifier.

The main structure of the algorithm is shown in Figure 1. We first take a set of images in frame(t) as samples and extract the low-dimensional features of these samples on the basis of compressive sensing theory. To deal with the multi-scale problem, the CT algorithm represents each sample $z \in R^{w\times h}$ by convolving it with a set of rectangular filters at multiple scales $\{h_{1,1},\ldots,h_{w,h}\}$, where $h_{i,j}$ is defined as

$$h_{i,j}=\begin{cases}1, & 1\leqslant x\leqslant i,\ 1\leqslant y\leqslant j,\\ 0, & \text{otherwise},\end{cases} \tag{1}$$

where, i and j are respectively the width and height of the rectangular filter, and the sample $z \in R^{w\times h}$ is a multi-scale sample represented as $x = \{x_1, \ldots, x_m\} \in R^m$, with $m = (wh)^2$. The dimensionality m is typically in the range $10^6 \sim 10^8$. We then use a random Gaussian matrix $R \in R^{n\times m} (n << m)$ satisfying the RIP to project a high-dimensional multi-scale sample z into a low-dimensional sample:

$$v = Rx. \tag{2}$$

According to the theory of compressive sensing, we can obtain x with minimum error and a particular high probability through the low-dimensional vector $v = (v_1, \ldots, v_n) \in R^n$, which is generated from the projection and dimensional reduction of the high-dimensional multi-scale sample z.

For each high-dimensional sample z, we assume that all the elements in low-dimensional representation v are independently distributed and that the prior probability $P(y = 1) = P(y = 0)$. We then model v with a naive Bayes classifier,

$$H(v) = \log\left(\frac{\prod_{i=1}^{n} p(v_i|y=1)p(y=1)}{\prod_{i=1}^{n} p(v_i|y=0)p(y=0)}\right) = \sum_{i=1}^{n} \log\left(\frac{p(v_i|y=1)}{p(v_i|y=0)}\right), \tag{3}$$

$$p(v_i|y=1) \sim N(\mu_i^1, \sigma_i^1),\ \ p(v_i|y=0) \sim N(\mu_i^0, \sigma_i^0), \tag{4}$$

where (μ_i^1, σ_i^1) and (μ_i^0, σ_i^0) are the mean and variance of the positive and negative samples, respectively. For the positive samples (μ_i^1, σ_i^1) can be updated according to

$$\begin{aligned} &\mu_i^1 \leftarrow \lambda\mu_i^1 + (1-\lambda)\mu^1, \\ &\sigma_i^1 \leftarrow \sqrt{\lambda(\sigma_i^1)^2 + (1-\lambda)(\sigma^1)^2 + \lambda(1-\lambda)(\mu_i^1 - \mu^1)^2}, \end{aligned} \tag{5}$$

where, λ is the learning rate adopted to control the degree of freshness of the mean and variance of the samples. For the negative samples (μ_i^0, σ_i^0) can be updated in the same way.

3 Updating of the target model based on mixed classifier decision

In this section, we present the details of our compressive sensing tracking algorithm via mixed classifier decision. The tracking task is essentially completed and formulated as a dynamic sample state estimation problem and usually involves updating of the model. The decision model adopted in our algorithm is shown in Figure 2. When the maximum response value of the classifier is less than zero, we preserve the original classifier and initialize a new classifier in the current frame. In this way, two classifiers are obtained. We can then selectively use these two classifiers for object tracking. However, the two classifiers need to be updated in different ways in order to track the target object correctly.

3.1 Mixed classifier decision model

Adaptive updating is used in a variety of tracking-by-detection algorithms to update models [27–32]. This is the case in particular in compressive sensing tracking, which uses (5) for updating. This target updating method combines the fixed parameter reference templates extracted from the first frame with subsequent tracked targets to update the target models. Its advantage is that it can dynamically reflect changes in the appearance of the targets. However, it uses a single classifier and updates the classifier with a constant learning rate, and if the target is subject to occlusion or variations in illumination, then the performance of its classifier will degrade, and finally the selected region will drift and the target will be lost.

The CT algorithm is basically a tracking method based on a discriminative model. It learns a binary classifier and applies it to classification of samples. The response value of this binary classifier $H(v)$ is defined as

$$\sigma(H(v)) = \frac{1}{1 + \mathrm{e}^{-H(v)}}. \tag{6}$$

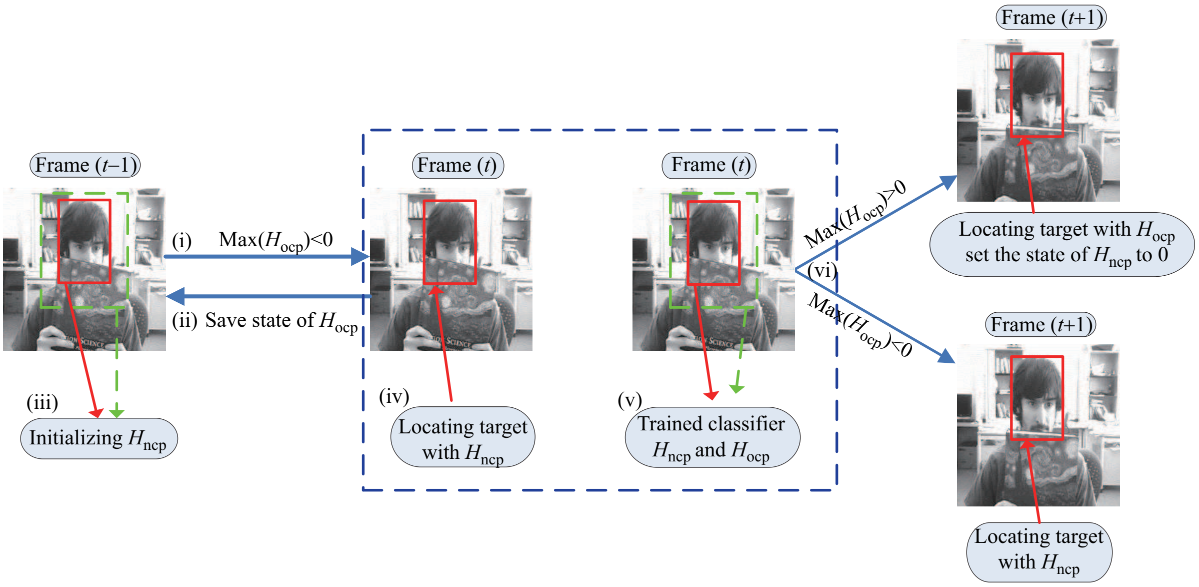
Frame (t–1)
(i) Max(H_{ocp})<0
(ii) Save state of H_{ocp}
(iii)
Initializing H_{ncp}
Frame (t)
(iv)
Locating target with H_{ncp}
Frame (t)
(v)
Trained classifier H_{ncp} and H_{ocp}
Max(H_{ocp})>0
(vi)
Max(H_{ocp})<0
Frame (t+1)
Locating target with H_{ocp} set the state of H_{ncp} to 0
Frame (t+1)
Locating target with H_{ncp}

Figure 2 (Color online) Tracking model based on mixed classifier decision.

After the classifier has differentiated all samples, we draw the one with the highest score, namely, $\max(\sigma(H(v)))$ as the result of current tracking. According to (6), when $\sigma(H(v)) > 0.5$, the corresponding classifier $H(v) > 0$, and the corresponding sample is labeled as positive. Conversely, when $\sigma(H(v)) < 0.5$, $H(v) < 0$, and the sample is labeled as negative. From the perspective of the Bayes classifier, if the maximum response value of the classifier is less than zero, i.e., $\max(H(v)) < 0$, then the classifier will label all the test samples as negative. Under this condition, it is not appropriate to take the negative samples with maximum likelihood as the target location. Furthermore, if we label positive and negative sample regions around the determined object location and then train the classifier, the performance of the classifier will degrade. The shortcomings of the CT algorithm are that it uses a single classifier for target location and a constant learning rate to update this classifier. With the aim of overcoming these two defects, this paper designs a new tracking algorithm based on mixed classifier decision. The main steps of this algorithm are shown in Figure 2.

To handle the problems of inaccurate location of the target and performance degradation of the classifier when the maximum response value of the original classifier is less than zero, we define two classifiers,

$$\begin{cases} H_{\rm ocp} = \{\mu_o^1, \sigma_o^1, \mu_o^0, \sigma_o^0\}, \\ H_{\rm ncp} = \{\mu_p^1, \sigma_p^1, \mu_p^0, \sigma_p^0\}. \end{cases} \tag{7}$$

As shown in Figure 2, in the process of tracking, the original classifier $H_{\rm ocp}$ locates the position of the target from frame$(t-1)$ to frame(t). If the maximum response value of the original classifier is less than zero and the classifier classifies all the samples as negative, then the target object may be influenced by factors such as serious occlusion and changes in appearance. Under this condition, the classifier $H_{\rm ocp}$ cannot correctly estimate the location of the real target object. However, this does not mean that $H_{\rm ocp}$ will work less than ideally in estimating target location in the subsequent frames. Therefore, to track the occluded target, we preserve the state parameters $\{\mu_o^1, \sigma_o^1, \mu_o^0, \sigma_o^0\}$ of $H_{\rm ocp}$ and initialize a new classifier $H_{\rm ncp}$ with the positive samples $\{\mu_p^1, \sigma_p^1\}$ and negative samples $\{\mu_p^0, \sigma_p^0\}$ that result from sampling in frame$(t-1)$, we thus have two classifiers $H_{\rm ocp}$ and $H_{\rm ncp}$. The first is related to the target state in a succession of frames from frame(1) to frame$(t-1)$, while the second is related to the target state influenced by several interference factors in frame$(t-1)$.

After obtaining two classifiers in frame$(t-1)$, we then need to locate the target in frame(t) with a classifier. We define the position of the target in frame(t) as

$$L_t^* = \max(H(v)). \tag{8}$$

Table 1 Updating strategy for classifiers

No.	State of $H_{\rm ocp}$	Learning rate of $H_{\rm ncp}$	Learning rate of $H_{\rm ncp}$	Learning rate of $H_{\rm ocp}$	Condition in frame(t)
1	Update	Initialization and decision	λ	$\lambda_{\rm occ}$	$\max(H_{\rm ocp}(v)) < 0$ && flag $= 0$
2	Update	Update and decision	λ	$\lambda_{\rm occ}$	$\max(H_{\rm ocp}(v)) < 0$ && flag $= 1$
3	Update and decision	–	–	λ	$\max(H_{\rm ocp}(v)) > 0$ && flag $= 0$
4	Update and decision	Set 0	–	λ	$\max(H_{\rm ocp}(v)) > 0$ && Cflag $= 1$

As classifier $H_{\rm ocp}$ fails to accurately locate the target in frame(t), we use the newly trained classifier $H_{\rm ncp}$ to estimate the target state in the current frame and define the location of the target in frame(t) as $L_t^* = \max(H_{\rm ncp}(v))$. It should be noted that what classifier $H_{\rm ncp}$ locates is a phony target influenced by some interference information. In the following tracking process, we first apply the original classifier $H_{\rm ocp}$ to test each frame and then use different classifiers to locate the target in line with the different maximum response values. Thus the decision model of the mixed classifiers can be defined as follows:

$$\begin{cases} L_t^* = \max(H_{\rm ncp}(v)), & \max(H_{\rm ocp}(v)) < 0, \\ L_t^* = \max(H_{\rm ocp}(v)), & \max(H_{\rm ocp}(v)) > 0. \end{cases} \tag{9}$$

3.2 Updating strategy for classifiers

In practical applications, the appearance of the targets often changes under the influences of a number of factors, such as occlusion and variations in illumination. The updating strategy for the classifier is therefore crucial for the performance of the tracking algorithm. The CT algorithm updates the classifier parameter with a certain fixed value. This indicates that no matter whether or not the target is within the objective frame, the tracker will always take the content within the objective frame as the true target and update the classifier with a constant learning rate. The learning rate λ is a parameter that measures the freshness of the classifier. The higher the value of λ, the less is the influence of sample training on the classifier, and vice versa. If the maximum response value of the original classifier is less than zero, i.e. $\max(H_{\rm ocp}(v)) < 0$, then the target to be tracked is influenced by some disruptive factors. In this case, to reduce the influence exerted by the updated classifier, we set a new learning rate $\lambda_{\rm occ}$ for the updating of the classifier $H_{\rm ocp}$ and we update the learning rate in the following incremental and iterative fashion:

$$\lambda_{\rm occ} = \lambda_{\rm occ} + (1 - \lambda_{\rm occ})\frac{n}{n+1}, \tag{10}$$

where n is number of the frames in which the maximum response value of the original classifier is less than zero and the classifier $H_{\rm ncp}$ is used to estimate the target state. This indicates that the learning rate $\lambda_{\rm occ}$ will continuously increase if the target object has been occluded for an extended period, and we thereby prevent the features of the true target tracked by the classifier from being replaced by those of the covered object.

To prevent that tracker from losing its target object or the classifier from being updated with incorrect sample information when its maximum response value is less than zero, our algorithm uses two classifiers, $H_{\rm ocp}$ and $H_{\rm ncp}$ to track the target object; the updating of these two classifiers is of great importance to the performance of the algorithm. In the process of tracking from frame($t{-}1$) to frame(t), different learning rates and classifier states are set in accordance with the maximum response values of the original classifier and the flags in Table 1, where $\max(H_{\rm ocp}(v))$ is the maximum response value of the classifier for the test sample in frame(t) and a flag is used to determine the state to which $H_{\rm ncp}$ should be set. The initial value of $H_{\rm ncp}$ is zero.

Our algorithm, to be used in combination with Table 1, is shown here as Algorithm 1. We first determine, in frame(t), whether condition 1 or condition 3 is satisfied. If the latter is the case, then the tracking process is the same as with the CT algorithm. If condition 1 is satisfied, then the maximum response value of the classifier $H_{\rm ocp}$ is less than zero, and the corresponding flag is zero. This indicates that the classifier $H_{\rm ocp}$ obtained in the previous (t−1) frames through adaptive updates cannot successfully locate the real target in frame(t). Thus, steps 3–6 need to be carried out to initialize the classifier $H_{\rm ncp}$, locate the target in frame(t), set the flag to one and collect samples in frame(t) to update both classifiers $H_{\rm ncp}$ and $H_{\rm ocp}$. It should be noted that in condition 1, the learning rate of $H_{\rm ncp}$ is constant, while that of $H_{\rm ocp}$ is $\lambda_{\rm occ}$, which is a dynamic value. We can then continue to handle the next frame. Since the flag is equal to one, only conditions 2 and 4 may occur; their corresponding procedures are respectively from steps 12–14 and steps 16–18, as shown in Algorithm 1. If condition 2 is satisfied, then to reduce the performance degradation of the classifier $H_{\rm ocp}$, we update the learning rate $\lambda_{\rm occ}$ in line with (10). If condition 4 is satisfied, the maximum response value of $H_{\rm ocp}$ is first employed to locate the target. We then update classifier $H_{\rm ncp}$ to zero, retain the learning rate $\lambda_{\rm occ}$ and set the flag to zero. After all this has been done, we jump back to step 2 and begin to deal with the next frame and to assess conditions 1 and 3.

Algorithm 1 Mixed classifier decision tracking

```
 1: Input: $t$-th video frame
 2: if condition 1 is satisfied then
 3:     Initial classifier $H_{\rm ocp}$ with both the positive and the negative samples collected in frame($t$−1), locate the target in frame($t$) as $L_t^* = \max(H_{\rm ncp}(v))$, initialize $\lambda_{\rm occ}$;
 4:     Collect both the positive and the negative samples in frame($t$), update classifier $H_{\rm ncp}$ with $\lambda$ in accordance with (5);
 5:     Update $\lambda_{\rm occ}$ according to (10), then update classifier $H_{\rm ocp}$ with the updated $\lambda_{\rm occ}$ in line with (5);
 6:     Set flag to 1, then go to step 11 to handle the next frame, $t$=$t$+1;
 7: else if condition 3 is satisfied then
 8:     Locate the target in frame($t$) as $L_t^* = \max(H_{\rm ocp}(v))$, update classifier $H_{\rm ocp}$ with $\lambda$ according to (5);
 9:     Jump back to step 2 to handle the next frame, $t$=$t$+1;
10: end if
11: if condition 2 is satisfied then
12:     Locate the target in frame($t$) as $L_t^* = \max(H_{\rm ncp}(v))$, update classifier $H_{\rm ncp}$ with $\lambda$ based on (5);
13:     Update $\lambda_{\rm occ}$ on the basis of (10), then update classifier $H_{\rm ocp}$ with the updated $\lambda_{\rm occ}$ on the basis of (5);
14:     Jump back to step 11, and deal with the next frame, $t$=$t$+1;
15: else if condition 4 is satisfied then
16:     Locate the target in frame($t$) as $L_t^* = \max(H_{\rm ocp}(v))$, update classifier $H_{\rm ocp}$ with $\lambda$ in accordance with (5);
17:     Set the state of classifier $H_{\rm ocp}$ to zero, retain $\lambda_{\rm occ}$, set flag to zero;
18:     Jump back to step 2 to deal with the next frame, $t$=$t$+1;
19: end if
20: Output: Tracking location $L_t^*$ and classifier parameters
```

3.3 Discussion

It is undeniable that the CT algorithm is a simple and efficient real-time tracking method. However, several aspects still need improvement. It may not work well when the tracking targets are influenced by occlusion, variations in illumination and background clutter, etc. In this section, we discuss the merits of the proposed algorithm in comparison with the CT algorithm.

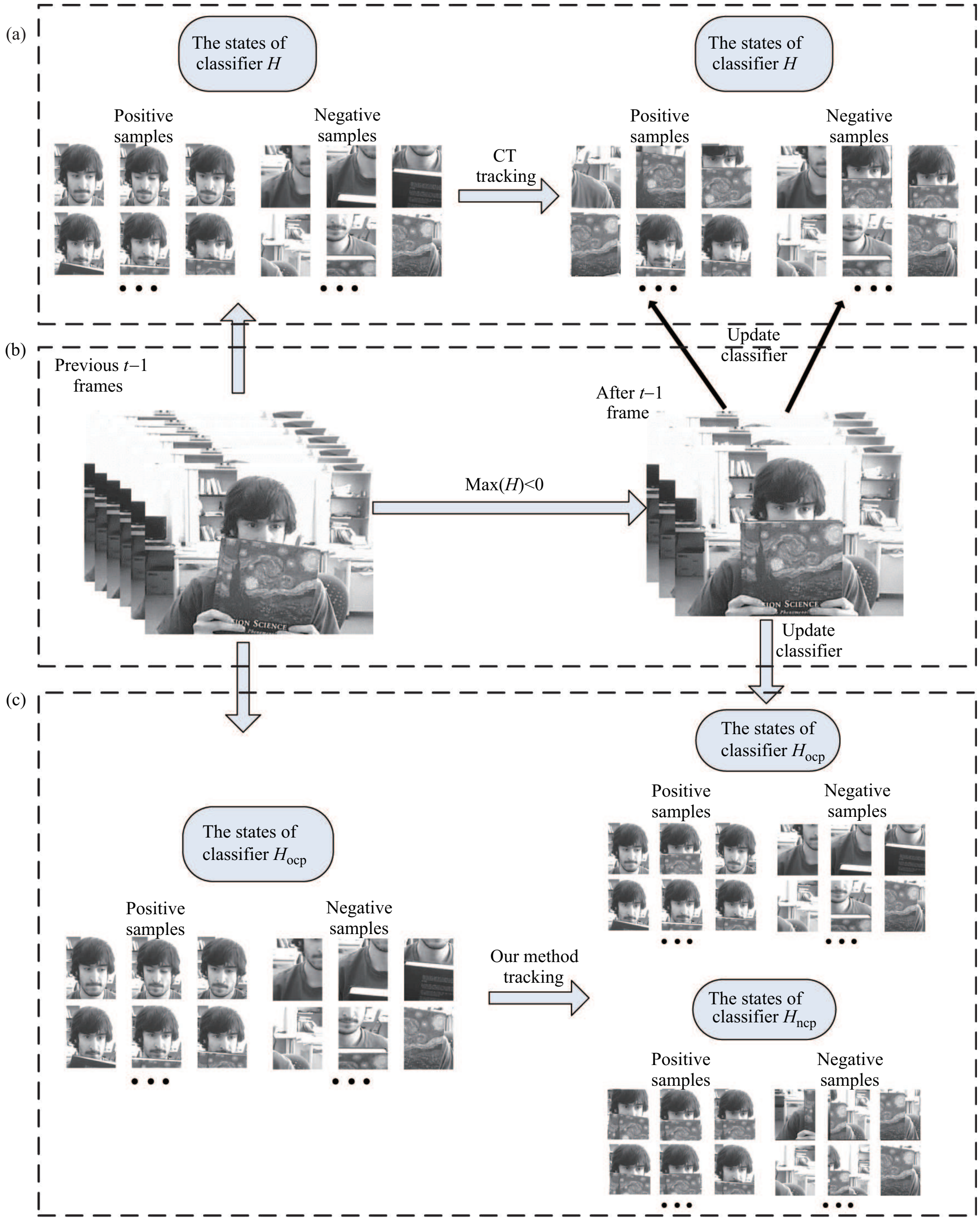

Figure 3 (Color online) Comparison between the CT algorithm and the proposed method. (a) Classifier state of the CT algotithm; (b) video sequence subject to serious occlusion; (c) two classifiers state of the proposed method.

Taking serious occlusion as an example, Figure 3 demonstrates that when dealing with the same video sequence subject to serious occlusion, the states of the classifiers in the CT algorithm and in our proposed method are quite different. As shown in Figure 3(b), if occlusion is serious, the classifiers in both algorithms will label all the test samples as negative samples, i.e., outlier objects, with the maximum response values of the current classifiers being less than zero. Under this condition, however, the CT algorithm will still take the outlier object as the current tracking result. This is because it keeps only a single classifier as its tracker, and therefore there will be a rather large error between the real target location and the position of the target located by the tracker. What makes this situation even worse is that the location of the target offered by the tracker may be totally wrong under some circumstances. Moreover, the classifier of the CT algorithm in this situation still needs to be updated in the subsequent frames. Therefore, much authentic information about both real positive and real negative samples is gradually replaced by the wrong information coming from the inaccurate target location. Thus, the performance of the classifier in the CT algorithm will continue to deteriorate. In Figure 3(a), H on the left represents the state of the classifier trained in the previous frame($t-1$), while H on the right shows the state of the classifier that has experienced a period of training in the frames that have suffered serious occlusion. It is clear that the classifier on the right has already taken much false information generated from the wrong positive and negative samples in the frames after frame(t), and the target under tracking may ultimately drift away and even be lost.

To deal with the shortcomings of the CT algorithm, we propose a method of efficient compressive sensing tracking via mixed classifier Decision. Our algorithm uses the mixed classifier model introduced in Subsection 3.1 and applies the updating strategy shown in Table 1. If the maximum response value of the classifier is less than zero, our method employs the classifier $H_{\rm ocp}$ to preserve the previously learned sample information and, at the same time, establishes a new classifier $H_{\rm ncp}$ based on the target located by $H_{\rm ocp}$ in the previous frame. Once the target object has been occluded for an extended period, the classifier of the CT algorithm will learn all the features of the covered object. When this happens, the original target will be lost. To avoid this shortcoming of the CT algorithm, as shown in the green frame in Figure 3(c), our classifier $H_{\rm ncp}$ takes the seriously occluded target as a positive sample and then functions as a new tracker to continuously track the target that has undergone occlusion.

Since the CT algorithm updates the classifier with a constant learning rate and the updated classifier tends to learn the current samples, the information about the previously learned samples is easily lost. To remedy this defect of the CT algorithm and to reduce the loss of features of the previously learned samples, we use the dynamic incremental learning rate defined in (10) to update the classifier $H_{\rm ocp}$ when the classifier $H_{\rm ncp}$ is used to track the target. As shown by the states of $H_{\rm ocp}$ on the right-hand side of Figure 3(c), only a small number of samples are replaced by samples that have experienced serious occlusion. As a result, the classifiers in our method are able not only to learn new samples in order to adapt to changes in appearance, but also to preserve the previously learned sample information.

4 Results and analysis of experiments

In this section, we compare our mixed classifier decision CT (MDCT) method with five state-of-the-art tracking algorithms on eight standard video sequences. The five tracking algorithms are (CT) [26], the circulant structure tracker (CSK) [33], the online multiple-instance learning tracker (MIL) [17], the online AdaBoot tracker (OAB) [34], and the tracking-learning-detection tracker (TLD) [35]. The eight video sequences cover a number of challenging problems caused by serious occlusion, pose adjustment and variations in illumination, etc. Our experiments are carried out in a Windows 7 environment, using Matlab R2013a, and the computer configuration is an Intel Core i5 with a 2.8 GHz CPU and 8 GB of RAM.

4.1 Experimental setup

We set the search radius that is used to detect the object location as $\gamma = 20$, the search radius for drawing positive samples of the update classifiers as $\partial = 4$, and the inner and outer search radii for drawing negative samples as $\xi = 8$ and $\beta = 30$, respectively. As all trackers are sensitive to the size and scale of the initial frame, for fair comparison, we set the value of the first frame in the ground truth files provided by reference [36] as the standard for all the locations and sizes of the initial object frames in our video sequences. We compare the experimental results from the proposed algorithm with those from the other algorithms on corresponding video sequences.

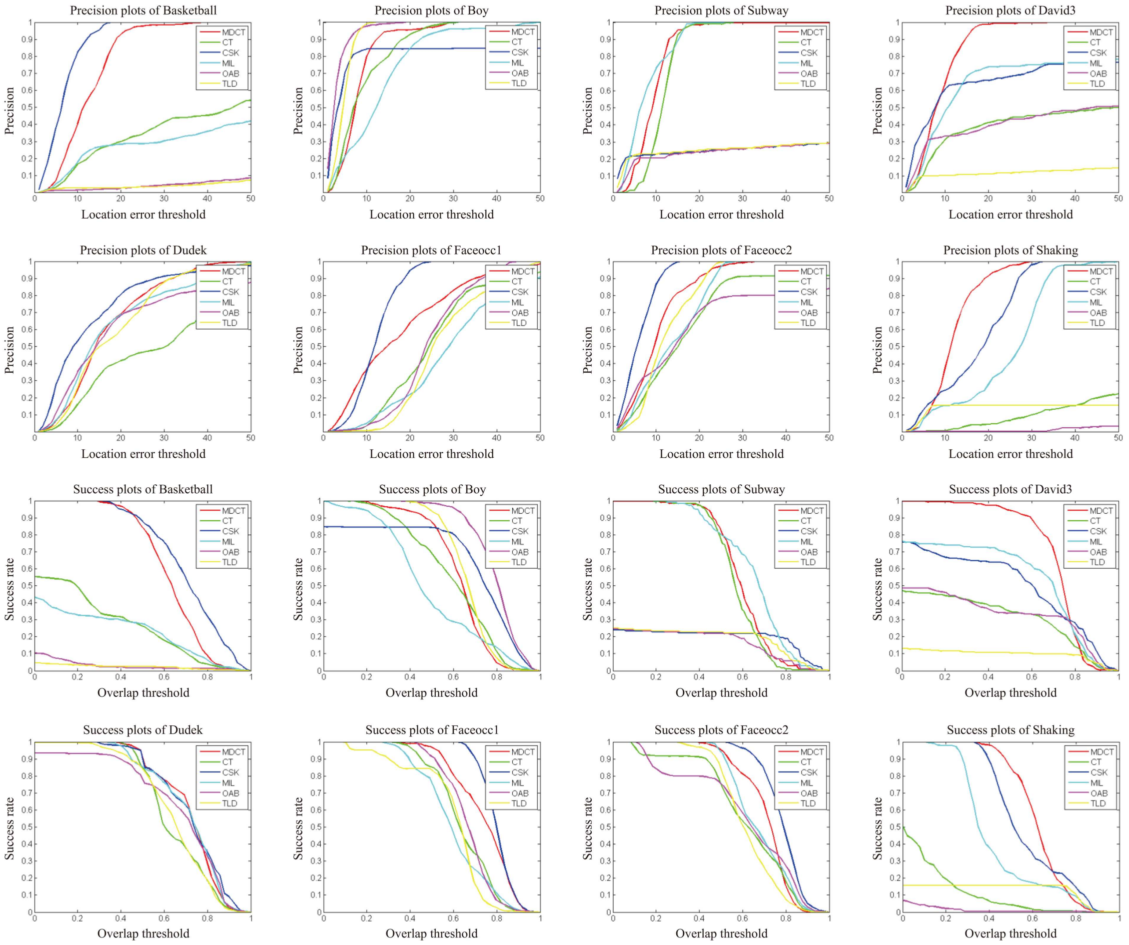

Figure 4 (Color online) Precision plots and success rate for all video sequences.

4.2 Experimental results

The center location error is a metric that is widely used to evaluate the performance of trackers and is defined as the average Euclidean distance between the center location of the tracked targets and the manually labeled ground truth. However, when the tracker loses the target, the output location can be random and therefore, the center location error may fail to measure tracking performance correctly. The bounding box overlap is a measurement for estimating the degree of overlap of the tracked bounding box r_t and the ground truth bounding box r_a. The overlap score is defined as $S = \frac{r_t \bigcap r_a}{r_t \bigcup r_a}$. In accordance with the evaluation criterion given in reference [36], we measure the overall tracking performance of the tracking algorithms using four important norms: average center location error, average overlap score, precision plots, and success plots. The precision plots indicate the percentage of frames whose estimated location is within the given threshold distance of the ground truth, and the success plots are the corresponding percentage of the frames whose overlaps S are larger than the given threshold from the total frames.

The experimental results about the precision plots and success rate of the six algorithms on the eight benchmark video sequences mentioned in this paper are shown in Figure 4. We also display the average center location errors and the average overlap scores of all the trackers on each video sequence. As shown in Tables 2 and 3, the bold fonts indicate the best performance, while the bold print fonts indicate the second-best performance.

According to the results, our proposed MDCT exhibits excellent performance for both the metric overlap and the center location error. As shown in Figure 4, when the reference threshold is set as 20 pixels from the center location error, the precision plots within the given threshold of the video sequences for Basketball, Boy, Subway, David3, Faceocc2 and Shaking are 92.83%, 95.51%, 98.29%, 98.81%, 90.76%

Table 2 Average center location errors (in pixel)

Sequence	MDCT	CT	CSK	MIL	OAB	TLD
Basketball	12	89	**7**	92	205	304
Boy	8	9	20	13	**3**	4
Subway	10	11	164	**8**	113	205
David3	**8**	89	56	30	83	245
Dudek	16	27	**14**	18	31	21
Faceocc1	18	26	**12**	30	25	27
Faceocc2	10	19	**6**	14	20	12
Shaking	**12**	80	17	24	192	227
Average value	**12**	44	37	29	84	131

Table 3 Average bounding box overlap score (%) and FPS

Sequence	MDCT	CT	CSK	MIL	OAB	TLD
Basketball	63.26	25.63	**70.73**	21.96	2.86	2.23
Boy	62.3	59.01	65.39	49.1	**79.1**	66.16
Subway	59.53	57.45	19.28	**64.81**	16.35	18.33
David3	**71.79**	30.64	49.16	53.68	32.55	9.67
Dudek	71.43	64.7	**71.76**	70.73	65.67	64.57
Faceocc1	73.52	63.68	**79.46**	59.58	66	58.47
Faceocc2	70.47	60.77	**77.97**	67.21	59.75	61.65
Shaking	**62.28**	10.12	56.76	42.68	1.33	12.76
Average value	**66.82**	46.5	61.31	53.72	40.45	36.73
Average FPS	36.196	45.248	192.971	19.988	3.98	19.381

and 91.52%, respectively. The average precision plot of MDCT on all these sequences is 87.6%, which is the highest among those of all the trackers whose threshold from the location error is within 20 pixels. Moreover, the success rate of MDCT reaches 88.4% when the overlap threshold S=0.5, surpassing the values of all the other trackers. The average overlap scores for each algorithm on the eight sequences are shown in Table 3, the last column of which shows the average frame per second (FPS) for each algorithm on the eight sequences. The average overlap scores achieved by MDCT are greater than 70% for most of the video sequences, and for all sequences, the scores are much greater than or close to 60%, as shown in Table 3. When dealing with interference factors such as occlusion, variations in illumination and background clutter, the CSK algorithm ranks first on FPS for the eight video sequences, reaching 192.971 FPS. The CT algorithm and our MDCT achieve the second and the third places, respectively, on FPS. Since MDCT requires a heavier computing load for the judging, establishing and updating the states of the classifiers, it will be slightly weaker than the CT algorithm with regard to FPS. Nevertheless, it still shows good real-time performance. Although both CT and the CSK algorithm perform better than our tracker on FPS, we can see from Tables 2 and 3 that our algorithm performs better than most of the other trackers, ranking either first or second for average center location error and average overlap score. Moreover, the average values of these two metrics on all the video sequences rank first. What deserves special attention is that the locations of the target boxes provided by the tracker in the TLD algorithm have gone out side the frames in some sequences. In this case, to ensure accuracy, we adopt the average value of the maximum location errors of the other algorithms as the original location error of the TLD algorithm and set the overlap score to 0. Figures 5 and 6 present screenshots from some video sequences for all the trackers.

4.3 Performance analysis on occlusion

To further illustrate the performance of our proposed method, we analyze its capacity to deal with occlusion. The tracking results for the video sequences Subway, David3, Faceocc1 and Faceocc2 are shown in Figure 5(a), (b), (c) and (d), respectively. The total number of these sequences is 2132. The

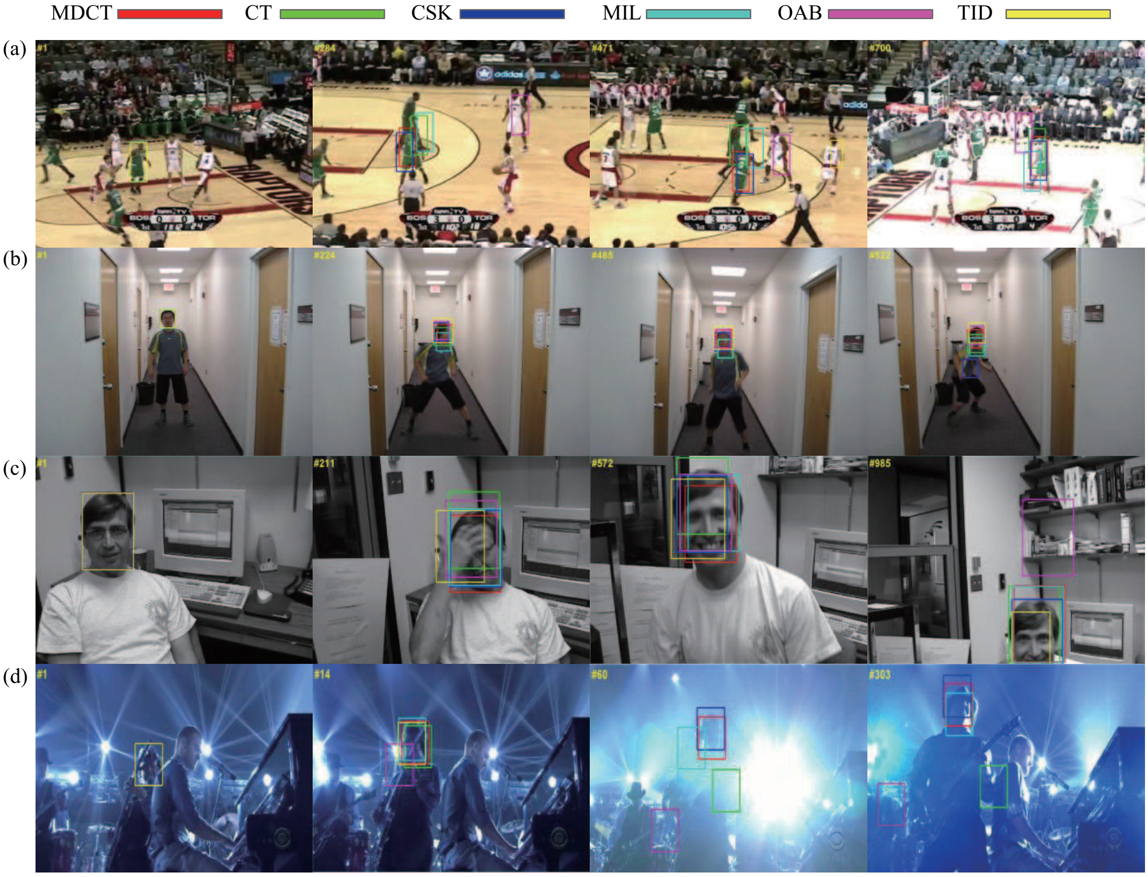

Figure 5 (Color online) Screenshots of some sampled tracking results. (a) Subway; (b) David3; (c) Faceocc1; (d) Faceocc2.

targets in Subway in #42 and #103, David3 in #28, #88 and #185, Faceocc1 in #720, #785 and #890, and Faceocc2 in #495, #707 and #709 all suffer from partial or serious occlusion. We can see that the performance of MDCT is clearly improved compared with that of CT. Taking Faceocc2 as an example, the face of the man is occluded by a book in the sequence from #686 to #736. Among these frames, more than 30 are subject to serious occlusion. As with CT, our method uses a single classifier during tracking. So, when the maximum response value of its classifier is less than zero, the algorithm will continue to take the sample with the maximum response value as the location of the target. Under this condition, the location error with the green target box will occur in frame #709 in Figure 5(d). Moreover, the CT algorithm will fail to take into account whether the object is occluded. Therefore, when the target in the Faceocc2 sequence is subject to serious occlusion for an extended period, the features of the covered object will gradually replace those of the original samples learned by the classifier and wrong locations will still occur even if the real target appears again in subsequent frames. The situation is shown in the green object box in #757.

In our algorithm, we use a mixed classifier decision model and a dynamic learning rate to handle the occlusion that occurs in the Faceocc2 sequence. When the maximum response value of our classifier is less than zero, we first preserve the previously learned sample information with the classifier $H_{\rm ocp}$ and then use a new classifier $H_{\rm ncp}$ to model the man's face that is occluded by a book. Thus, from the discussion above, we can see that when the maximum response value of the classifier is less than zero, the MDCT can still track the occluded object, whereas the CT algorithm uses the sample with the maximum response value, which is less than zero, as the tracking result. From a comparison between the tracking results from MDCT (the red target box in Figure 5(d)) and CT (the green box in the same frame), we can see that our method is more accurate than the CT algorithm. While the classifier $H_{\rm ncp}$ is tracking the target, we update the classifier $H_{\rm ocp}$ with a dynamic incremental learning rate. Thus, much of the

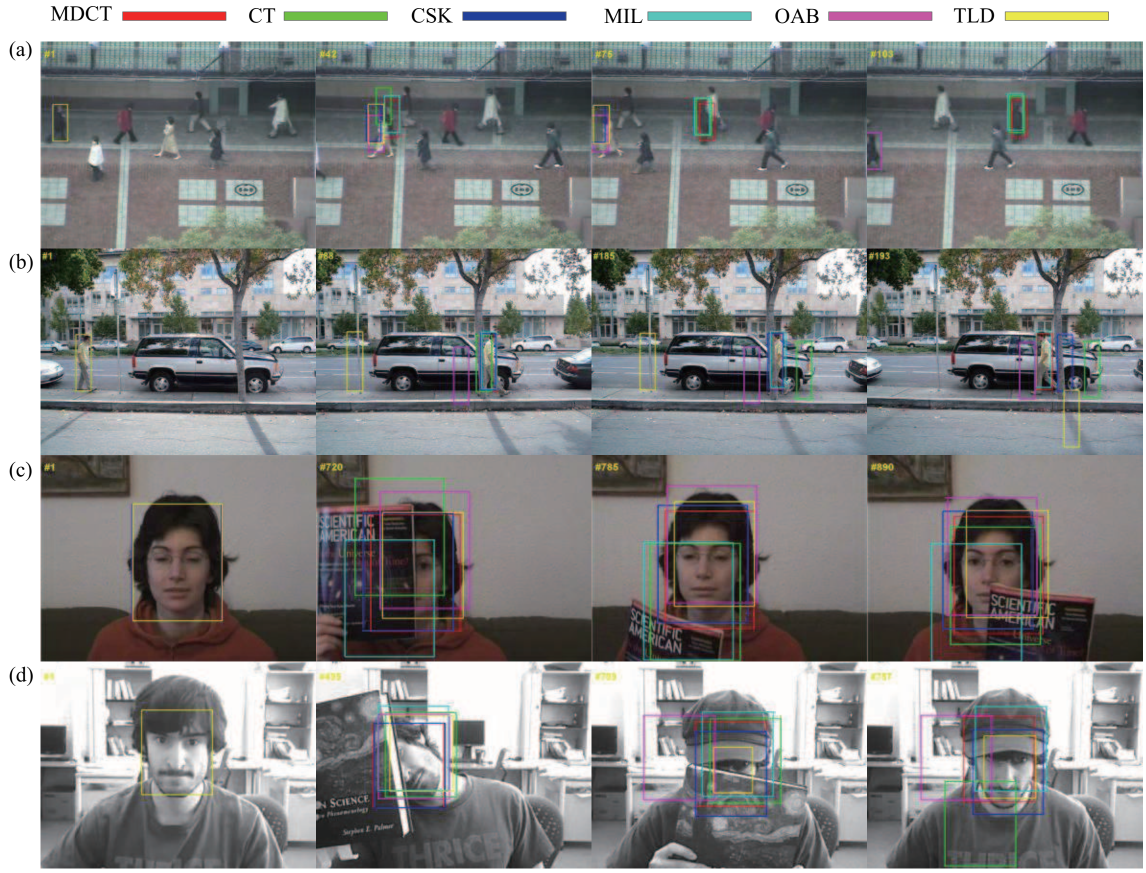

Figure 6 (Color online) Screenshots of some sampled tracking results. (a) Basketball; (b) Boy; (c) Dudek; (d) Shaking.

sample information that was learned before serious occlusion occurred is preserved by $H_{\rm ocp}$, and when the true target reappears, we can again locate it correctly. This situation is illustrated in the red box in Figure 5(d). In this case, both the uses of mixed classifiers and the dynamic learning rate are very important.

In addition, for a short period of serious occlusion, as shown in the Subway sequence from #40 to #43, since the CT algorithm employs a single classifier to track the target, location error occurs, as can be seen in Figure 5(a). In contrast, our method applies a mixed classifier decision model and uses the classifier $H_{\rm ocp}$ to track the occluded target. Its tracking result is therefore more accurate than that of the CT algorithm. As the occlusion continues for only a short period, only a small number of the features of the target previously learned by the classifier of the CT algorithm are replaced by the features of the covered object. Therefore, the CT algorithm may still find the target in the following frames to some degree. From the analysis above, the use of a mixed classifier decision model is more important than the dynamic learning rate in these sequences. To sum up, we can see from the tracking results that when compared with the CT algorithm, which uses a single classifier and updates the classifier parameters with a constant learning rate, the performance of our method in dealing with occlusion is improved to some extent.

4.4 Performance analysis on some interference factors

Since variations in illumination, motion blur and background clutter are very common in practice, we analyze the performance of MDCT on these factors. Interference factors make it difficult to track the targets in the video sequences shown in Figure 6. The targets in Basketball in #700 as well as in Shaking in #60 and #303 are subject to variations in illumination, and the target in the Basketball sequence in #284 and #471 moves into regions with similar texture. We can see from Tables 2 and 3 that the average center location errors and the average overlap score of MDCT in the Basketball sequence are 12 and

63.26%, and in the Shaking sequence they are 12 and 62.28%. This illustrates that when dealing with variations in illumination and background clutter, the proposed tracker is the only one that can match CSK, whose average center location error and average overlap score in the Basketball sequence are 7 and 70.73% and those in the Shaking sequence are 7 and 56.64%. Meanwhile, the other algorithms such as CT and MIL function perform less well. As shown in the Boy sequence in Figure 6(b) and the Dudek sequence in Figure 6(c), the targets undergo motion blur or camera shake. In #445 of the Boy sequence and #221 of the Dudek sequence, the tracking results of the CT algorithm are illustrated in the green target boxes. They show that the real targets drift away owing to motion blur and camera shake, while the tracking results demonstrated in the red target boxes prove that MDCT can still correctly locate the targets despite these challenging factors. Although MDCT fails to achieve the best performance on these two sequences, it is still better than the CT algorithm. The reason is that MDCT not only uses a mixed classifier decision strategy but also updates its classifiers with a dynamic learning rate.

5 Conclusion

We proposed an efficient and robust tracking algorithm using a mixed classifier decision strategy. The establishment of the target model and the updating strategy of the classifier are of great importance for the performance of a tracker. In our algorithm, the two classifiers are designed to establish different target models at different times and to use different learning rates to update the models. Thus, the amount of error in the information used to update the target appearance model can be effectively reduced and the robustness of the compressive sensing tracking algorithm in some complicated scenarios can be improved. From a comparison with five state-of-the-art algorithms on some challenging benchmark video sequences, we can conclude that the proposed algorithm performs much better in handling factors such as serious occlusion, variations in illumination and background clutter. In future work, we will focus on several other challenging problems in a tracking-by-detection framework, such as scale changes and relocation of the target after it has been lost.

Acknowledgements This work was supported in part by National Basic Research Program of China (973 Program) (Grant No. 2012CB719905) and National Natural Science Foundation of China (Grant No. 61471274).

Conflict of interest The authors declare that they have no conflict of interest.

References

1 Cannons K. A Review of Visual Tracking. Technical Report CSE-2008-07. 2008

2 Yilmaz A, Javed O, Shah M. Object tracking: a survey. ACM Comput Surv, 2006, 38: 1–35

3 Comaniciu D, Ramesh V, Meer P. Kernel-based object tracking. Pattern Anal Mach Intell, 2003, 25: 564–577

4 Ross D, Lim J, Lin R-S, et al. Incremental learning for robust visual tracking. Int J Comput Vision, 2008, 77: 125–141

5 Mei X, Ling H. Robust visual tracking using l1 minimization. In: Proceedings of IEEE International Conference on Computer Vision, Nice, 2009. 1436–1443

6 Fan J, Shen X, Wu Y. Scribble tracker: a matting-based approach for robust tracking. Pattern Anal Mach Intell, 2012, 34: 1633–1644

7 Wu Y, Huang T S. Robust visual tracking by integrating multiple cues based on co-inference learning. Int J Comput Vision, 2004, 58: 55–71

8 Kwon J, Lee K M. Visual tracking decomposition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, 2010. 1269–1276

9 Adam A, Rivlin E, Shimshoni I. Robust fragments-based tracking using the integral histogram. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, New York, 2006. 798–805

10 Mei X, Ling H. Robust visual tracking and vehicle classification via sparse rep-resentation. Pattern Anal Mach Intell, 2011, 33: 2259–2272

11 Li H, Shen C, Shi Q. Real-time visual tracking using compressive sensing. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Providence, 2011. 1305–1312

12 Liu B Y, Huang J Z, Yang L, et al. Robust tracking using local sparse appearance model and k-selection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Providence, 2011. 1313–1320

13 Jia X, Lu H, Yang M-H. Visual tracking via adaptive structural local sparse appearance model. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Providence, 2012. 1822–1829

14 Zhang T, Ghanem B, Liu S, et al. Robust visual tracking via structured multi-task sparse learning. Int J Comput Vision, 2013, 101: 367–383

15 Collins R, Liu Y, Leordeanu M. Online selection of discriminativetracking features. Pattern Anal Mach Intell, 2005, 27: 1631–1643

16 Babenko B, Yang M-H, Belongie S. Visual tracking with online multiple instance learning. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Miami, 2009. 983–990

17 Babenko B, Yang M-H, Belongie S. Robust object tracking with online multiple instance learning. Pattern Anal Mach Intell, 2011, 33: 1619–1632

18 Kalal Z, Matas J, Mikolajczyk K. P-N learning: bootstrapping binary classifier by structural constraints. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, 2010. 49–56

19 Zhang Y, Du B, Zhang L. A sparse Representation-Based binary hypothesis model for target detection in hyperspectral images. IEEE Trans Geosci Remote Sens, 2015, 53: 1346–1354

20 Tao D, Cheng J, Song M, et al. Manifold ranking-based matrix factorization for saliency detection. IEEE Trans Neural Netw Lear Syst, in press. doi: 10.1109/TNNLS.2015.2461554

21 Tao D, Lin X, Jin L, et al. Principal component 2-dimensional long short-term memory for font recognition on single Chinese characters. IEEE Trans Cybernetics, in press. doi: 10.1109/TCYB.2015.2414920

22 Tao D C, Tang X O, Li X L, et al. Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. IEEE Trans Pattern Anal Mach Intell, 2006, 28: 1088–1099

23 Tao D C, Li X L, Wu X D, et al. General tensor discriminant analysis and gabor features for gait recognition. IEEE Trans Pattern Anal Mach Intell, 2007, 29: 1700–1715

24 Xu C, Tao D C, Xu C. Multi-view intact space learning. IEEE Trans Pattern Anal Mach Intell, 2015, 37: 2531–2544

25 Liu T L, Tao D C. Classification with noisy labels by importance reweighting. IEEE Trans Pattern Anal Mach Intell, in press. doi: 10.1109/TPAMI.2015.2456899

26 Zhang K, Zhang L, Yang M-H. Real-time compressive tracking. In: Proceedings of European Conference on Computer Vision, Florence, 2012. 864–877

27 Avidan S. Support vector tracking. IEEE Trans Pattern Anal Mach Intell, 2004, 26: 1064–1072

28 Collins R, Liu Y, Leordeanu M. Online selection of discriminative tracking features. IEEE Trans Pattern Anal Mach Intell, 2005, 27: 1631–1643

29 Avidan S. Ensemble tracking. IEEE Trans Pattern Anal Mach Intell, 2007, 29: 261–271

30 Grabner H, Leistner C, Bischof H. Semi-supervised on-line boosting for robust tracking. In: Proceedings of European Conference on Computer Vision, Prague, 2008. 234–247

31 Zhou Q, Lu H, Yang M H. Online multiple support instance tracking. In: Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition, Ljubljana, 2011. 545–552

32 Hare S, Saffari A, Torr P. Struck: structured output tracking with kernels. In: Proceedings of IEEE International Conference on Computer Vision, Barcelona, 2011. 263–270

33 Henriques F, Caseiro R, Martins P, et al. Exploiting the circulant structure of tracking-by-detection with kernels. In: Proceedings of European Conference on Computer Vision, Florence, 2012. 702–715

34 Grabner H, Grabner M, Bischof H. Real-time tracking via online boosting. In: Proceedings of British Machine Vision Conference, Edinburgh, 2006. 47–56

35 Kalal Z, Mikolajczyk K, Matas J. Tracking-learning-detection. IEEE Trans Pattern Anal Mach Intell, 2011, 34: 1409–1422

36 Wu Y, Lim J, Yang M H. Online object tracking: a benchmark. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Portland, 2013. 2411–2418