Pattern Recognition 46 (2013) 397–411

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

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

Real-time visual tracking via online weighted multiple instance learning

Kaihua Zhang [a,*], Huihui Song [b]

[a] Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China
[b] Department of Geography and Resource Management, Chinese University of Hong Kong, Hong Kong, China

ARTICLE INFO

Article history:
Received 12 December 2011
Received in revised form
30 June 2012
Accepted 18 July 2012
Available online 1 August 2012

Keywords:
Visual tracking
Multiple instance learning
Tracking by detection
Sliding window

ABSTRACT

Adaptive tracking-by-detection methods have been widely studied with promising results. These methods first train a classifier in an online manner. Then, a sliding window is used to extract some samples from the local regions surrounding the former object location at the new frame. The classifier is then applied to these samples where the location of sample with maximum classifier score is the new object location. However, such classifier may be inaccurate when the training samples are imprecise which causes drift. Multiple instance learning (MIL) method is recently introduced into the tracking task, which can alleviate drift to some extent. However, the MIL tracker may detect the positive sample that is less important because it does not discriminatively consider the sample importance in its learning procedure. In this paper, we present a novel online weighted MIL (WMIL) tracker. The WMIL tracker integrates the sample importance into an efficient online learning procedure by assuming the most important sample (i.e., the tracking result in current frame) is known when training the classifier. A new bag probability function combining the weighted instance probability is proposed via which the sample importance is considered. Then, an efficient online approach is proposed to approximately maximize the bag likelihood function, leading to a more robust and much faster tracker. Experimental results on various benchmark video sequences demonstrate the superior performance of our algorithm to state-of-the-art tracking algorithms.

1. Introduction

Object tracking has been widely studied in computer vision because it is very important in many applications, e.g., automated surveillance, video indexing, traffic monitoring, and human–computer interaction, etc. [1]. Numerous algorithms have been proposed during the past decades [2–10,12,14,15,17,18,31–33]. However, it is still a very challenging task to design a robust tracking system because many factors such as illumination changes, appearance changes, shape variations, and partial or full occlusions [1], are very challenging to cope with.

The appearance model is a prerequisite for the success of a tracking system. How to design a robust appearance model which can be adaptive to the factors mentioned above is a key task in most recently proposed algorithms [3–6,10,12,14,15,17]. In general, the recently proposed tracking models can be categorized into two classes based on their different appearance representation schemes: generative models [2–6] and discriminative ones [7,8,10,12,14,15,17,18]. Generative models first learn an appearance model to represent the object, and then search for the object appearance at each frame most similar to the learned appearance. Black et al. [2] learned a subspace appearance model offline. However, the offline learned appearance model is difficult to adapt the appearance variations. To deal with appearance variations, some online models have been proposed such as the WLS tracker [3] and IVT method [4]. Adam et al. [5] utilized multiple fragments to design an appearance model which is robust to partial occlusions. Recently, sparse representation has been introduced to the tracking task [6], demonstrating good performance to partial occlusions, illumination changes and pose variations. However, these generative models do not take into account background information, throwing away some very useful information that can help to discriminate object from background [19].

Discriminative models take tracking as a binary classification task to separate object from its surrounding background. These methods are also called tracking-by-detection methods which take tracking as a detection task [7]. The adaptive tracking-by-detection methods first train a classifier in an online manner using samples extracted from the current frame. When the next frame is coming, a sliding window method is then used to extract some samples around the old object location at this frame, and then the afore-trained classifier is applied to these samples. The location of the sample with the maximum classifier score is the new object location at this frame. In [8], the boosting method is used to train a strong classifier using the labeled pixels inside a

* Corresponding author. Tel.: +852 64326108.
E-mail addresses: zhkhua@gmail.com (K. Zhang), hhsongsherry@gmail.com (H. Song).

0031-3203/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2012.07.013

rectangle around the object and surrounding the rectangle, and then mean-shift method [9] is used to find the location with the maximum classifier score at the new frame as the new object location. The features used in [8] may contain redundant and irrelevant information which affects the classification performance. To improve the classification performance, feature selection is needed. Collins et al. [10] have demonstrated that selecting discriminative features in an online manner can greatly improve the tracking performance. Inspired by the advances in face detection [11], many boosting feature selection methods have been proposed. Grabner et al. [12] proposed an online boosting feature selection method motivated by the online ensemble method [13] (*cf.* Fig. 1(a)). However, all the above-mentioned discriminative methods [7,8,10,12] only utilize one positive sample (i.e., the tracking result in current frame) and multiple negative samples to update the classifier. If the object location detected by the current classifier is not precise, the extracted positive sample will be imprecise, leading to a suboptimal updated classifier. Then the inaccuracy will be accumulated to degrade the classifier seriously. Finally, this can lead to tracking failure (drift) [15]. In order to alleviate the drifting problem, an online semi-supervised approach [14] was proposed to train the classifier by only labeling the samples at the first frame while leaving the samples at the coming frames unlabeled (*cf.* Fig. 1(c)). However, this semi-supervised approach discards some useful information which is very helpful in the problem domain [15] (e.g., the inter-frame object motion can be safely assumed to be smooth [1]). Multiple positive samples and negative samples are also applied to update the classifier in an online manner [15,17,18]: the positive samples are cropped around the object location, while the negative ones are far from the object location. However, the *ambiguity* problem may occur which confuses the classifier [15]. Inspired by using the multiple instance learning method to solve the ambiguity in face detection [16], an online multiple instance learning (MIL) method [15] (*cf.* Fig. 1(d)) was proposed to handle the ambiguity problem generated by using multiple positive samples and negative ones to update the classifier. The MIL tracker puts the positive samples and negative ones into some positive and negative bags, respectively, and then trains a classifier in an online manner using the bag likelihood function. The MIL tracker has demonstrated good performance to handle drift. However, the Noisy-OR model used in the MIL tracker [15] does not take into account any information about the importance of the positive samples. Therefore, MIL tracker may select less effective positive sample [18].

In this paper, we propose a novel weighted MIL (WMIL) tracker that integrates the sample importance into the learning procedure (*cf.* Fig. 1(e)). We propose a new bag probability function that combines the weighted instance probability: the weight for the instance near the object location is larger than that far from the object location which means the instance near the object location contributes larger to the bag probability. Then, we propose an efficient method to approximately optimize the bag likelihood function. Empirical results on challenging video sequences demonstrate the superior performance of our method in robustness, stability and efficiency to state-of-the-art methods in the literature.

The paper is organized as follows: In Section 2, we introduce our tracking system, the principle of our method and its advantages over state-of-the-art methods in detail. Section 3 gives the detailed experiment setup and results. Finally, Section 4 concludes the paper.

2. Tracking with online weighted multiple instance learning

2.1. Preliminaries

The posterior probability of labeling sample x to be positive is computed using the Bayesian theorem

$$p(y=1|x)=\frac{p(x|y=1)p(y=1)}{\sum_{y\in\{0,1\}}p(x|y)p(y)}=\sigma\left(\ln\left(\frac{p(x|y=1)p(y=1)}{p(x|y=0)p(y=0)}\right)\right) \quad (1)$$

where $\sigma(z)=1/(1+e^{-z})$ is a sigmoid function and $y\in\{0,1\}$ is a binary label of sample x.

Our discriminative appearance model is a classifier $H_K(x)$ defined as

$$H_K(x)=\ln\left(\frac{p(x|y=1)p(y=1)}{p(x|y=0)p(y=0)}\right) \quad (2)$$

Thus, the posterior probability (1) can be represented as

$$p(y=1|x)=\sigma(H_K(x)) \quad (3)$$

where sample x is represented by a feature vector function $\mathbf{f}(x)=(f_1(x),\ldots,f_K(x))^T$. We assume that the features in $\mathbf{f}(x)$ are independently

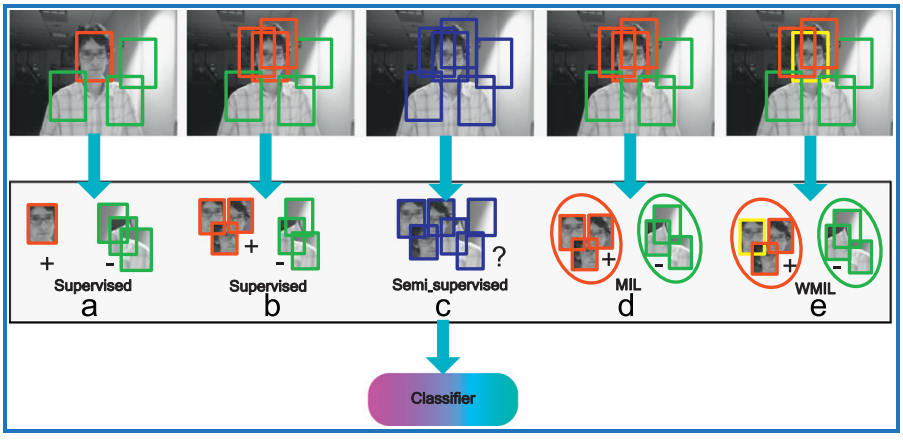

Fig. 1. The paradigms for some representative discriminative appearance models: (a) the online AdaBoost boosting (OAB) method uses one positive sample when training the classifier [12]. (b) Multiple positive samples are treated *equally* when training the classifier. The OAB(45) method in [15]. (c) The semi-supervised method only labels the first frame when training the classifier [14]. (d) Multiple instance learning (MIL) tracker [15]. (e) Our WMIL tracker discriminatively weights the positive samples according to their importance to the bag probability (the sample in the yellow rectangle is the most important sample). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

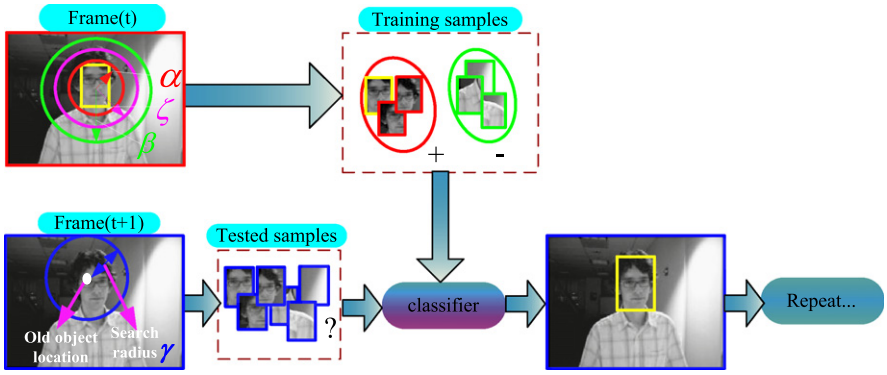

Fig. 2. The basic flow of our system.

distributed and assume uniform prior $p(y=0)=p(y=1)$ as MIL tracker [15]. Then, the classifier $H_K(\cdot)$ in (2) is described with the feature $\mathbf{f}(\cdot)$ as

$$H_K(x)=\ln\left(\frac{p(\mathbf{f}(x)|y=1)p(y=1)}{p(\mathbf{f}(x)|y=0)p(y=0)}\right)=\sum_{k=1}^{K}h_k(x) \tag{4}$$

where

$$h_k(x)=\ln\left(\frac{p(f_k(x)|y=1)}{p(f_k(x)|y=0)}\right) \tag{5}$$

Eq. (5) is a discriminative function which is a weak classifier in MIL tracker [15] and our tracker.

2.2. System overview

The basic flow of our tracking system is illustrated in Fig. 2. The main procedures of our system are as follows: Let $l_t(x)\in R^2$ denote the location of sample x at the tth frame. We assume that the location of sample x_0 is the object location without loss of generality. First, we densely crop some positive samples $X^{\alpha}=\{x|\,||l_t(x)-l_t(x_0)||<\alpha\}$ within a search radius α centering at object location $l_t(x_0)$. Second, we randomly crop some negative samples from set $X^{\zeta,\beta}=\{x|\zeta<||l_t(x)-l_t(x_0)||<\beta\}$, where $\alpha<\zeta<\beta$ because this set is potentially very large. Third, we utilize these positive and negative samples to update the classifier $H_K(\cdot)$ (i.e., (4)). When the $(t+1)$th frame is coming, we crop some samples $X^{\gamma}=\{x|\,||l_{t+1}(x)-l_t(x_0)||<\gamma\}$ with a large radius γ surrounding the old object location at the $(t+1)$th frame. Next, we apply the afore-trained classifier to these samples to find the sample with the maximum confidence as $x^*=\text{argmax}_x\ p(y=1|x)$. Finally, the location $l_{t+1}(x^*)$ is the new object location at the $(t+1)$th frame. Based on the newly detected object location $l_{t+1}(x^*)$, our tracking system is to repeat the above-mentioned procedures.

From the above introduction, we can see that the core component of our system is how to design the classifier $H_K(\cdot)$ (i.e., (4)). Many related works [12,14,15] using online feature selection techniques have been proposed. We will briefly review these works in the following section.

2.3. Related works on feature selection

We first briefly review some related works about online boosting feature selection [12,14], and then introduce the online MIL boosting method [15] which is closely related to our work in detail.

2.3.1. Online boosting feature selection

The AdaBoost method proposed by Freund et al. [23] learns its model in batch mode: some weak classifiers are trained sequentially using a greedy scheme. After each weak classifier is trained, the training samples are re-weighted to weight more on samples misclassified by this classifier, and finally, the trained weak classifiers are linearly weighted into a strong classifier. In computer vision, the seminal work of face detection [11] first introduces AdaBoost method [23] for feature selection: each feature is used to train a weak classifier which is greedily selected by AdaBoost method. Meanwhile, its corresponding feature is selected. Oza [13] proposed an online AdaBoost algorithm which can converge to the offline version if it runs for infinite time. However, the algorithm proposed by Oza [13] cannot be used for feature selection if the weak classifiers are decision stump functions [15]. Grabner et al. [12] extended the algorithm in [13] by maintaining a pool of candidate weak classifiers, and then at each iteration, all of the classifiers in the pool are updated where the weak classifier with the minimum error is selected. This feature selection method is similar to that proposed by Wu et al. [26] which can significantly improve the efficiency of face detection. Friedman et al. [24] have shown that AdaBoost method is equivalent to forward stagewise additive modeling using an exponential loss function. Motivated by this viewpoint, many other loss functions have been used to improve the robustness of boosting method such as the logistic regression function in [24]. Leistner et al. [22] proposed an online gradient boosting [25] feature selection method and compared the performance of different loss functions applied to tracking and other tasks. However, the online boosting feature selection techniques used in [12,22] have to update all of the weak classifiers in the pool after selecting a feature which is still a time-consuming procedure for tracking task.

2.3.2. Online MIL boosting feature selection

Recently, Babenko et al. [15] proposed an online MIL boosting feature selection technique intrigued by the seminal face detection work in [16]. The instance probability is modeled by $p_{ij}=\sigma(H_K(x_{ij}))$ (i.e., (3)), where i indexes the bag, j indexes the instance and $H_K(\cdot)=\sum_{k=1}^{K}h_k(\cdot)$, where $h_k(\cdot)$ (i.e., (5)) is a weak classifier related to each Haar-like feature $f_k(\cdot)$. The conditional distributions in $h_k(\cdot)$ are modeled as a Gaussian function, i.e., $p(f_k(x_{ij})|y_i=1)\sim N(\mu_1,\sigma_1)$ and $p(f_k(x_{ij})|y_i=0)\sim N(\mu_0,\sigma_0)$. Then, the update schemes for the parameters μ_1 and σ_1 are

$$\mu_1\leftarrow\eta\mu_1+(1-\eta)\frac{1}{N}\sum_{j|y_i=1}f_k(x_{ij}) \tag{6}$$

$$\sigma_1\leftarrow\eta\sigma_1+(1-\eta)\sqrt{\frac{1}{N}\sum_{j|y_i=1}(f_k(x_{ij})-\mu_1)^2} \tag{7}$$

where N is the number of positive samples and η is a learning rate parameter. The update schemes for μ_0 and σ_0 have similar formations. Then, the ith bag probability based on the Noisy-OR model is

$$p_i = 1 - \prod_j (1 - p_{ij}) \tag{8}$$

We can see that the Noisy-OR model (8) does not discriminatively treat the positive samples according to their importance to the bag probability, and even if the sample is far from the location of the tracked object at the current frame, it still may have a large probability to contribute largely to the bag probability which makes the MIL tracker easily select less effective features [18].

Algorithm 1. Online MIL Boost.

Input: Dataset $\{X_i, y_i\}_{i=0}^{1}$ where $X_i = \{x_{i1}, x_{i2}, \ldots\}$ is the ith bag and $y_i \in \{0,1\}$

1. Update all M weak classifiers in the pool with data $\{x_{ij}, y_i\}$
2. Initialize $H_{ij} = 0$ for all i,j
3. **for** $k = 1$ to K **do**
4. Set $\ell_m = 0,\ m = 1,\ldots,M$
5. **for** $m = 1$ to M **do**
6. **for** $i = 0$ to 1 **do**
7. **for** $j = 0$ to N+L−1 **do**
8. $p_{ij}^m = \sigma(H_{ij} + h_m(x_{ij}))$
9. **end for**
10. $p_i^m = 1 - \prod_j (1 - p_{ij}^m)$
11. $\ell_m \leftarrow \ell_m + y_i \log(p_i^m) + (1 - y_i) \log(1 - p_i^m)$
12. **end for**
13. **end for**
14. $m^* = \text{argmax}_m(\ell_m)$
15. $h_k(x_{ij}) \leftarrow h_{m^*}(x_{ij})$
16. $H_{ij} = H_{ij} + h_k(x_{ij})$
17. **end for**

Output: Classifier $H_K(x_{ij}) = \sum_k h_k(x_{ij})$, and $p(y = 1 \mid \cdot) = \sigma(H_K(\cdot))$

The MIL tracker first maintains a weak classifier pool $\Phi = \{h_1, \ldots, h_M\}$, and then greedily selects K weak classifiers from the pool Φ by the following criterion:

$$h_k = \underset{h \in \Phi}{\text{argmax}}\ \ell(H_{k-1} + h) \tag{9}$$

where $\ell = \sum_i (y_i \log p_i + (1 - y_i) \log(1 - p_i))$ is the bag log-likelihood function and $H_{k-1} = \sum_{m=1}^{k-1} h_m$ is a strong classifier with $k-1$ selected weak classifiers. This is a sequential forward selection method [20] which is suboptimal but much more efficient than brute force selection. Finally, the selected K weak classifiers construct a strong classifier $H_K = \sum_{k=1}^{K} h_k$. The strong classifier is then applied to the samples cropped from the next frame where the location of sample with the maximum score is the new object location.

The pseudo-code for online MIL boost feature selection is shown by Algorithm 1. As shown by Algorithm 1, all of the instances and bag probabilities (i.e., step 8 and step 10) must be updated M times after selecting a feature which is still a heavily computational burden. In our WMIL tracker, we propose an efficient online approach to approximately optimize the likelihood function which avoids computing the instance probability and bag probability M times after selecting each weak classifier.

2.4. Principle of online weighted MIL

We assume that there exist N positive samples $\{x_{1j}, j = 0, \ldots, N-1\}$ and L negative samples $\{x_{0j}, j = N, \ldots, N+L-1\}$. Without loss of generality, we assume that the location of sample x_{10} is the tracking result at current frame. Then, we put the positive samples and negative ones into two bags $\{X^+, X^-\}$. Similar to the MIL tracker in [15], we also assume that the instance label is the same as the bag label. Then, we define the positive bag probability as follows:

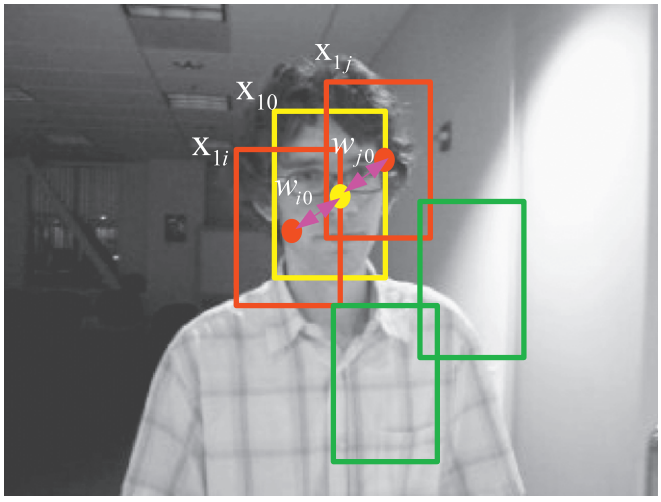

Fig. 3. Illustration of the principle of the weighted MIL tracker. Location of the yellow rectangle is the tracking result. The solid circles are the central locations of each sample. Red and yellow rectangles are the positive samples, while the green rectangles are the negative ones. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$p(y = 1 \mid X^+) = \sum_{j=0}^{N-1} w_{j0}\, p(y_1 = 1 \mid x_{1j}) \tag{10}$$

where $p(y_1 = 1 \mid x_{1j})$ is the posterior probability (i.e., (1)) for sample x_{1j}, and weight w_{j0} (see Fig. 3) is a monotone decreasing function with respect to the Euclidean distance between the locations of sample x_{1j} and sample x_{10}

$$w_{j0} = \frac{1}{c} e^{-|\mathbf{l}(x_{1j}) - \mathbf{l}(x_{10})|} \tag{11}$$

where $l(\cdot) \in R^2$ is the location function and c is a normalization constant. Eq. (10) weighs the positive instances according to their importance to the bag probability, i.e., the instances near the tracking location at the current frame contribute more to the bag probability than those far from the tracking location. Therefore, different from the Noisy-OR model (i.e., (8)) used by MIL tracker, our WMIL tracker naturally integrates the sample importance into the learning procedure by using weighted sum of instance probability (10). Xu et al. [28] proposed a similar bag probability as (10) but assumes that all of the instances contribute equally and independently to the bag's label. In Section 3, we demonstrate that using equally weighted sum of instance probability cannot yield superior performance as our WMIL tracker because the equally weighted scheme assumes equal contribution of positive samples to the positive bag probability which confuses the classifier.

Because all of the instances in the negative bag are very far from the tracking result at the current frame, they are dissimilar to the tracking result at all (see Fig. 3). Therefore, we can assume that all of the negative instances contribute equally to the negative bag

$$p(y = 0 \mid X^-) = \sum_{j=N}^{N+L-1} w p(y_0 = 0 \mid x_{0j}) = w \sum_{j=N}^{N+L-1} (1 - p(y_0 = 1 \mid x_{0j})) \tag{12}$$

where w is a constant. Finally, the bag log-likelihood function is

$$\ell = \sum_{s=0}^{1}(y_s \log(p(y=1|X^+)) + (1-y_s)\log(p(y=0|X^-)))$$
$$= \sum_{s=0}^{1}\left(y_s \log\left(\sum_{j=0}^{N-1} e^{-|\mathbf{l}(x_{1j})-\mathbf{l}(x_{10})|} p(y=1|x_{1j})\right) + (1-y_s)\log\left(\sum_{j=N}^{N+L-1}(1-p(y=1|x_{0j}))\right) + \log(c^{-y_s}w^{1-y_s})\right) \quad (13)$$

We eliminate the trivial constant terms in (13)

$$\ell(H) = \sum_{s=0}^{1}\left(y_s \log\left(\sum_{j=0}^{N-1}\tilde{w}_{j0}p(y=1|x_{1j})\right) + (1-y_s)\log\left(\sum_{j=N}^{N+L-1}(1-p(y=1|x_{0j}))\right)\right) \quad (14)$$

where $\tilde{w}_{j0} = e^{-|\mathbf{l}(x_{1j})-\mathbf{l}(x_{10})|}$, $p(y=1|x_{1j})$ is defined as (3). The weak classifier $h_k(\cdot)$ in $p(y=1|x_{1j})$ is defined by (5). The conditional distributions in $h_k(\cdot)$ are also modeled as a Gaussian function as MIL tracker [15], i.e., $p(f_k(\mathbf{x}_{ij})|y_i=1) \sim N(\mu_1,\sigma_1)$and $p(f_k(\mathbf{x}_{ij})|y_i=0) \sim N(\mu_0,\sigma_0)$. However, different from MIL tracker [15], the parameters μ_1 and σ_1 are updated by the following schemes:

$$\mu_1 \leftarrow \eta\mu_1 + (1-\eta)\overline{\mu} \quad (15)$$

$$\sigma_1 \leftarrow \sqrt{\eta(\sigma_1)^2 + (1-\eta)\frac{1}{N}\sum_{j=0|y_i=1}^{N-1}(f_k(\mathbf{x}_{ij})-\overline{\mu})^2 + \eta(1-\eta)(\mu_1-\overline{\mu})^2} \quad (16)$$

where η is a learning rate parameter and $\overline{\mu} = (1/N)\sum_{j=0|y_i=1}^{N-1} f_k(x_{ij})$ is the average of the kth features extracted from the positive samples at the current frame. We can update μ_0 and σ_0 by the similar rules. (15) and (16) can be easily deduced by Lemma 1 (refer to Appendix A) where we set η as a fixed learning rate parameter to moderate the balance between the former frames and the current frame. We have also tried the parameter update method used by MIL tracker but found most of the experimental results unstable.

Similar to MIL tracker, our tracker also first maintains a weak classifier pool $\Phi=\{h_1,...,h_M\}$, and then we can greedily select the most discriminative weak classifier by maximizing the log-likelihood function

$$h_k = \underset{h\in\Phi}{\operatorname{argmax}}\ \ell(H_{k-1}+h) \quad (17)$$

where $h\in\Phi$ is a weak classifier in the classifier pool and the strong classifier $H_{k-1} = \sum_{m=1}^{k-1} h_m$ (i.e., (4)) where h_m (i.e., (5)) is the mth selected weak classifier.

However, we use a more efficient criterion than the criterion (17) to select the weak classifiers from the pool Φ. Similar to AnyBoost method [29], we first approximate $\ell(H_{k-1}+h)$ by using the first-order Taylor expansion as$\ell(H_{k-1}+h) \approx \ell(H_{k-1}) + \langle h, \nabla\ell(H)\rangle|_{H=H_{k-1}}$, where the inner product is defined by $\langle h, \nabla\ell(H)\rangle \triangleq 1/(N+L)\sum_{j=0}^{N+L-1} h(x_{ij})\nabla\ell(H)(x_{ij})$, where

$$\nabla\ell(H)(x_{ij}) = \left.\frac{\partial\ell(H+\delta 1_{x_{ij}})}{\partial\delta}\right|_{\delta=0}$$
$$= \frac{\partial}{\partial\delta}\sum_{s=0}^{1}\left(y_s\log\left(\sum_{m=0}^{N-1}\tilde{w}_{m0}\sigma(H(\mathbf{x}_{1m})+\delta 1_{x_{ij}})\right) + (1-y_s)\log\left(\sum_{m=N}^{N+L-1}(1-\sigma(H(x_{0m})+\delta 1_{x_{ij}}))\right)\right)\Big|_{\delta=0}$$
$$= \frac{\partial}{\partial\delta}\left(y_i\log\left(\sum_{m=0}^{N-1}\tilde{w}_{m0}\sigma(H(x_{1m})+\delta 1_{x_{ij}})\right) + (1-y_i)\log\left(\sum_{m=N}^{N+L-1}(1-\sigma(H(x_{0m})+\delta 1_{x_{ij}}))\right)\right)\Big|_{\delta=0}$$
$$= y_i\frac{\tilde{w}_{j0}\sigma(H(x_{ij}))(1-\sigma(H(x_{ij})))}{\sum_{m=0}^{N-1}\tilde{w}_{m0}\sigma(H(x_{im}))} - (1-y_i)\frac{\sigma(H(x_{ij}))(1-\sigma(H(x_{ij})))}{\sum_{m=N}^{N+L-1}(1-\sigma(H(x_{im})))} \quad (18)$$

where $1_{x_{ij}} \triangleq y_i \in \{0,1\}$ is the indicator function (i.e., the instance label here) of sample x_{ij}. Then, we select weak classifier h_k by using the following criterion:

$$h_k = \underset{h\in\Phi}{\operatorname{argmax}}\langle h, \nabla\ell(H)\rangle|_{H=H_{k-1}} \quad (19)$$

The weak classifier selection criterion in (19) is much more efficient than that directly maximizes the log-likelihood function used by MIL tracker (i.e., (9)) because we do not need to compute the instance probability and bag probability M times after selecting one weak classifier.

The pseudo-code for online weighted MIL boost feature selection is shown in Algorithm 2.

Algorithm 2. Online weighted MIL Boost.

Input: Dataset $\{X^+,X^-\}$ where $X^+=\{x_{1j},y_1=1,j=0,...,N-1\}$and $X^-=\{x_{0j},y_0=0,j=N,...,N+L-1\}$

```
Update all M weak classifiers in the pool with data {X+,X-}
Initialize H_0(x_ij)=0 for all i∈{0,1},j
for k=1 to K do
    Compute ∇ℓ(H)(x_ij)|_{H=H_{k-1}} by (18)
    Set ℓ_m=0, m=1,...,M
    for m=1 to M do
        for i=0 to 1 do
            for j=0 to N+L-1 do
                ℓ_m ← ℓ_m + h_m(x_ij)∇ℓ(H)(x_ij)|_{H=H_{k-1}}
            end for
        end for
    end for
    m* = argmax_m (ℓ_m)
    h_k(x_ij) ← h_{m*}(x_ij)
    H_k(x_ij) ← H_k(x_ij) + h_k(x_ij)
end for
```

Output: Classifier $H_K(x_{ij}) = \sum_{k=1}^{K} h_k(x_{ij})$, and $p(y=1|\cdot)=\sigma(H_K(\cdot))$

2.5. Discussion

We discuss the advantages of our tracker over the closely related MIL tracker [15] and related work.

2.5.1. Equal or different weights

In (11), We weight larger on the samples near the tracking result (i.e., x_0) at the current frame based on the assumption that the tracking result at the current frame can be assumed to be the most *correct* positive sample. This assumption is used in most generative models [2–6] and some discriminative models [7,8,10,12] with one positive sample. In fact, it is impossible to ensure a complete drift-free tracker without any prior model and learning everything online [30]. However, our tracker can well deal with the drift problem under an MIL framework. Thus, the tracking result at current frame can be assumed to be the most *correct* positive sample which contributes largest to

the probability of positive bag among all the positive samples. If we utilize the equal weights for different positive samples, the classifier can become confused which cannot discriminate the most *correct* positive instance because each positive sample contributes equally to the probability of the positive bag. Similar problem also exists on the OAB(45) tracker (*cf.* Fig. 1(b)) in [15].

2.5.2 *Effective feature selection*

As demonstrated by Babenko et al. [15], using online MIL method can alleviate the location ambiguity problem in tracking task. However, the MIL tracker may still suffer from unstable results in some very challenging sequences [18,19], because the Noisy-OR model used by MIL tracker does not discriminatively treat the sample importance, leading it easily to select less effective features [18]. However, our tracker naturally integrates the sample importance into the learning procedure which makes it able to select the most effective features, resulting in a much more stable result than MIL tracker.

2.5.3 *Efficient feature selection*

Since we integrate the information of the sample importance when training the classifier, our tracker can select the most effective features which can much better discriminate the object from its background than MIL tracker. Therefore, our tracker only needs to select less features to yield better discrimination than MIL tracker. In our experiments, we only need to select $K=15$ features from a feature pool with $M=150$ features, which is much less computational complexity than MIL tracker which sets $K=50$ and $M=250$. Note that we also tested the parameter setting $K=15$, $M=150$ in MIL tracker but found the results are unstable for most experiments. From Algorithms 1 and 2, we can see that the mainly *computational complexity* for MIL tracker and our tracker is the number of feature combinations (i.e., MK). Thus, the number of feature combinations for our tracker ($M=150$, $K=15$) is about 5.5 times less than that of MIL tracker ($M=250$, $K=50$). Moreover, as shown in Algorithm 2, our tracker does not compute the instance probability and bag probability (i.e., Steps 8 and 10 in Algorithm 1) M times after selecting each weak classifier which reduces lots of computation burden.

2.6. Image features

The image features we use are the same as those used in [15,27], which are very easy to implement and efficient to compute using the integral image [11]. We represent each sample x by a feature vector $\mathbf{f}(x)=(f_1(x),...,f_K(x))^T$, where $K=15$. Each feature $f_i(\cdot)$ is a Haar-like feature computed by the sum of weighted pixels in several rectangles. The pixels in the same rectangle have the same weight that is randomly generated from the range (0,1]. The number of rectangles for each sample ranges from 2 to 4. The heights and the weights of the rectangles are generated randomly. The locations of these rectangles in the sample are also randomly selected.

3. Experiments

In this section, we compare our WMIL tracker with 5 latest state-of-the-art trackers on 10 challenging video clips. We use the source codes or the binaries released by the authors[1, 2, 3, 4] with default parameter settings except for the IVT method [4] that we empirically tune its parameters for best performance. The five trackers are fragment (Frag) tracker [5], online AdaBoost (OAB) tracker [12], Semi-supervised Boosting (SemiB) tracker [14], IVT method [4] and multiple instance learning (MIL) tracker [15]. Moreover, to demonstrate the effect of different combinations (i.e., weight, bag probability and likelihood function) of the components of our tracker and MIL tracker, we implement three different MIL trackers (i.e., EWMIL, MIL1, MIL2) with different combinations of weight, bag probability and likelihood function. These three different MIL trackers with the MIL tracker [15] and our WMIL tracker are summarized in Table 1. The eight video clips (from *Sylvester* to *Tiger 2* in Table 1) are provided by the authors[3] [15] to evaluate the performance of MIL tracker while two video clips (*Boy* and *Biker*) are our own. Since all of the comparative algorithms involve some randomness except for the Frag tracker [5], we ran them 10 times on each sequence, and then took the average for comparison. Our WMIL tracker is implemented in MATLAB without any optimization on Windows XP system, and runs at about 27 frames per second (FPS) on a core 2 with CPU 2.10 GHz and 1.95 GB RAM notebook computer. We have also implemented the MIL tracker algorithm in MATLAB but the speed is about 1.5 FPS by using the same computer which means our tracker (27 FPS) is more than 18 times faster than MIL tracker. The speeds of other compared methods are shown by the bottom row of Tables 2 and 3. Although all our compared methods except for IVT method are implemented by C or C++ which is intrinsically efficient than MATLAB, our WMIL tracker implemented in MATLAB still runs fastest. The MATLAB source code can be found at http://code.google.com/p/online-weighted-miltracker/. The videos for the comparison experiments can be found at http://www.youtube.com/user/zhkhua/videos

Table 1
Five different MIL trackers with different combinations of weight, bag probability and likelihood function.

Tracker	Weight	Bag probability	Likelihood function
EWMIL	Equal	Weighted sum model (10)	Approximate
MIL1	–	Noisy-OR model (8)	Approximate
MIL2	Different	Weighted sum model (10)	Original
MIL [15]	–	Noisy-OR model (8)	Original
Ours	Different	Weighted sum model (10)	Approximate

Table 2
Failure rate (FR) (%) and average frames per second (FPS) for our tracker and state-of-the-art trackers. Bold fonts indicate the best performance, while the italic fonts indicate the second best ones. (Total number of evaluated frames is 5593.)

Sequence	Ours	IVT	OAB	SemiB	Frag
Sylvester	**27**	55	*31*	33	66
David indoor	**0**	**0**	*68*	55	92
Occluded face	*3*	**0**	10	4	**0**
Occluded face2	**3**	*13*	54	59	48
Twinings	*8*	51	**3**	78	31
Cliff bar	**7**	54	76	*37*	78
Tiger 1	**34**	92	77	*71*	81
Tiger 2	**39**	82	*61*	81	87
Biker	**46**	*57*	61	60	77
Boy	**5**	61	*28*	55	66
Overall	**16**	*40*	41	46	56
Average FPS	**27**	*11*	8	6	3

3.1. Parameter settings

We use a radius $\alpha=4$–8 to crop the positive samples at each frame. This generates 45–190 positive samples. For most sequences,

[1] http://www.cs.technion.ac.il/~amita/fragtrack/fragtrack.htm.
[2] http://www.vision.ee.ethz.ch/boostingTrackers.
[3] http://vision.ucsd.edu/~bbabenko/project_MIL tracker.shtml.
[4] http://www.cs.toronto.edu/~dross/ivt.

Table 3
Failure rate (FR) (%) and average frames per second (FPS) for different MIL trackers. Bold fonts indicate the best performance, while the italic fonts indicate the second best ones. (Total number of evaluated frames is 5593.)

Sequence	Ours	EWMIL	MIL1	MIL2	MIL
Sylvester	27	32	*25*	30	**22**
David indoor	**0**	*6*	*6*	17	34
Occluded face	**3**	77	41	29	*5*
Occluded face2	3	50	*2*	**1**	**1**
Twinings	**8**	33	55	*16*	29
Cliff bar	**7**	16	59	*8*	33
Tiger 1	*34*	52	46	**23**	60
Tiger 2	**39**	63	62	*45*	57
Biker	**46**	*52*	69	64	77
Boy	*5*	14	47	**4**	59
Overall	**16**	42	34	*23*	28
Average FPS	**27**	**27**	**27**	**27**	*10*[a]

[a] The FPS is 1.5 in our implemented MATLAB code.

Table 4
Center location errors (in pixels) for our tracker and state-of-the-art trackers. Bold fonts indicate the best performance, while the italic fonts indicate the second best ones. (Total number of evaluated frames is 5593.)

Sequence	Proposed tracker			IVT			OAB			SemiB			Frag		
	Max	Mean	Std	Max	Mean	Std	Max	Mean	Std	Max	Mean	Std	Max	Mean	Std
Sylvester	**28**	**11**	**5**	484	137	140	*64*	*13*	*11*	88	*13*	13	138	47	40
David indoor	**18**	**8**	**4**	*24*	**8**	*6*	146	59	39	106	*36*	28	146	73	37
Occluded face	58	16	11	*33*	*12*	8	110	17	27	55	17	**5**	**29**	**9**	*6*
Occluded face2	**30**	**13**	**5**	40	*16*	*10*	102	34	23	118	36	39	140	58	53
Twinings	**27**	**8**	**5**	67	23	21	*30*	*9*	**5**	150	71	70	49	15	*12*
Cliff bar	**24**	**8**	**4**	148	37	40	129	35	42	144	56	35	*121*	*34*	*27*
Tiger 1	**42**	**10**	**7**	107	45	27	105	41	30	151	40	36	*99*	*39*	*25*
Tiger 2	**22**	**9**	**4**	125	43	29	94	*21*	*21*	*88*	27	*21*	125	37	25
Biker	**42**	**16**	**7**	247	52	65	*89*	*28*	*16*	130	30	24	153	50	36
Boy	**40**	**11**	**8**	313	107	90	*51*	*18*	*13*	405	124	130	288	119	90
Overall	**58**	**11**	**6**	484	56	52	*146*	*25*	*21*	405	35	31	288	44	34

Table 5
Center location errors (in pixels) for different MIL trackers. Bold fonts indicate the best performance, while the italic fonts indicate the second best ones. (Total number of evaluated frames is 5593.)

Sequence	Proposed tracker			EWMIL			MIL1			MIL2			MIL		
	Max	Mean	Std	Max	Mean	Std	Max	Mean	Std	Max	Mean	Std	Max	Mean	Std
Sylvester	**28**	*11*	**5**	**28**	12	*6*	64	*11*	10	42	12	*6*	*31*	**10**	7
David indoor	**18**	**8**	**4**	*29*	*11*	*6*	36	*11*	7	37	16	8	41	18	9
Occluded face	*58*	**16**	**11**	90	57	24	77	32	*20*	106	35	35	**55**	*19*	**11**
Occluded face2	**30**	**13**	**5**	50	27	12	**30**	**13**	8	**30**	*14*	*6*	*36*	17	7
Twinings	*27*	**9**	**5**	40	16	16	37	21	8	34	*10*	8	**26**	15	*7*
Cliff bar	**24**	**8**	**4**	*26*	*9*	*5*	55	19	12	32	**8**	*5*	46	15	8
Tiger 1	*42*	**10**	*7*	**39**	*13*	**6**	64	*13*	9	100	*13*	21	116	27	28
Tiger 2	**22**	**9**	**4**	*25*	*12*	*5*	59	*12*	8	129	32	37	99	19	17
Biker	**42**	**16**	**7**	255	53	75	67	*21*	*12*	291	59	86	*51*	24	*12*
Boy	**40**	**11**	*8*	57	*18*	11	267	40	52	*47*	**11**	**7**	204	57	54
Overall	**58**	**12**	**6**	255	24	14	267	*18*	*13*	291	20	18	*204*	19	*13*

Table 6
Statistical test results for all compared trackers. LC represents location, while FR represents the failure rate. Bold font indicates the performance that the hypothesis H_1 is accepted.

Sequence	IVT		OAB		SemiB		Frag		EWMIL		MIL1		MIL2		MIL	
	LC	FR	LC	FR	LC	FR	LC	FR	LC	FR	LC	FR	LC	FR	LC	FR
Sylvester	**140**	**30**	**4**	**6**	**6**	**10**	**63**	**65**	**11**	**5**	**6**	−1.5	**8**	**3**	−2	−2
David indoor	**2**	–	**39**	**10**	**45**	**12**	**89**	**80**	**12**	**3**	**13**	**4**	**15**	**8**	**13**	**12**
Occluded face	−3	−2	**2**	**3**	**2**	**5**	−2	−1.5	**13**	**30**	**12**	**21**	**3**	**20**	**6**	**30**
Occluded face2	**5**	**6**	**4**	**20**	**11**	**25**	**20**	**50**	**11**	**20**	**4**	−1	**3**	−1	**5**	−2
Twinings	**12**	**20**	**2**	−2	**20**	**50**	**5**	**30**	**5**	**20**	**10**	**25**	**3**	**15**	**9**	**18**
Cliff bar	**13**	**20**	**22**	**40**	**20**	**80**	**11**	**50**	**2**	**21**	**8**	**24**	**2**	**32**	**9**	**22**
Tiger 1	**31**	**60**	**23**	**30**	**18**	**95**	**13**	**30**	**3**	**15**	**4**	**12**	**6**	**6**	**17**	**19**
Tiger 2	**10**	**60**	**8**	**80**	**14**	**30**	**20**	**24**	**4**	**12**	**5**	**21**	**16**	**3**	**7**	**9**
Biker	**10**	**74**	**6**	**45**	**15**	**21**	**32**	**30**	**10**	**8**	**5**	**11**	**25**	**9**	**9**	**21**
Boy	**101**	**18**	**8**	**26**	**189**	**46**	**128**	**39**	**6**	**6**	**20**	**10**	1	**3**	**30**	**14**
Overall	**44**	**26**	**9**	**19**	**23**	**28**	**37**	**42**	**9**	**14**	**8**	**10**	**7**	**9**	**7**	**11**

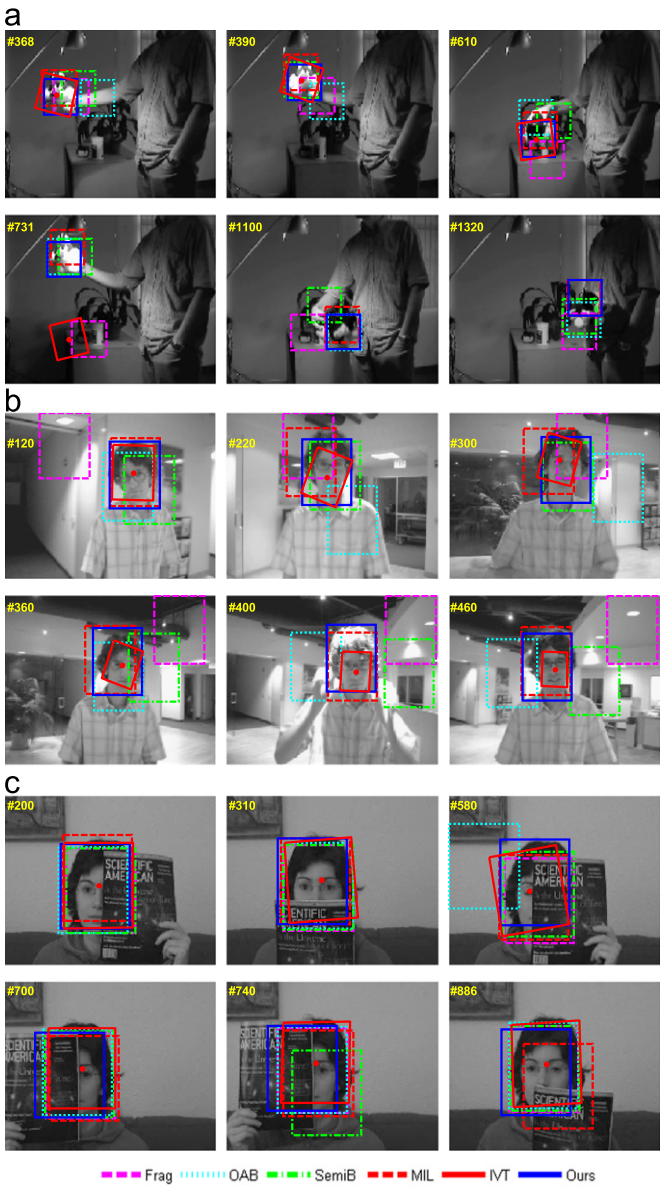

Fig. 4. Sampled tracking results for tested sequences *Sylvester* (a), *David indoor* (b) and *occluded face* (c).

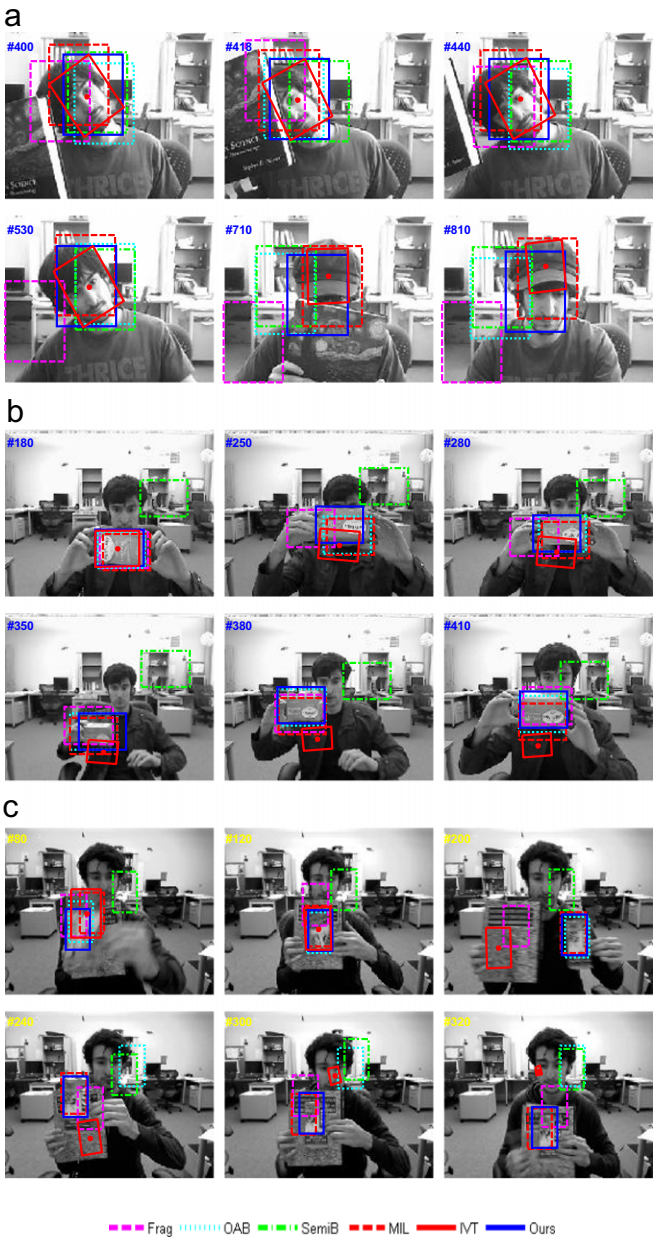

Fig. 5. Sampled tracking results for tested sequences *Occluded face2* (a), *Twinings* (b) and *Cliff bar* (c).

we found the results are robust with different α. Only when the appearance changes very fast, we need a large α (refer to *boy* sequence where we set $\alpha=8$) to capture more positive samples. The inner and outer radii for the negative sample set $X^{\zeta,\beta}$ are set as $\zeta=2\alpha$ and $\beta=1.5\gamma$, respectively. These settings can crop negative samples with less overlap with the positive samples. Then, we uniformly randomly select 42–100 negative samples from set $X^{\zeta,\beta}$ because this set includes large number of negative samples. We found the results are robust with different numbers of selected negative samples. We think the reasons are twofold: first, the negative samples have less overlap with the positive samples; second, we uniformly and randomly select the negative samples which can cover most of the representative negative samples around the target object. Thus, the selected small number of negative samples contain enough discriminative information to separate target object from its neighbor background. The radius for searching the new object location at the next frame is set $\gamma=25$–35. Note that this generates 2000–3800 samples which is much larger than the training positive samples (45–190) and negative ones (42–100). Thus, the detection procedure is the most time-consuming part in our tracker. We found for most of our test sequences, the results were robust to the different γ. However, when the object moves very fast (e.g., *boy* sequence), we should set a large γ to capture the object at the next frame because the distance of the object locations between two consecutive frames will be very large. We set $\gamma=25$ for all of our test sequences except for the *boy* sequence where we set $\gamma=35$. Our WMIL tracker selects $K=15$ (we found the algorithm is fairly robust to the range $K=15$–50) features to design the classifier which is much more efficient than MIL tracker that sets $K=50$ (we found only $K\geq 50$ can generate good results for most of our test sequences). The number of candidate features in the feature pool is set $M=90$–250. If the object appearance changes fast and largely, then M should be set larger in order to contain enough different candidate features to make the selected features more discriminative. However, a large M also increases the computational complexity. Thus, we set $M=150$ (less than MIL tracker ($M=250$)) for all our test sequences as a tradeoff between the efficiency and the discriminative

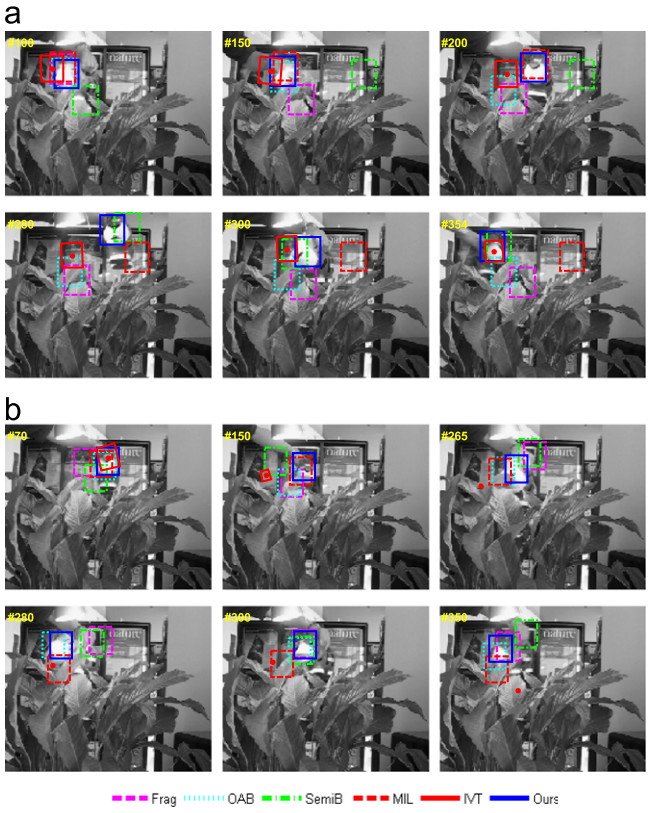

Fig. 6. Sampled tracking results for tested sequences *Tiger 1* (a) and *Tiger 2* (b).

capability of the selected features. The learning parameter η is set $\eta=0.7$–0.9. From (15) and (16), we can see that a large η makes the updated parameters weight more on the old parameters. Thus, if the object appearance does not change very fast over time, we can set a large η to make the parameters keep as stable as possible. Otherwise, we set a small η to adapt the large appearance changes (e.g., *Biker* and *Boy* sequences). We set $\eta=0.85$ for all of our test sequences except for *Biker* and *Boy* sequences which we set $\eta=0.7$.

3.2. Quantitative analysis

All of the tested sequences are gray-scale. We use two measures [31] to compare the proposed tracker and the reference trackers. The first is the failure rate (FR) which is defined as the number of failure frames divided by the total number of frames in a video sequence. Based on the evaluation metric of the PASCAL VOC object detection [34] which is also applied to evaluate the performance of some state-of-the-art tracking algorithms in [19], the failure frame is indicated when the intersection of the ground truth bounding box and the tracking bounding box is less than half of the union of the ground truth bounding box and the tracking bounding box. Tables 2 and 3 show the average failure rate. We can see that our tracker achieves the best or second best performance compared with IVT, OAB, SemiB and Frag. Moreover, our tracker achieves the best or second best performance for most of sequences when compared with the different MIL trackers. Only for the *Sylvester* and *Occluded face2* sequences, the performance of our tracker is somewhat inferior to MIL1 and MIL trackers. The second measure is the center location error which is defined as the Euclidean distance from the detected object

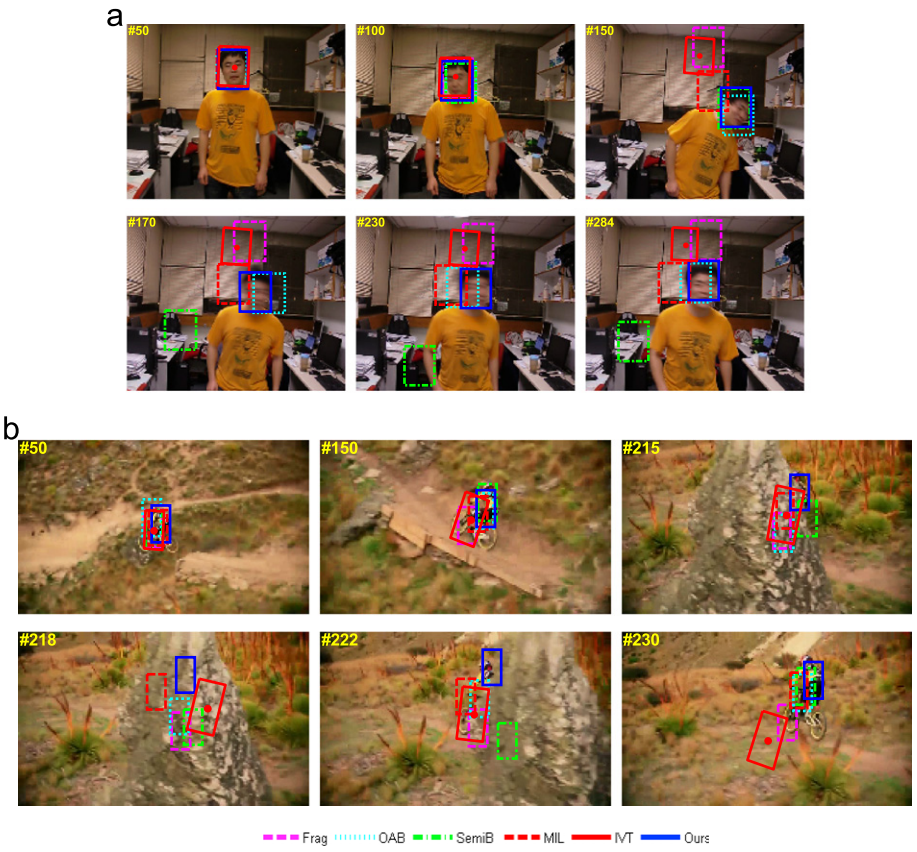

Fig. 7. Sampled tracking results for tested sequences *Boy* (a) and *Biker* (b).

center to the ground truth center at each frame. The maximum, mean and standard deviation of the center location error per frame are summarized in Tables 4 and 5. We can see that our tracker runs the best or second best performance on the mean location error measure in all the tested trackers for all sequences except for the *Occluded face2* sequence. Moreover, our tracker runs the best based on the standard deviation measure for all sequences except for the *Occluded face2* sequence, which means our tracker can achieve much more accurate and stable results than other compared trackers. Overall, our tracker achieves the lowest failure rates, maximum and average location errors and standard deviation of location errors among all the compared trackers.

In order to further evaluate the performance of our tracker, we use a standard statistical one-sided hypothesis testing [31,32]. The null hypothesis H_0 states the proposed tracker is not superior to the reference trackers, while the hypothesis H_1 states the proposed tracker is much better than the reference trackers. At the ith repetition, a sample-performance-difference [32] is defined as

$$\Delta^i = C^i_{REF} - C^i_{WMIL} \tag{20}$$

where C^i_{REF} and C^i_{WMIL} denote the quantitative performance of the reference tracker and the proposed WMIL tracker, respectively. C^i represents the mean location error or the failure rate at ith repetition [31]. The sample-performance-difference mean is

$$\overline{\Delta} = \frac{1}{I}\sum_{i=1}^{I} \Delta^i \tag{21}$$

and its standard error is

$$\sigma_{\overline{\Delta}} = \sqrt{\frac{1}{I^2}\sum_{i=1}^{I} (\Delta^i - \overline{\Delta})^2} \tag{22}$$

The hypothesis H_1 is accepted (H_0 is rejected) at a significance level of α if the test statistic $\overline{\Delta}/\sigma_{\overline{\Delta}} > \mu_\alpha$, where μ_α is a point on the standard Gaussian distribution corresponding to the upper-tail probability of α. We set $\mu_\alpha = 1.65$ where its corresponding significant level $\alpha = 0.05$ [31,32]. Table 6 shows the results of the hypothesis testing on location error and failure rate for all the tested video sequences using the compared trackers. We can see that the test statistic in Table 6 is larger than $\mu_\alpha = 1.65$ for most of experiments. Furthermore, the hypothesis H_1 is accepted based on the overall performance for all the experiments. Thus, it is reasonable to infer that our tracker outperforms other compared trackers.

3.3. Qualitative analysis

3.3.1. Sylvester and David indoor

These two sequences comprise challenging scale, illumination and pose changes. The Frag tracker cannot adaptively adjust these changes, resulting in serious drift (see all the frames shown in Fig. 4(a)). The OAB tracker also yields severe drift problem as shown by frames #368, #390 of *Sylvester* sequence and frames #220, #300, #400, #460 of *David indoor* sequence. The SemiB tracker drifts away at frames #360, #400, #460 of *David indoor* sequence. The IVT method drifts away after frame #731 of *Sylvester* sequence. Although MIL tracker yields more stable results than Frag, OAB and SemiB and IVT in general, some of its results (see frames #220, #300 of *David indoor* sequence) are imprecise because the Nosy-OR model used by MIL tracker may select the less effective features [18]. Our WMIL tracker achieves the best performance compared with other five trackers for these two sequences.

3.3.2. Occluded face and Occluded face2

These two sequences comprise heavily partial occlusions. The *Occluded face* sequence in Fig. 4(c) is designed by the author of Frag tracker [5]. The Frag tracker works well for this sequence because it is based on the part-based model which can solve the partial occlusion well. However, in *Occluded face2* sequence (see Fig. 5(a)), when the appearance changes much (e.g., when the face turns, the hat is put on the head or the face reappears), Frag tracker works poorly (see the frames #418, #530, #710, #810 of *Occluded face2* in Fig. 5(a)). The OAB and SemiB trackers also perform poorly when the appearance changes significantly (see the frames #440, #710, #810 of *Occluded face2*). The IVT method works well on the *Occluded face* sequence. However, when the occlusion is severe in *Occluded face2* (see frames #710 in *Occluded face2*), IVT drifts to the region of cap. Then, it updates its appearance model based on the appearance of the cap, and finally snaps to the region of cap (see #810 in *Occluded face2*). The MIL tracker yields some unstable results when the face reappears (see frame #886 of *Occluded face* and frames #440, #530, #810 of *Occluded face2*). In general, our WMIL tracker performs well when the partial occlusions and appearance changes much at the same time.

3.3.3. Twinings and Cliff bar

Both of these two sequences comprise large scale appearance changes. Moreover, the *Cliff bar* sequence also exhibits serious motion blur and very similar texture between the background and the object. As shown by frames #250, #280 in *Twinings*

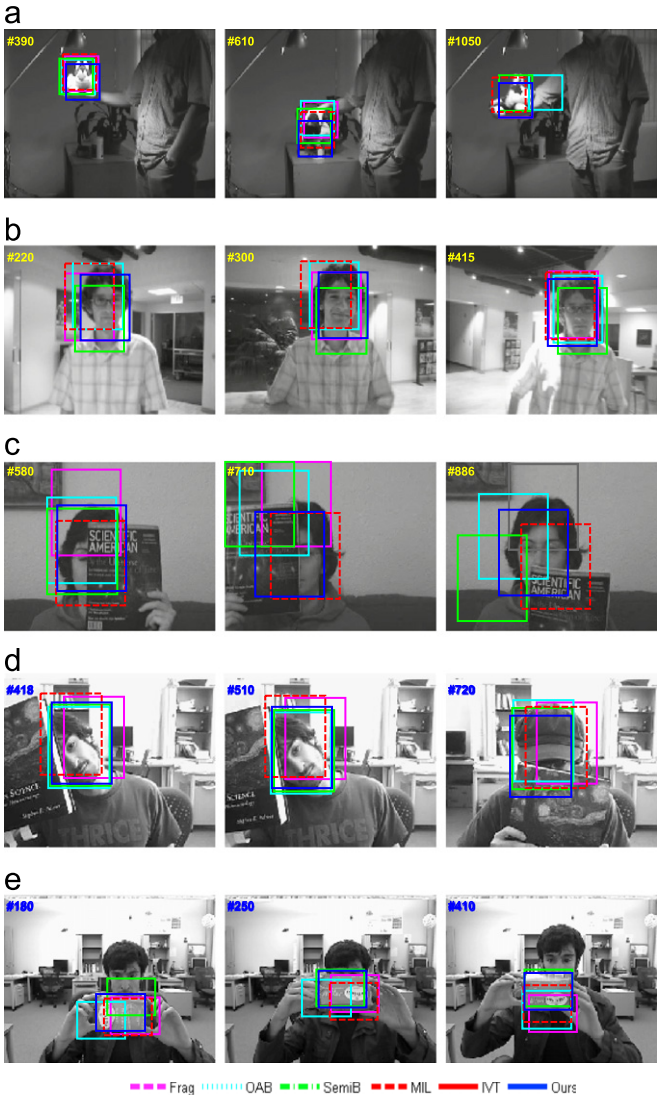

Fig. 8. Sampled tracking results for tested sequences: (a) Sylvester, (b) David indoor, (c) Occluded face, (d) Occluded face 2, and (e) Twinings.

sequence, the appearance of the box changes much because of the rotation. Frag, SemiB and IVT methods drift much because they cannot handle well the significant appearance variations. When the scale and orientation changes (in *Cliff bar* sequence, the object is rotated upside down during tracking), both our method and MIL tracker perform well because they utilize the same features which are invariant to scale and orientation changes. The SemiB tracker performs poorly which drifts away for most frames (see the error plots in Fig. 11). As shown by the frames #120, #200 of *Cliff bar* sequence, the Frag tracker is distracted to track the background because the background and the object have a very similar texture. The IVT method also snaps to the background because it is a generative model that does not utilize the background information, which limits its discriminative capability. The tracker ultimately drifts to the background because the background texture is very similar to the object of interest. Frames #80 and #320 of *Cliff bar* show the results by MIL tracker are inaccurate because of the severe motion blur. Our WMIL tracker can generally handle scale and appearance changes well, yielding a much more stable and accurate result than other compared trackers.

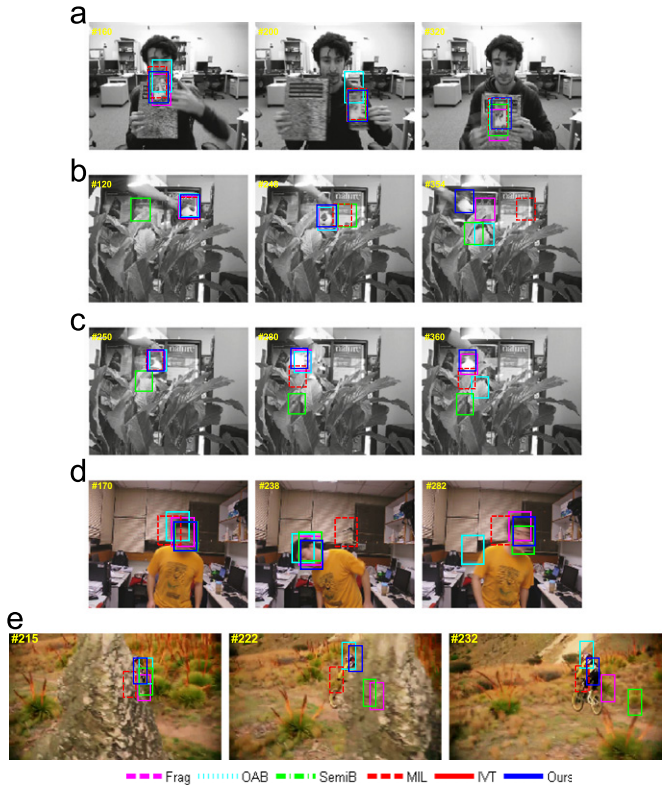

Fig. 9. Sampled tracking results for tested sequences: (a) Cliff bar, (b) Tiger 1, (c) Tiger 2, (d) Boy, and (e) Biker.

3.3.4. *Tiger 1 and Tiger 2*

These two sequences comprise illumination changes, pose variations, motion blur, and partial occlusion at the same time which make them very challenging. The pose changes of the toy tigers also include out of plane rotations (the faces of toy tigers rotate from left to right shown by frames #300, #354 of *Tiger 1* and #280, #300, #350 of *Tiger 2*). Although the MIL tracker in general performs well for other 6 sequences mentioned above, it finally drifts away for these two challenging sequences (at frames #280, #300, #354 of *Tiger 1* and #280, #300, #350 of *Tiger 2*). From the sampled results shown by Fig. 6 and the error plots in Fig. 11, we can see Frag, OAB, SemiB and IVT all drift away at many frames (e.g., #200, #280, #300, #354 of *Tiger 1* and #150, #265, #280, #350 of *Tiger 2*). Our WMIL tracker can yield more stable and more accurate results than the other five trackers when the object undergoes challenging illumination changes, pose variations, motion blur, and partial occlusion.

3.3.5. *Boy and Biker*

These two sequences mainly comprise very fast motion. For the *Biker* sequence, the object is fully occluded for some frames (Frames #216∼#221 in *Biker* sequence) which makes it very challenging to re-detect the object of interest. As shown by frames #170, #230 of *Boy* sequence, the appearance of object of interest is drastically changed because of motion blur. All the compared trackers except for OAB cannot handle the severe appearance variations well and exhibit severe drift (see #170, #230, #284 in *Boy* sequence). Although OAB can track the object of interest all the time, its results are imprecise at some frames (see #170, #230, #284 in *Boy* sequence). In general, our tracker achieves the best performance in terms of both accuracy and robustness. The *Biker* sequence comprises partial or full occlusion frames from #215 to #222. The appearance of object of interest changes drastically and fast during these frames (see frames #215, #222 (partial occlusion), #218 (full occlusion) in *Biker* sequence). All of our compared trackers drift to the region of stone when the object begins to move or reappear from the back of the stone (see frames #215, #222 in *Biker* sequence) (Fig. 7). Our tracker can track the object of interest precisely and stably all the time for this sequence because of the following reasons: first, we set a relatively small learning parameter (i.e., $\eta=0.7$) which makes the

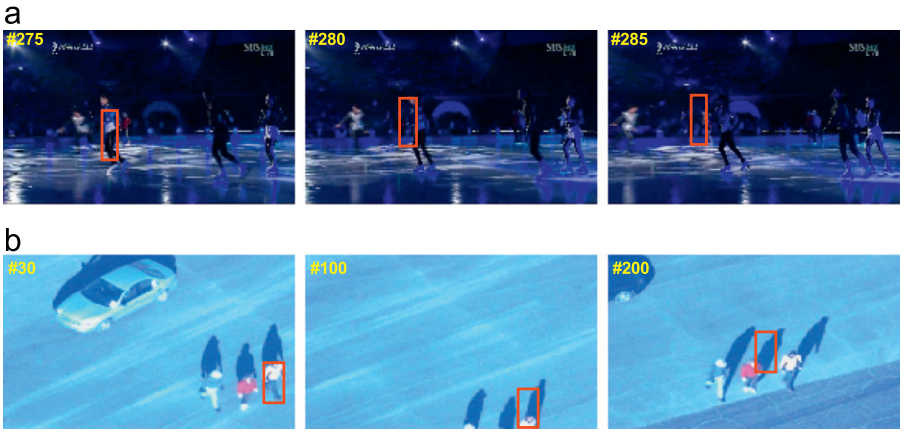

Fig. 10. Two failed tracking cases: (a) severe illumination changes and (b) object of interest leaves completely out of screen and reappears (a small search radius is used).

updated parameter weight large on the features extracted from the recent frames. Thus, the updated weak classifiers can quickly adapt the fast appearance changes. Second, the selected features by our weighting scheme are much more discriminative than those by MIL tracker because our tracker integrates the sample importance into its learning procedure.

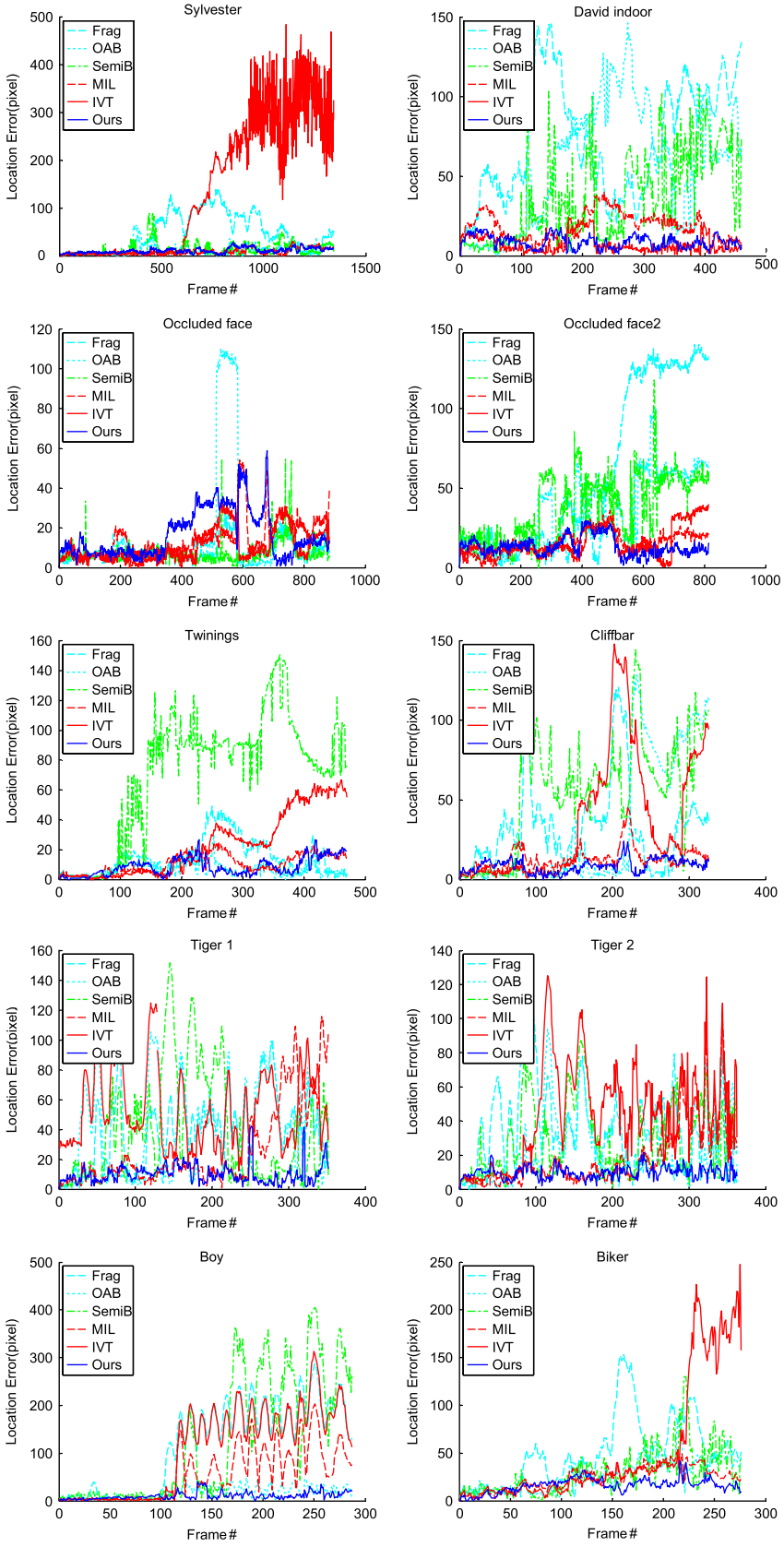

Fig. 11. Error plots of all tested sequences for state-of-the-art algorithms.

3.4. Comparisons between different MIL trackers

The five MIL trackers with different combinations of weight, bag probability and likelihood function are summarized in Table 1. We also utilize the above 10 sequences to evaluate the 5 different MIL trackers. The quantitative comparisons are shown by Tables 3, 5 and 6. Fig. 12 shows the error plots of different MIL trackers applied to all of the tested sequences. We can observe that our tracker

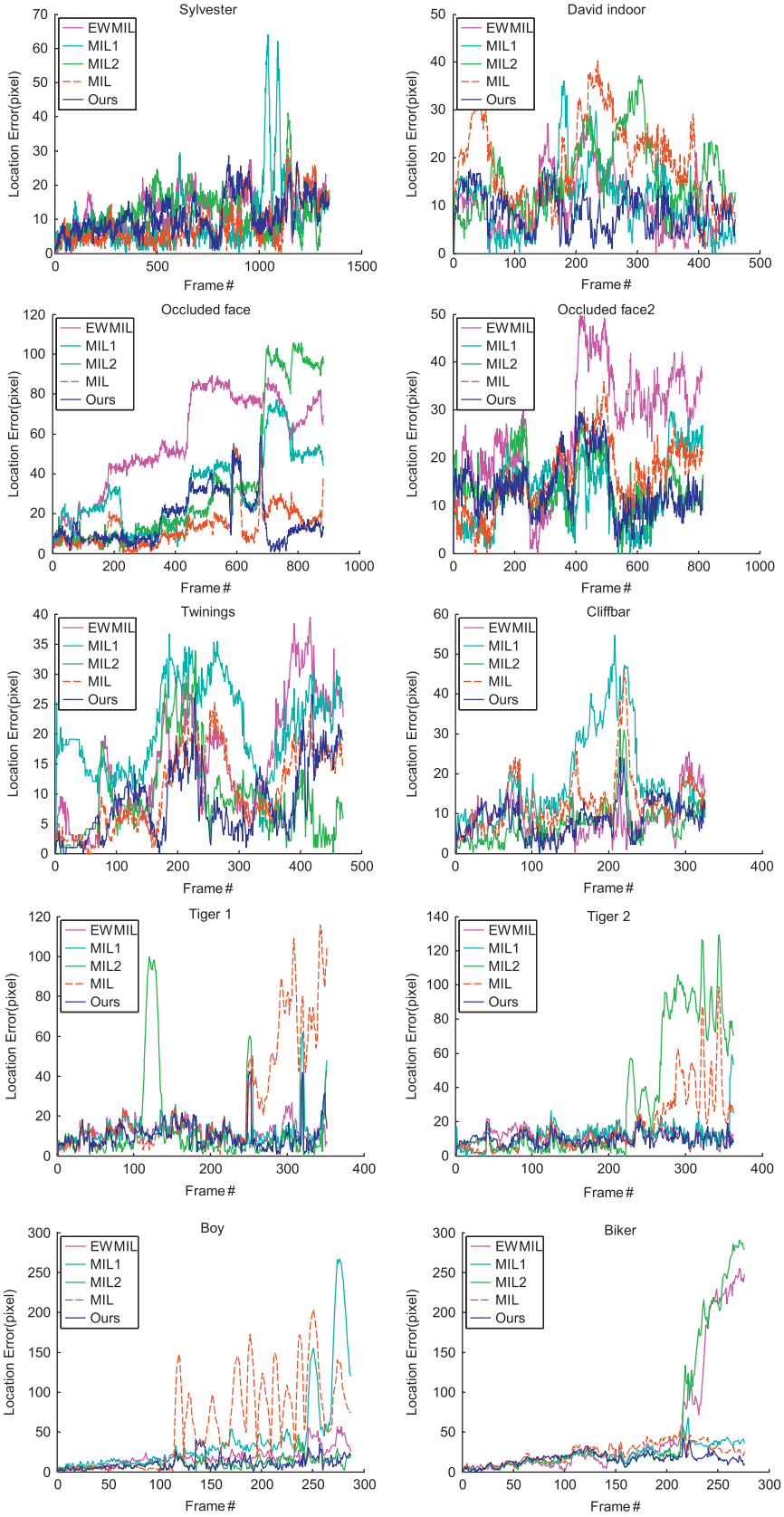

Fig. 12. Error plots of all tested sequences for different MIL trackers.

achieves the best or second best performance for most of tested sequences, and the overall performance of our tracker is best based on all these measures. Figs. 8 and 9 show some sampled results for the tested sequences. We can see that the EWMIL tracker cannot work well when the appearance of object of interest changes much (frames #1050 of *Sylvester*, #580, #710, #886 of *Occluded face*, #320 of *Cliff bar*, #250, #410 of *Twinings*, #282 of *Boy*, #222, #232 of *Biker*). Different from MIL tracker [15] whose objective function is likelihood function, the MIL1 tracker uses the approximated likelihood function. As shown by frames #1050 of *Sylvester* sequence, #180 of *Twinings* sequence, #200 of *Cliff bar* sequence, MIL tracker works better than MIL1 tracker whose results have obvious drift. However, for some frames (see frames #120, #248 of *Tiger 1* sequence, #250, #280 of *Tiger 2* sequence, #170, #238 of *Boy* sequence, #215, #222 of *Biker* sequence), MIL1 tracker achieves better performance than MIL tracker. These sequences often comprise large scale pose variations or drastic appearance changes. The MIL2 tracker uses the weighted sum model as our tracker but its objective function is a likelihood function that is different from our tracker. We can observe the MIL2 tracker works poorly for some sequences with large scale pose variations and appearance changes (see frames #120, #248, #354 of *Tiger 1* sequence, #250, #280, #360 of *Tiger 2* sequence, #222, #232 of *Biker* sequence). Our tracker achieves the best performance for the most of sequences in terms of accuracy and robustness. Therefore, we can conclude both the weighted scheme and the approximate likelihood function affect the performance of the tracker and the combination of these two terms (e.g., our WMIL tracker) can make the tracker yield much more robust and accurate results than those only use one term.

3.5. Discussion

Our experiments have demonstrated the good performance of our tracker in terms of accuracy, robustness and efficiency. However, our tracker may fail when the illumination changes drastically or the object of interest leaves completely out of screen and reappears. Fig. 10(a) (the sequence is proved by the authors of [33]) shows the illumination changes drastically after frames #280. Our tracker drifts away because of the very low contrast between the background and the object of interest caused by illumination changes. Fig. 10(b) shows the object of interest completely out of the screen and reappears. This case is similar to that of *Biker* sequence in Fig. 9(e) where the object of interest is fully occluded by a stone at some frames. However, different from the *Biker* sequence whose camera is stable, the camera of this sequence in Fig. 10(b) shakes fast which makes the distance between the objects at two consecutive frames very far. Thus, we should set a large search radius γ to ensure the object of interest is in the cropped samples when it reappears. Fig. 10(b) shows the results when we set $\gamma=25$, and our tracker drifts away ultimately. If we set $\gamma=35$, our tracker can yield a stable results for the sequence in Fig. 10(b). Refer to the videos *Pedestrian25.mp4* and *Pedestrian35.mp4* at http://www.youtube.com/user/zhkhua/videos

4. Conclusion

In this paper, we presented a novel online weighted multiple instance learning (WMIL) tracker which can naturally integrate the sample importance into the learning procedure. The WMIL tracker assumes the tracking location at current frame is the location of most *correct* positive sample to make each instance contribute differently to the bag probability: the farther the instance is from the tracking location, the less it contributes to the bag probability. Our tracking system is very easy to implement and can reach at a real-time speed in MATLAB. Experimental results on challenging video sequences have demonstrated the superiority of our system to state-of-the-art tracking systems in accuracy, stability and efficiency. Our WMIL technique can be extended to other tracking algorithms such as semi-supervised tracking system to make them select more effective features by using the proper assumption. Moreover, it can be applied to other applications in computer vision such as object detection.

Appendix A. Lemma 1 and its proof

Lemma 1. *$D_{t-1}=\{f(x_0),\ldots,f(x_{n-1})\}$ denotes the feature extracted from all of the positive samples from the first frame to the $(t-1)$th frame; $D=\{f(x_n),\ldots,f(x_{n+m-1})\}$ denotes the feature extracted from the current positive samples; $D_t=\{D_{t-1},D\}$. Suppose all of the features in D_t are independent and Gaussian distributed with the same mean μ_t and standard deviation σ_t. Then, the maximum likelihood estimates for μ_t and σ_t are*

$$\tilde{\mu}_t=\eta\tilde{\mu}_{t-1}+(1-\eta)\tilde{\mu} \tag{A.1}$$

$$\tilde{\sigma}_t=\sqrt{\eta\tilde{\sigma}_{t-1}^2+(1-\eta)\tilde{\sigma}^2+\eta(1-\eta)(\tilde{\mu}_{t-1}-\tilde{\mu})^2} \tag{A.2}$$

where $\tilde{\mu}_{t-1}=(1/n)\sum_{i=0}^{n-1}f(x_i)$, $\tilde{\sigma}_{t-1}=\sqrt{(1/n)\sum_{i=0}^{n-1}(f(x_i)-\tilde{\mu}_{t-1})^2}$, $\tilde{\mu}=(1/m)\sum_{i=n}^{n+m-1}f(x_i)$, $\tilde{\sigma}=\sqrt{(1/m)\sum_{i=n}^{n+m-1}(f(x_i)-\tilde{\mu})^2}$ *and* $\eta=n/(m+n)$.

Proof. It is easy to obtain the maximum likelihood estimates for μ_t and σ_t as $\tilde{\mu}_t=(1/(m+n))\sum_{i=0}^{m+n-1}f(x_i)$ and $\tilde{\sigma}_t=\sqrt{(1/(m+n))\sum_{i=0}^{m+n-1}(f(x_i)-\tilde{\mu}_t)^2}$, respectively [21]. Then, we have

$$\begin{aligned}
\tilde{\mu}_t&=\tfrac{1}{m+n}\sum_{i=0}^{m+n-1}f(x_i) & \tilde{\sigma}_t&=\sqrt{\tfrac{1}{m+n}\sum_{i=0}^{m+n-1}(f(x_i)-\tilde{\mu}_t)^2}\\
&=\tfrac{1}{m+n}\left(\sum_{i=0}^{n-1}f(x_i)+\sum_{i=n}^{m+n-1}f(x_i)\right) & &=\sqrt{\tfrac{1}{m+n}\sum_{i=0}^{m+n-1}f^2(x_i)-\tilde{\mu}_t^2}\\
&=\eta\tfrac{1}{n}\sum_{i=0}^{n-1}f(x_i)+(1-\eta)\frac{1}{m}\sum_{i=n}^{m+n-1}f(x_i) & &=\sqrt{\tfrac{1}{m+n}\left(\sum_{i=0}^{n-1}f^2(x_i)+\sum_{i=n}^{m+n-1}f^2(x_i)\right)-(\eta\tilde{\mu}_{t-1}+(1-\eta)\tilde{\mu})^2}\\
&=\eta\tilde{\mu}_{t-1}+(1-\eta)\tilde{\mu} & &=\sqrt{\frac{1}{m+n}(n\tilde{\sigma}_{t-1}^2+m\tilde{\sigma}^2)+\frac{1}{m+n}(n\tilde{\mu}_{t-1}^2+m\tilde{\mu}^2)-(\eta\tilde{\mu}_{t-1}+(1-\eta)\tilde{\mu})^2}\\
& & &=\sqrt{\eta\tilde{\sigma}_{t-1}^2+(1-\eta)\tilde{\sigma}^2+\eta(1-\eta)(\tilde{\mu}_{t-1}-\tilde{\mu})^2}
\end{aligned}$$

where η, $\tilde{\mu}_{t-1}$, $\tilde{\sigma}_{t-1}$, $\tilde{\mu}$, $\tilde{\sigma}$ are the same as those in (A.1) and (A.2). □

References

[1] A. Yilmaz, O. Javed, M. Shah, Object tracking: a survey, ACM Computing Surveys 38 (2006).
[2] M. Black, A. Jepson, Eigentracking: robust matching and tracking of articulated objects using a view-based representation, in: European Conference on Computer Vision, 1996, pp. 329–342.
[3] A. Jepson, D. Fleet, T. El-Maraghi, Robust online appearance models for visual tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (2003) 1296–1311.
[4] D. Ross, J. Lim, R. Lin, M. Yang, Incremental learning for robust visual tracking, International Journal of Computer Vision 77 (2008) 125–141.
[5] A. Adam, E. Rivlin, I. Shimshoni, Robust fragments-based tracking using the integral histogram, in: IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 798–805.
[6] X. Mei, H. Ling, Robust visual tracking using l1 minimization, in: International Conference on Computer Vision, 2009, pp. 1436–1443.
[7] S. Avidan, Support vector tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2004) 1064–1072.
[8] S. Avidan, Ensemble tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (2007) 261–271.
[9] D. Comaniciu, V. Ramesh, P. Meer, Kernel-based object tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (2003) 564–575.
[10] R. Collins, Y. Liu, M. Leordeanu, Online selection of discriminative tracking features, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2005) 1631–1643.
[11] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: IEEE Conference on Computer Vision and Pattern Recognition, 2001, pp. 511–518.
[12] H. Grabner, M. Grabner, H. Bischof, Real-time tracking via online boosting, in: British Machine Vision Conference, 2006, pp. 47–56.
[13] N. Oza, Online Ensemble Learning, Ph.D. Thesis, University of California, Berkeley, 2001.
[14] H. Grabner, C. Leistner, H. Bischof, Semi-supervised on-line boosting for robust tracking, in: European Conference on Computer Vision, 2008, pp. 234–247.
[15] B. Babenko, M. Yang, S. Belongie, Robust object tracking with online multiple instance learning, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (2011) 1619–1632.
[16] P. Viola, J. Platt, C. Zhang, Multiple instance boosting for object detection, Neural Information Processing Systems (2005) 1417–1426.
[17] Z. Kalal, J. Matas, K. Mikolajczyk, P–N learning: bootstrapping binary classifiers by structural constraints, in: IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 49–56.
[18] Q. Zhou, H. Lu, M. Yang, Online multiple support instance tracking, in: IEEE Conference on Automatic Face and Gesture Recognition, 2011, pp. 545–552.
[19] Q. Wang, F. Chen, W. Xu, M. Yang, An experimental comparison of online object tracking algorithms, in: Proceedings of SPIE: Image and Signal Processing Track, 2011.
[20] A. Webb, Statistical Pattern Recognition, Oxford University Press, New York, 1999.
[21] C. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
[22] C. Leistner, A. Saffari, P. Roth, H. Bischof, On robustness of on-line boosting—a competitive study, in: On-line Learning for Computer Vision Workshop, 2009, 1362–1369.
[23] Y. Freund, R. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences 55 (1997) 119–139.
[24] J. Friedman, T. Hastie, R. Tibshirani, Additive logistic regression: a statistical view of boosting, Annals of Statistics 28 (2000) 337–407.
[25] J. Friedman, Greedy function approximation: a gradient boosting machine, Annals of Statistics 29 (2001) 1189–1232.
[26] J. Wu, J.M. Rehg, M. Mullin, Learning a rare event detection cascade by direct feature selection, Neural Information Processing Systems (2003) 1523–1530.
[27] P. Dollár, Z. Tu, H. Tao, S. Belongie, Feature mining for image classification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.
[28] X. Xu, E. Frank, Logistic regression and boosting for labeled bags of instances, in: Proceedings Eighth Pacific–Asia Conference, 2004, pp. 272–281.
[29] L. Mason, J. Baxter, P. Bartlett, M. Frean, Functional gradient techniques for combining hypotheses, in: A.J. Smola, P.L. Bartlett, B. Schölkopf, D. Schuurmans (Eds.), Advances in Large Margin Classifiers, MIT Press, Cambridge, MA, 2000, pp. 221–247.
[30] J. Ho, K. Lee, M. Yang, D. Kriegman, Visual tracking using learned linear subspace, in: IEEE Conference on Computer Vision and Pattern Recognition, 2004, pp. 782–789.
[31] T. Bai, Y. Li, Robust visual tracking with structured sparse representation appearance model, Pattern Recognition 45 (2012) 2390–2404.
[32] M. Kristan, S. Kovacic, A. Leonardis, J. Pers, A two-stage dynamic model for visual tracking, IEEE Transactions on System, Man and Cybernetics – Part B: Cybernetics 40 (6) (2010) 1505–1520.
[33] J. Kwon, K. Lee, Visual tracking decomposition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 1269–1276.
[34] M. Everingham, L. Van Gool, C. Williams, J. Winn, A. Zisserman, The pascal visual object classes (VOC) challenge, International Journal of Computer Vision 88 (2) (2010) 303–338.

Kaihua Zhang received his B.S. degree in Technology and Science of Electronic Information from Ocean University of China in 2006 and Master degree in Signal and Information Processing from the University of Science and Technology of China (USTC) in 2009. Currently he is a PhD candidate in the Department of Computing, The Hong Kong Polytechnic University. His research interests include segment by level set method and visual tracking by detection.

Huihui Song received her B.S. degree in Technology and Science of Electronic Information from Ocean University of China in 2008 and Master degree in Communication and Information System. Currently she is a PhD candidate in the Department of Geography and Resource Management, Chinese University of Hong Kong. Her research interests include remote sensing image processing.