Robust Object Tracking via Active Feature Selection

Kaihua Zhang, Lei Zhang, *Member, IEEE,* Ming-Hsuan Yang, *Senior Member, IEEE,* and Qinghua Hu, *Member, IEEE*

***Abstract*—Adaptive tracking by detection has been widely studied with promising results. The key idea of such trackers is how to train an online discriminative classifier, which can well separate an object from its local background. The classifier is incrementally updated using positive and negative samples extracted from the current frame around the detected object location. However, if the detection is less accurate, the samples are likely to be less accurately extracted, thereby leading to visual drift. Recently, the multiple instance learning (MIL) based tracker has been proposed to solve these problems to some degree. It puts samples into the positive and negative bags, and then selects some features with an online boosting method via maximizing the bag likelihood function. Finally, the selected features are combined for classification. However, in MIL tracker the features are selected by a likelihood function, which can be less informative to tell the target from complex background. Motivated by the active learning method, in this paper we propose an active feature selection approach that is able to select more informative features than the MIL tracker by using the Fisher information criterion to measure the uncertainty of the classification model. More specifically, we propose an online boosting feature selection approach via optimizing the Fisher information criterion, which can yield more robust and efficient real-time object tracking performance. Experimental evaluations on challenging sequences demonstrate the efficiency, accuracy, and robustness of the proposed tracker in comparison with state-of-the-art trackers.**

***Index Terms*—Active learning, fisher information, multiple instance learning, visual tracking.**

I. Introduction

VISUAL tracking is a very active research topic in the field of computer vision because of its importance in many applications, such as vehicle navigation, traffic monitoring, and human–computer interaction [1]. Although object tracking has been studied for several decades and numerous algorithms have been proposed, it is still a very challenging problem since the appearance of the target object can be drastically changed due to the factors such as illumination changes, pose variations, full or partial occlusions, abrupt motion, etc. Thus, how to design a robust appearance model that can adaptively handle the above factors over time is the key to develop a high-performance tracking system.

Some appearance models are only designed to represent the object, while the other models consider both the object and its local background. The latter methods often perform better than the former ones because they often treat tracking as a binary classification problem, which separates object from its local background via a discriminative classifier. Considering that these methods are closely related to the object detection task, they are often referred to as tracking by detection. When training the classifier, the selection of positive and negative samples affects the performance of the tracker. Most trackers only choose one positive sample, i.e., the tracking result in the current frame. If the tracked target location is not accurate, the classifier will be updated based on a less effective positive sample, thereby leading to visual drift over time. To alleviate the drifting problem, multiple samples near the tracked target location can be used to train the classifier. However, the ambiguity occurs if the traditional supervised learning method is used to train the classifier [2].

Recently, a multiple instance learning (MIL) approach [2] was proposed to solve the ambiguity problem in tracking. The samples are put into bags and only the labels of the bags are provided. The bag is positive if one or more instances in it are positive while the bag is negative when all of the instances in it are negative. The samples near the tracking location are put into the positive bag while the samples far from the tracking location are put into the negative bag. Then, a classifier is designed by optimizing the bag likelihood function. To handle the appearance variations over time, an online MIL boosting algorithm is proposed to greedily select the discriminative features from a feature pool by maximizing the bag likelihood function. Finally, the selected weak classifiers (each corresponds to a feature) are linearly combined to a strong classifier. The strong classifier is then used to separate object from background in the next frame. Empirical studies on some challenging sequences have shown that the MIL tracker can better handle visual drift than most state-of-the-art trackers [2].

Despite its success, the MIL tracker [2] has the following shortcomings. First, the selected features may be less informative. In order to make the classifier discriminative enough, a relatively large number of features are selected from the

Manuscript received December 9, 2012; revised April 19, 2013; accepted April 28, 2013. Date of publication June 18, 2013; date of current version November 1, 2013. This work was supported in part by the HKPU Internal Research Grant, the National Natural Science Foundation of China under Grant 61222210, NSF CAREER Grant 1149783, and NSF IIS Grant 1152576. This paper was recommended by Associate Editor Francesco G. B. De Natale.

K. Zhang and L. Zhang are with the Department of Computing, Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, (e-mail: cskhzhang@comp.polyu.edu.hk; cslzhang@comp.polyu.edu.hk).

M.-H. Yang is with the Department of Electrical Engineering and Computer Science, University of California, Merced, CA 95344 USA (e-mail: mhyang@ucmerced.edu).

Q. Hu is with the School of Computer Science and Technology, Tianjin University, Tianjin 300072, China (e-mail: huqinghua@tju.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2013.2269772

1051-8215 © 2013 IEEE

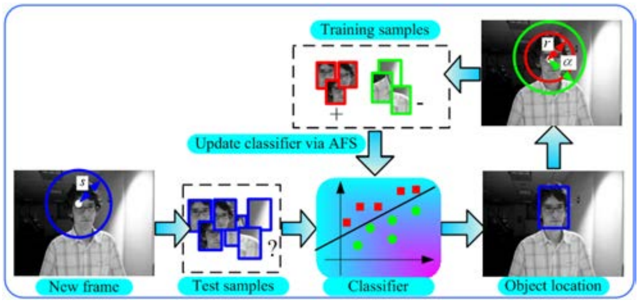

Fig. 1. Illustration of how our tracking system works.

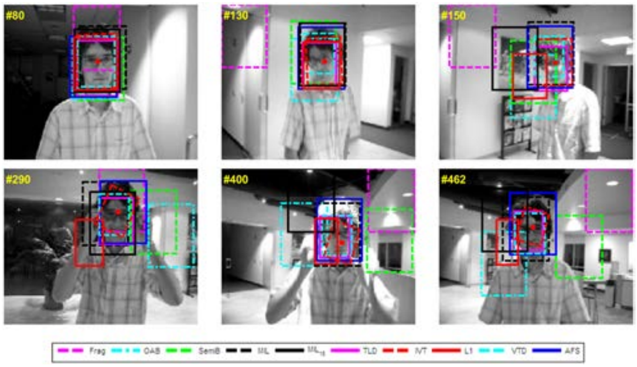

Fig. 2. Some sampled tracking results of the *David indoor* sequence.

feature pool. This enlarges the computational burden. Second, the more features are selected, the higher the probability that less discriminative features are included. These less discriminative features can degrade the performance of the classifier, and cause drift over time.

To address the above problems, inspired by the active learning method [3], we propose a novel feature selection scheme to select the more informative features for visual tracking, namely, the active feature selection (AFS)-based tracker. An online feature selection scheme is proposed by optimizing a bag Fisher information function instead of the bag likelihood function. Thus, the selected features are much more informative than those selected by the bag likelihood function in MIL tracker [2]. Consequently, we can use less features to design a classifier, which is more efficient and robust than the classifier induced by the MIL tracker. Our experimental evaluations on challenging video clips validate the superior performance of AFS tracker to state-of-the-art trackers in terms of efficiency, accuracy, and robustness.

The rest of this paper is organized as follows. Some related work is reviewed in Section II. In Section III, we introduce our tracking algorithm in detail. Section IV compares our tracker with state-of-the-art trackers. Finally, Section V concludes this paper.

II. Related Work

Visual tracking has been extensively studied, and a good review can be found in [1]. The recent algorithms can be mainly categorized into two classes according to how they deal with the appearance variations of target object and the background: the generative methods [4]–[12] and the discriminative methods [2], [13]–[21]. The generative methods learn an appearance model for the target object by minimizing the difference between the search region and the reference object model. Black *et al.* [4] represented the object by learning a subspace model offline. To handle appearance variations of the object over time, some online appearance update models have been proposed. Jepson *et al.* [5] proposed a Gaussian mixture model, which is updated by an online expectation maximization (EM) algorithm. Ho *et al.* [6] and Ross *et al.* [7] used the incremental subspace update schemes to adapt the appearance variation. Adam *et al.* [8] proposed a fragment-based appearance model to deal with the pose variation and partial occlusion. Recently, sparse representation methods have been proposed to handle the partial occlusion in visual tracking [9]. Kwon *et al.* [10] decomposed the observation model into multiple basic observation models, which cover different kinds of features and motions to handle pose variations, illuminations and scale changes. Sun *et al.* [11] proposed an object appearance model, which combines the local scale-invariant feature and the global incremental principle component analysis (PCA).

The discriminative methods treat tracking as a binary classification problem by training a discriminative classifier to separate object from background. Avidan [13] trained an offline support vector machine (SVM) and combined it into an optic-flow based tracker. To adapt the appearance changes of the object and background over time, Avidan [14] proposed an online boosting method to train the classifier: some weak classifiers are updated in an online manner and then ensembled into a strong classifier. Collins *et al.* [15] proposed an online feature selection scheme to evaluate the multiple features and integrated this scheme into a mean-shift tracking system [12] to select the most discriminative features. In [16], the relationship between the object and the structured environments is exploited to improve the performance of tracking. Grabner *et al.* [17] developed an online boosting feature selection technique, which demonstrates good performance to adaptively handle appearance changes. To better handle visual drift, Grabner *et al.* [18] proposed an online semi-supervised tracker, which only labels the samples in the first frame while leaving the samples in the sequent frames unlabeled. Babenko *et al.* [2] proposed to use an online MIL approach to handling the ambiguity in tracking location to reduce visual drift. Kalal *et al.* [19] proposed a semi-supervised learning approach to select the positive and negative samples via an online classifier with structural constraints. Recently, an efficient tracking algorithm [21] based on compressive sensing theory [22] was proposed, which demonstrates that the low dimensional features randomly extracted from the high dimensional multiscale image feature space can preserve the discriminative capability, thereby facilitating object tracking.

III. Tracking with Adaptive Feature Selection

A. System Overview

Fig. 1 illustrates the basic flow of our tracking system. There are two important components in our tracking system. One is how to detect the object location in the new frame, and the

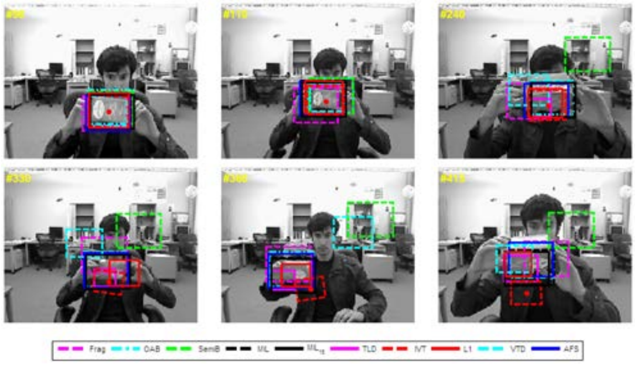

Fig. 3. Some sampled tracking results of the *Twinings* sequence.

other is how to update the classifier. We represent the object location in the t-th frame as $\boldsymbol{l}_t^*$. A set of patches near the old object location are cropped as $D^s = \{\boldsymbol{x}|\,|\boldsymbol{l}(\boldsymbol{x}) - \boldsymbol{l}_{t-1}^*| \; < s\}$, where s is a search radius and x denotes the image patch. Then, we compute the classifier response $H(\text{x})$ for all $\boldsymbol{x}{\in}D^s$, where the classifier $H(\boldsymbol{x}) = \sum_k h_k(\boldsymbol{x})$ is a linear combination of some weak classifiers $hk(\text{x})$. Finally, we update the object location using a greedy strategy

$$\boldsymbol{l}_t^* = \boldsymbol{l}(\arg\max_{\boldsymbol{x}\in D^s} H(\boldsymbol{x})). \qquad (1)$$

After the object location is updated, a set of samples $D^r = \{x||\boldsymbol{l}(\boldsymbol{x}) - \boldsymbol{l}_t^*| \; < \; r\}$, where r is a scalar radius, are cropped and put into a positive bag. For the negative samples, we take a small random set of samples from set $D^{r,\beta} = \{x|r < |\boldsymbol{l}(\boldsymbol{x}) - \boldsymbol{l}_t^*| < \beta\}$, where β is a scalar radius, because $D^{r,\beta}$ contains a large number of samples. If the background between two consecutive frames do not changes much, the negative patches, which are not from the boundary area around the target, may be beneficial for classification because they will much correlate with each other. However, if the background changes significantly, such negative patches may have a side effect on classification because they will be less correlated. To make a compromise, we only consider the negative patches near the target. We put all the negative samples into a negative bag, and update the classifier via maximizing the bag Fisher information loss function in an online manner.

B. MIL Tracker

We first briefly review the MIL tracker [2], which is most related to our work. The MIL method was introduced by Dietterich *et al.* [23] to deal with the drug activity prediction. Suppose we have a set of N bags $\{X_1, \ldots, X_N\}$, where each bag $X_i = \{\boldsymbol{x}_{i1}, \ldots, \boldsymbol{x}_{in_i}\}$ has n_i instances. Let $y_i \in \{0,1\}$ be the label of bag X_i and $y_{ij} \in \{0, 1\}$ the label of instance $\boldsymbol{x}_{ij}$. The MIL defines that if bag X_i is positive, then at least one of the instance labels in it is positive. If the bag label is zero, then all of the corresponding instance labels are zero. The MIL tracker seeks for the discriminative classifier $H(\boldsymbol{x})$, which can return the conditional probability $p(y = 1|\boldsymbol{x})$. Since the discriminative classifier is an instance classifier that is related to the conditional probabilities of the instances, the Noisy-OR model is used to exploit the conditional probabilities of the instances to estimate the bag probability

$$p(y_i = 1|X_i) = 1 - \prod_j (1 - p(y_{ij} = 1|\boldsymbol{x}_{ij})). \qquad (2)$$

where the instance probability $p(y_{ij} = 1|\boldsymbol{x}_{ij})$ is modeled as

$$p(y_{ij} = 1|\boldsymbol{x}_{ij}) = \sigma(H(\boldsymbol{x}_{ij})) \qquad (3)$$

where $\sigma(z) = \frac{1}{1+e^{-z}}$ is the sigmoid function, and the classifier $H(\boldsymbol{x})$ is learned by maximizing the following bag log likelihood loss function

$$\mathcal{L} = \sum_i (y_i \log(p(y_i = 1|X_i)) + (1 - y_i) \log(1 - p(y_i = 1|X_i))). \qquad (4)$$

To handle the appearance changes over time, an online MIL boosting approach is proposed to update the classifier $H(\boldsymbol{x})$. First, a weak classifier pool is maintained, and then a small number of weak classifiers are greedily selected from the pool by maximizing the log likelihood of the bag

$$h_k = \arg\max_{h\in\Phi} \mathcal{L}(H_{k-1} + h) \qquad (5)$$

where $H_{k-1} = \sum_{m=1}^{k-1} h_m$ is a strong classifier by assembling the first k - 1 weak classifiers, and $\Phi = \{h_1, \ldots, h_M\}$is the weak classifier pool with M candidate weak classifiers. Similar to the boosting feature selection method in face detection [24], weak classifier selection can be viewed as feature selection because each weak classifier corresponds to a feature. Feature selection has proved to be very useful forreducing visual drift in visual tracking [15]. Moreover, the classifier can run efficiently because the number of the selected features is much smaller than the size of the feature pool.

C. Principle of AFS

From the formulation of the log likelihood function in (4), we can see that the feature selection scheme in (5) is to select the weak classifiers that maximize the conditional probability $p(y_i = 1|X_i)$ of the positive bag X_i while minimizing the conditional probability $p(y_j = 1|X_j)$ of the negative bag X_j. We argue that the selected features can be less informative than those selected by optimizing the Fisher information function in our method to be introduced below. Therefore, to ensure the enough discriminative information, in the MIL tracker [2], a relatively large number of features ($K = 50$) are selected from a feature pool with a relatively large size ($M = 250$), while in our AFS tracker only $K = 15$ features are selected from a pool with $M = 50$ features. Moreover, if too many features are selected, the discrimination between the object and background features can be reduced.

Similar to the MIL tracker [2], we take the classifier as the following form

$$H(\boldsymbol{x}) = \boldsymbol{\alpha}^T \boldsymbol{h}(\boldsymbol{x}) \qquad (6)$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m)^T$ is a weight vector and $\boldsymbol{h} = (h_1, \ldots, h_m)^T$ is a weak classifier vector. Each element in $\boldsymbol{h}$ is a decision stump function that returns the binary labels (i.e., +1 or -1). In order to devise the classifier $H(\boldsymbol{x})$, we need to estimate its corresponding parameters $\boldsymbol{\alpha}$. The Cramer–Rao

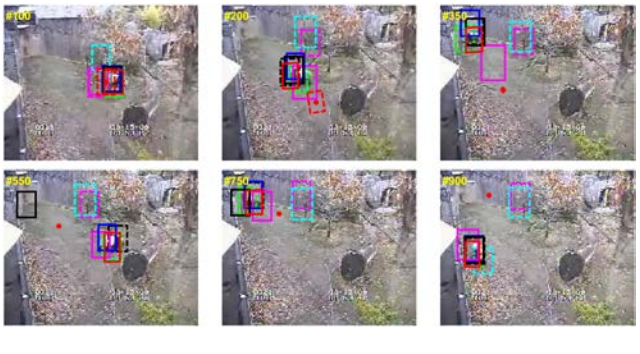

Fig. 4. Some sampled tracking results of the *Panda* sequence.

inequality [25] shows that for any unbiased estimator $\boldsymbol{t}_n$ of $\boldsymbol{\alpha}$ based on n independent and identically distributed samples from the probability $p(y|\boldsymbol{\alpha})$, the covariance of $\boldsymbol{t}_n$ should satisfy that $\mathrm{cov}(\boldsymbol{t}_n) - \frac{1}{n}I(\boldsymbol{\alpha})^{-1}$ is a nonnegative definite matrix, where $I(\boldsymbol{\alpha})$ is the Fisher information matrix [25] defined as

$$I(\boldsymbol{\alpha}) = -\int p(y|\boldsymbol{\alpha})\frac{\partial^2}{\partial\boldsymbol{\alpha}^2}\log p(y|\boldsymbol{\alpha})dy. \tag{7}$$

The Fisher information matrix represents the overall uncertainty of the classification model, which is often used in active learning method [26]. In [26], for each query in active learning, an unlabeled sample that can decrease the Fisher information most is selected. To measure the uncertainty of the classification model in our AFS tracker, we use the Fisher information matrix based on the samples from the bag probability

$$I(\boldsymbol{\alpha}) = \sum\nolimits_i \left[\begin{array}{l} y_i p(y_i|X_i,\boldsymbol{\alpha})\frac{\partial^2}{\partial\boldsymbol{\alpha}^2}\log p(y_i|X_i,\boldsymbol{\alpha}) \\ +(1-y_i)p(y_i|X_i,\boldsymbol{\alpha})\frac{\partial^2}{\partial\boldsymbol{\alpha}^2}\log p(y_i|X_i,\boldsymbol{\alpha}) \end{array}\right] + \delta I_m \tag{8}$$

where $y_i \in\{0,1\}$ is the bag label and δI_m (where $\delta > 0$ is a scalar parameter and I_m is an identity matrix) is added to make $I(\boldsymbol{\alpha})$ nonsingular. Note that δI_m is a trivial term. which is unrelated to the weak classifiers. Therefore, how to set δI_m does not affect the feature selection procedure. In (8), $p(y_i = 1|X_i, \boldsymbol{\alpha})$ and $p(y_i = 0|X_i, \boldsymbol{\alpha})$ are expressed as follows by combining (2), (3) and (6):

$$\begin{array}{l} p(y_i = 1|X_i,\boldsymbol{\alpha}) = 1 - \prod_j (1-\sigma(\boldsymbol{\alpha}^T\boldsymbol{h}(\boldsymbol{x}_{ij}))) \\ p(y_i = 0|X_i,\boldsymbol{\alpha}) = \prod_j (1-\sigma(\boldsymbol{\alpha}^T\boldsymbol{h}(\boldsymbol{x}_{ij}))). \end{array} \tag{9}$$

Note that our information matrix (8) is different from the objective functions of the recently developed multiple-instance active learning (MIAL) methods [27] and [28] because our objective is to measure the uncertainty of the classification model for the selected features when the bag labels are known, while the objective of MIAL is to measure the uncertainty of the classification model for an unlabeled sample.

The inverse Fisher information matrix $I(\boldsymbol{\alpha})^{-1}$ is the lower bound of the covariance matrix of the estimated $\boldsymbol{\alpha}$ [25]. As a particular case, $\det(I(\boldsymbol{\alpha}))^{-1}$ is the lower bound of the product of the variances for the elements in $\boldsymbol{\alpha}$. Thus, Liao *et al.* [29] proposed to select the samples that maximize $\det(I(\boldsymbol{\alpha}))$ for active learning to reduce the uncertainty of $\boldsymbol{\alpha}$. However, since it is difficult to compute $\det(I(\boldsymbol{\alpha}))$ in our objective function (8), we relax it to minimizing the trace of matrix $I(\boldsymbol{\alpha})$ (denoted by $\mathrm{tr}(I(\boldsymbol{\alpha}))$) because the upper bound of $\det(I(\boldsymbol{\alpha}))$ is $\left(\frac{1}{m}\mathrm{tr}(I(\boldsymbol{\alpha}))\right)^m$. It is easy to validate that $\det(I(\boldsymbol{\alpha})) \le \left(\frac{1}{m}\mathrm{tr}(I(\boldsymbol{\alpha}))\right)^m$ as follows. Since $I(\boldsymbol{\alpha})$ is a positive definite symmetric matrix [25], all of its eigenvalues $\{\lambda_i > 0, i = 1, ..., m\}$ are positive [30]. Thus, we have the following inequality [30]:

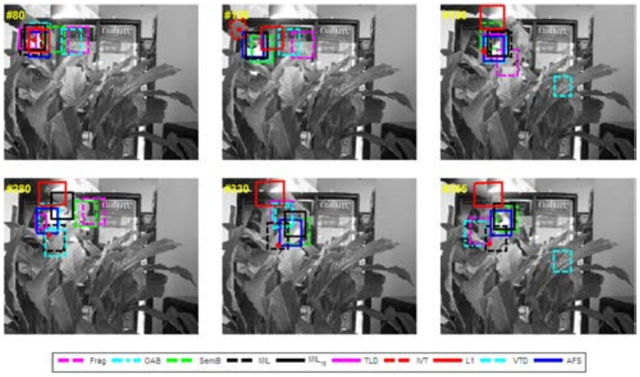

Fig. 5. Some sampled tracking results of the *Tiger 2* sequence.

$$\det(I(\boldsymbol{\alpha})) = \prod\nolimits_{i=1}^m \lambda_i \le \left(\frac{1}{m}\sum\nolimits_{i=1}^m \lambda_i\right)^m = \left(\frac{1}{m}\mathrm{tr}(I(\boldsymbol{\alpha}))\right)^m \tag{10}$$

where $\mathrm{tr}(I(\boldsymbol{\alpha}))$ is represented by

$$\begin{array}{l} \mathrm{tr}(I(\boldsymbol{\alpha})) \\ = -m\sum\nolimits_i \left[\begin{array}{l} y_i \left(\begin{array}{l} p(y_i|X_i,\boldsymbol{\alpha})\sum_j p(y_{ij}|\boldsymbol{x}_{ij},\boldsymbol{\alpha})(1-p(y_{ij}|\boldsymbol{x}_{ij},\boldsymbol{\alpha})) \\ +\sum_j p(y_{ij}|\boldsymbol{x}_{ij},\boldsymbol{\alpha})\left(\frac{p(y_{ij}|\boldsymbol{x}_{ij},\boldsymbol{\alpha})}{p(y_i|X_i,\boldsymbol{\alpha})} - 1\right) \end{array}\right) + \\ (1-y_i)p(y_i|X_i,\boldsymbol{\alpha})\sum_j p(y_{ij}|\boldsymbol{x}_{ij},\boldsymbol{\alpha})(1-p(y_{ij}|\boldsymbol{x}_{ij},\boldsymbol{\alpha})) \end{array}\right]. \\ +m\delta \end{array} \tag{11}$$

In (11), we have set $\boldsymbol{h}(\boldsymbol{x})^T\boldsymbol{h}(\boldsymbol{x}) = m$ because each element $h_i \in \{+1, -1\}$ in $\boldsymbol{h}(\boldsymbol{x})$ is a decision stump function. Please refer to **Appendix A** for the deviation of (11).

Although (11) seems complex, its physical meaning is simple. For the positive bag, as learning proceeds and the bag probability approaches to the target, we have $p(y_i = 1|X_i, \boldsymbol{\alpha}) \approx 1$ [31]. Thus, the component of the positive bag in $\mathrm{tr}(I(\boldsymbol{\alpha}))$ can be simplified to $-m(p(y_i = 1|X_i, \boldsymbol{\alpha}) - 1)\sum_j \left[\begin{array}{l} p(y_{ij} = 1|\boldsymbol{x}_{ij},\boldsymbol{\alpha}) \\ (1-p(y_{ij} = 1|\boldsymbol{x}_{ij},\boldsymbol{\alpha})) \end{array}\right]$. In order to minimize this function, we need to maximize two terms $p(y_i = 1|X_i, \boldsymbol{\alpha})$ and$p(y_{ij} = 1|\boldsymbol{x}_{ij}, \boldsymbol{\alpha})(1 - p(y_{ij} = 1|\boldsymbol{x}_{ij}, \boldsymbol{\alpha}))$. Similar to the bag log likelihood function (4), the first term is to maximize the conditional probability of the positive bag. The second term reaches its maximum value at $p(y_{ij} = 1|\boldsymbol{x}_{ij}, \boldsymbol{\alpha}) = 0.5$, which measures the most classification uncertainty for instance $\boldsymbol{x}_{ij}$. The component of the negative bag in $\mathrm{tr}(I(\boldsymbol{\alpha}))$ also contains two parts: $p(y_{ij} = 0|\boldsymbol{x}_{ij}, \boldsymbol{\alpha})(1 - p(y_{ij} = 0|\boldsymbol{x}_{ij}, \boldsymbol{\alpha}))$and $p(y_i = 0|X_i, \boldsymbol{\alpha})$. The analysis for these two components is the same as those for the positive bag. Therefore, minimizing $\mathrm{tr}(I(\boldsymbol{\alpha}))$ can be deemed as a tradeoff between the bag probability and the classification uncertainty for the instances. In the following, we propose an online AFS approach to selecting the informative features via minimizing$\mathrm{tr}(I(\boldsymbol{\alpha}))$.

Algorithm 1 Online AFS Boosting

```
Input: Dataset $\{X_i, y_i\}_{i=0}^{N}$, where $X_i = \{\boldsymbol{x}_{i1}, \boldsymbol{x}_{i2}, ...\}$ is the $i$-th bag and $y_i \in \{0, 1\}$.
1. Update all the $M$ weak classifiers in the pool with data $\{\boldsymbol{x}_{ij}, y_i\}$.
2. Initialize $H_0(\boldsymbol{x}_{ij}) = 0$ for all $i, j$
3. For $k = 1$ to $K$ do
4.   for $m = 1$ to $M$ do
5.     $\mathcal{F}_m = \mathcal{F}(H_{k-1} + h_m)$
6.   end for
7.   $m^* = \arg\min_m(\mathcal{F}_m)$
8.   $h_k \leftarrow h_{m^*}$

9.   $H_k \leftarrow H_{k-1} + h_k$

10. End for

Output: Classifier $H(\boldsymbol{x}) = \sum_k h_k(\boldsymbol{x})$.
```

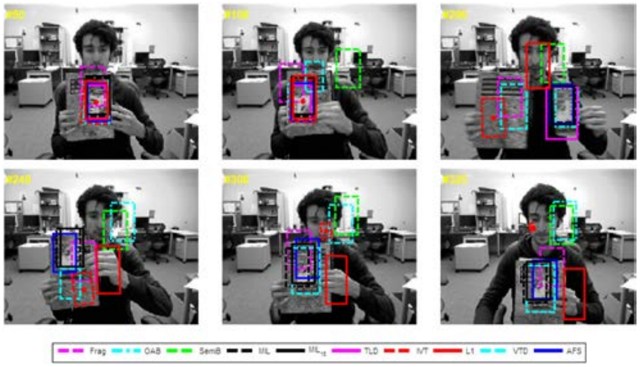

Fig. 6. Some sampled tracking results on the *Cliff bar* sequence.

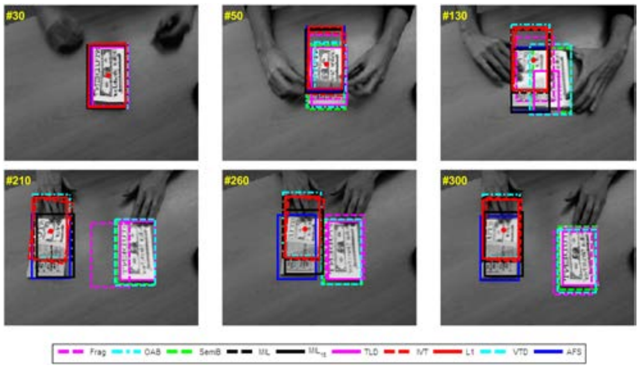

Fig. 7. Some sampled tracking results of the *Coupon book* sequence.

D. Online AFS Boosting

We take a statistical view of boosting [32] where the weak classifiers (each weak classifier corresponds to a feature) are selected sequentially to optimize a specific objective function $\mathcal{F}$ as

$$(h_k, \alpha_k) = \arg \min_{h \in \Phi, \alpha \in \mathbb{R}} \mathcal{F}(H_{k-1} + \alpha h) \qquad (12)$$

where $H_{k-1} = \sum_{i=1}^{k-1} h_i$ is a strong classifier with the first k - 1 weak classifiers and Φ is the set of all possible weak classifiers. For online learning, we always maintain a pool of M candidate weak classifiers. When updating the strong classifier, we first incrementally update the weak classifiers in the pool with the newly cropped samples, and then select sequentially $K<M$ the most discriminative weak classifiers from the pool by minimizing the Fisher information criterion

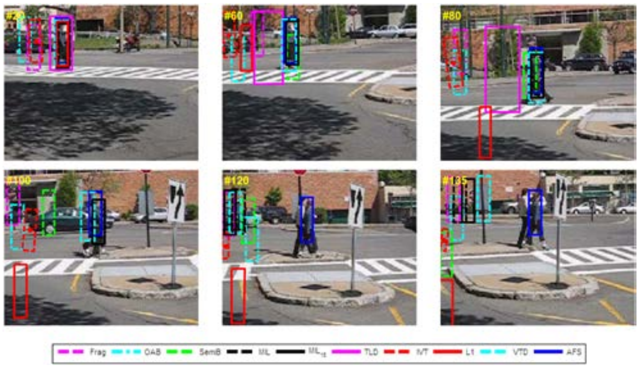

Fig. 8. Some sampled tracking results of the *Pedestrian* sequence.

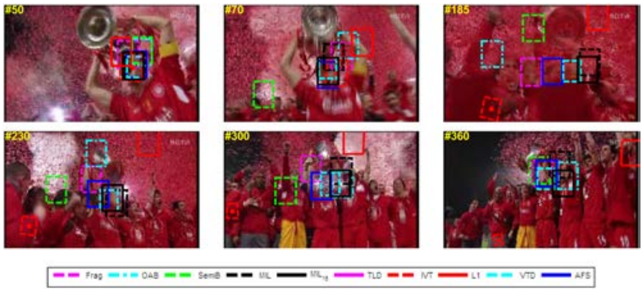

Fig. 9. Some sampled tracking results of the *Soccer* sequence.

TABLE I

AVERAGE FRAMES PER SECOND (FPS) OF AFS AND OTHER STATE-OF-THE-ART TRACKERS

Tracker	Frag	OAB	MIL	MIL$_{15}$	SemiB	IVT	L1	VTD	**AFS**
Average FPS	3	8	10	25	6	11	0.1	0.01	15/35[1]

$$(h_k, \alpha_k) = \arg \min_{h \in \{h_1, \ldots, h_M\}, \alpha \in \mathbb{R}} \mathcal{F}(H_{k-1} + \alpha h) \qquad (13)$$

where $\mathcal{F}(H_{k-1} + \alpha h) = \mathcal{F}(\boldsymbol{\alpha}^T \boldsymbol{h}) = \mathrm{tr}(I(\boldsymbol{\alpha}))$ with $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{k-1}, \alpha)^T$ and $\boldsymbol{h} = (h_1, \ldots, h_{k-1}, h)^T$. To simplify the problem, as in the MIL tracker [2], in our implementation, we integrated the scalar weights $\boldsymbol{\alpha}$ into the weak classifiers $\boldsymbol{h}$ in order to return real values. Therefore, the weight vector $\boldsymbol{\alpha}$ cannot be used to indicate the importance of the weak classifiers. Note that our feature selection criterion (13) is a greedy forward feature selection method. Though this greedy feature selection method is suboptimal, it is very efficient for visual tracking.

Algorithm 1 shows the pseudo-code of online AFS boosting, which is the key part of the tracking algorithm illustrated in Fig. 1.

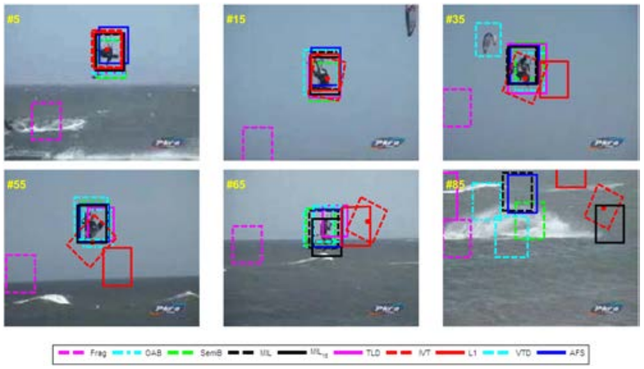

Fig. 10. Some sampled tracking results of the *Kitesurf* sequence.

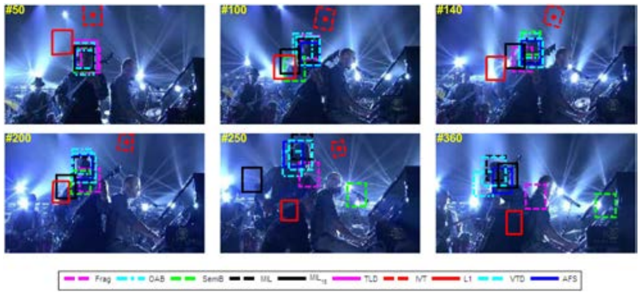

Fig. 11. Some sampled tracking results of the *Shaking* sequence.

E. Advantages Over the MIL Tracker

Our Fisher information criterion (13) can select the features that are much more informative than those selected from the log likelihood criterion (5) in the MIL tracker [2]; because our criterion maximizes the uncertainty of the selected features. Thus, we only need to actively select a small number of weak classifiers, which are more discriminative than those used in the MIL tracker. In our experiments, we select $K = 15$ weak classifiers from a pool with $M = 50$ candidate weak classifiers, which are much less than the MIL tracker where $K = 50$ and $M = 250$. Although our objective function (11) seems more complex than that used in MIL tracker (i.e., (4)), their computational complexities are comparative because only addition and multiplication are needed to compute bag and instance probabilities. Moreover, the MIL tracker needs to update more classifiers ($M = 250$) than ours ($M = 50$), and select more weak classifier ($K = 50$) than our method ($K = 15$). Thus, overall, our tracker is more efficient than MIL tracker (please refer to our experimental results in next section). In addition, because our selected weak classifiers are more informative than those selected by the MIL tracker, our appearance model (i.e., the strong classifier) is able to better handle visual drift.

F. Implementation Details

We use the same Haar-like image features as those used by the MIL tracker [2], which can be efficiently computed using the integral image technique [24]. Each feature f_i is a Haar-like image feature computed by the sum of weighted pixels in 2–4 randomly selected rectangles. Each weak classifier h_i returns the log odds ratio

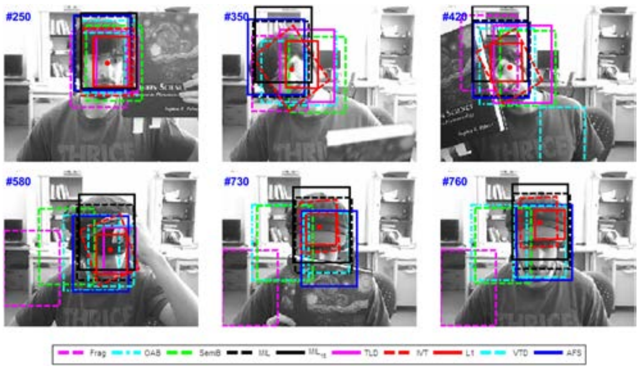

Fig. 12. Some sampled tracking results of the *Occluded face* sequence.

$$h_i = \log\left[\frac{p(y=1|f_i(\boldsymbol{x}))}{p(y=0|f_i(\boldsymbol{x}))}\right] = \log\left[\frac{p(f_i(\boldsymbol{x})|y=1)}{p(f_i(\boldsymbol{x})|y=0)}\right] \tag{14}$$

where we assume uniform prior $p(y = 1) = p(y = 0)$, $p(f_i(\boldsymbol{x})|y = 0) \sim N(\mu_0, \sigma_0)$, and $p(f_i(\boldsymbol{x})|y = 1) \sim N(\mu_1, \sigma_1)$. The parameters $\mu_t, \sigma_t, t \in \{0, 1\}$ can be incrementally updated based on maximal likelihood estimation [33]

$$\begin{cases} \mu_t \leftarrow \gamma\mu_t + (1-\gamma)\mu \\ \sigma_t \leftarrow \sqrt{\gamma\sigma_t^2 + (1-\gamma)\sigma^2 + \gamma(1-\gamma)(\mu_t-\mu)^2} \end{cases} \tag{15}$$

where $\{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_n, y_n)\}$ are the new data, $0 < \gamma < 1$ is a learning parameter, $\mu = \frac{1}{n}\sum_{k|y_i=t} f_i(\boldsymbol{x}_k)$, and $\sigma = \sqrt{\frac{1}{n}\sum_{k|y_i=t}(f_i(\boldsymbol{x}_k)-\mu)^2}$.

IV. Experimental Results

As the proposed AFS tracker is developed to address several issues of MIL-based tracking method (Section I), we compare it with the MIL tracker [2] on 12 challenging video sequences (all are publicly available). The other compared trackers are fragment tracker (Frag) [8], online AdaBoost tracker (OAB) [17], Semisupervised boosting tracker (SemiB) [18], incremental visual tracker (IVT) [7], L1 tracker [9], and visual tracking decomposition (VTD) method [10]. The default setting for the MIL tracker is to select $K = 50$ weak classifiers from a pool with $M = 250$ candidate weak classifiers. We also test the MIL tracker with setting $K = 15$ and $M = 50$ (we call it MIL_{15}).

We fix the parameters of the proposed tracker for all the experiments to demonstrate its robustness and stability.For the other competing algorithms, we use the original source codes or binary codes provided in [7]–[10], [17], and [18] and tune their parameters for best performance. Since all the competing trackers (except for [8]) involve randomness, we repeat each experiment ten times and report the average results. Our tracker is implemented in MATLAB and runs at 15 frames per second on a Pentium Dual-Core 2.10 GHz CPU with 1.95 GB RAM. The videos used in the experiments can be found at http://youtu.be/3UobcBa-V1Q. Table I lists the speed of all trackers in terms of average frames per second (FPS). Note that the source code of the MIL tracker is written

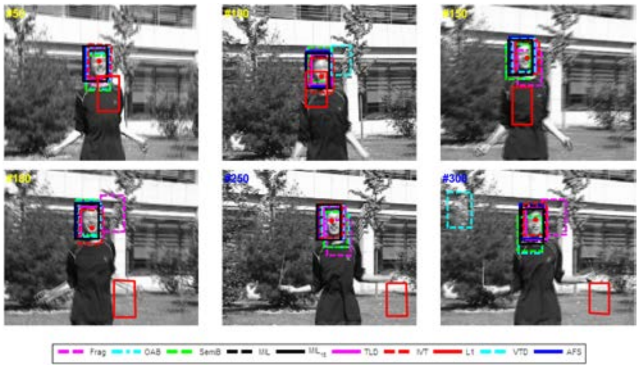

Fig. 13. Some sampled tracking results of the *Jumping* sequence.

TABLE II

SUCCESS RATE (%). BOLD FONTS INDICATE THE BEST PERFORMANCE WHILE THE *Italic* FONTS INDICATE THE SECOND BEST

Sequences	Frag	OAB	SemiB	MIL	MIL$_{15}$	TLD	IVT	L1	VTD	**AFS**
David indoor	8	32	45	66	52	*98*	**100**	41	83	92
Twinings	69	**97**	22	71	67	46	49	83	**97**	*92*
Panda	7	69	67	*75*	47	29	7	56	4	**76**
Tiger2	13	39	19	43	*44*	41	18	12	12	**56**
Cliff bar	22	24	63	67	*72*	67	46	38	47	**92**
Coupon book	27	98	37	*99*	68	16	**100**	**100**	37	**100**
Pedestrian	5	4	23	53	45	21	10	18	*64*	**80**
Kitesurf	1	73	73	78	*79*	45	30	27	41	**80**
Soccer	27	8	9	17	20	10	19	13	*39*	**41**
Shaking	28	40	31	*85*	17	16	1	10	**97**	79
Occluded face	52	46	41	*99*	57	46	87	84	67	**100**
Jumping	36	86	84	*99*	*99*	98	98	9	87	**100**

in C++, which runs at 10 FPS, while the MIL$_{15}$ tracker runs at 25 FPS. However, as shown in Section IV-B, the MIL$_{15}$ tracker performs poorly in most experiments. We also implemented our algorithm in C++ and it runs at 35 FPS without optimization, which is more than three times faster than the MIL tracker. The source codes of our AFS tracker can be found at http://www4.comp.polyu.edu.hk/∼cslzhang/code.htm

A. Experimental Setup

We set the radius $r = 4$ for cropping the samples in the positive bag, which generates 45 samples. The out radius for the set $D^{r,\beta}$ that generates negative samples is set to β = 35. Then, we randomly select 45 negative samples from $D^{r,\beta}$ to construct the negative bag. The radius for searching the new object location in the next frame is set to $s = 25$ and about 2000 samples are drawn, which is the same as that in the MIL tracker [2]. We tested different values of parameter s and found the tracking results are stable when we set $20<s<30$. Hence, in all our experiments, we set $s = 25$. Therefore, this procedure is time consuming if too many weak classifiers are used to design the strong classifier. Our tracker uses $K = 15$ weak classifiers and, thus, is much more efficient than the MIL tracker [2], which sets $K = 50$. Moreover, in AFS the number of candidate weak classifiers in the pool is set to $M = 50$, which is also less than that of the MIL tracker ($M = 250$). The learning parameter is set to γ=0.85.

B. Qualitative Evaluation

1) *Scale and Pose Changes:* Although our tracker only estimates the translational motion, which is similar to most state-of-the-art algorithms (Frag, OAB, SemiB and MIL), it can also handle scale and orientation changes because of the Haar-like features. In the *David indoor* sequence, the target has big scale and pose changes. Note that the IVT, MIL, VTD, and our AFS trackers perform well on this sequence while the Frag, OAB, SemiB, L1, and MIL$_{15}$ have severe drifts (see frames #130, #150, #290, #400 in Fig. 2). The Haar-like features make MIL and AFS trackers able to handle the scale and pose changes well. Nonetheless, our AFS tracker yields much more accurate results (see frames #290, #462 in Fig. 2) than the MIL tracker because it can select more informative features to better separate object from background. The MIL$_{15}$ tracker suffers from severe drift at frames #150, #400, and #462, which verifies that the selected features by the MIL tracker are less informative than those by our AFS tracker. In the *Twinings* sequence (Fig. 3), the target undergoes out-of-plane rotation. The Frag tracker has severe drift at frames #110, #240, #330, #360, and #415 because its template does not update online, making it unable to handle large appearance changes. The SemiB tracker completely drifts to the background at frames #240, #330, #360, and #415 because it throws away some very useful information that can well separate object from its background [2]. The VTD method also has severe drift at frames #240, #330, #360, and #415 because it does not use the information from the background. In the *Panda* sequence (Fig. 4), the target undergoes large scale nonrigid deformation. The Frag, IVT, and VTD methods drift to the background (see frames #200, #350, #550, #750, #900) because they are not specially designed for nonrigid deformation. The MIL$_{15}$ tracker drifts to the background at frames #550 and #750 while the MIL and our AFS trackers perform well at these frames.

2) *Background Clutter and Pose Variation:* We use four sequences (*Tiger 2*, *Cliff bar*, *Coupon book*, and *Pedestrian*) to demonstrate the superior performance of our tracker in handling background clutter and pose variation. In the *Tiger 2* sequence, there are also partial occlusion and out-of-plane rotation, which make object tracking more difficult. From the tracking results shown in Fig. 5, we observe that all the other trackers drift to the background at some frames (see frames #280 and #330) expect for AFS tracker, which tracks the object stably and accurately. In the *Cliff bar* sequence, the background has similar texture to the target. Moreover, the target undergoes in-plane rotation. The Frag, OAB, SemiB, IVT, L1, and VTD methods drift to the background while the MIL, MIL$_{15}$, and our AFS trackers perform well on this sequence. The reason the Frag tracker cannot work well on this sequence is that its template does not update online, making it unable to adaptively capture the difference between the target and the background over time. The SemiB, L1, and VTD methods cannot work well on this sequence because they do not use the useful information from the background to discriminate object. Because of the same reason, in the *Coupon book* sequence shown by Fig. 7, the SemiB, LIT, and VTD methods also drift to another coupon book after the top coupon book is taken away (see frames #210, #260 and #300).

Our AFS and MIL trackers perform well on these two sequences due to the following reasons. First, the Haar-like

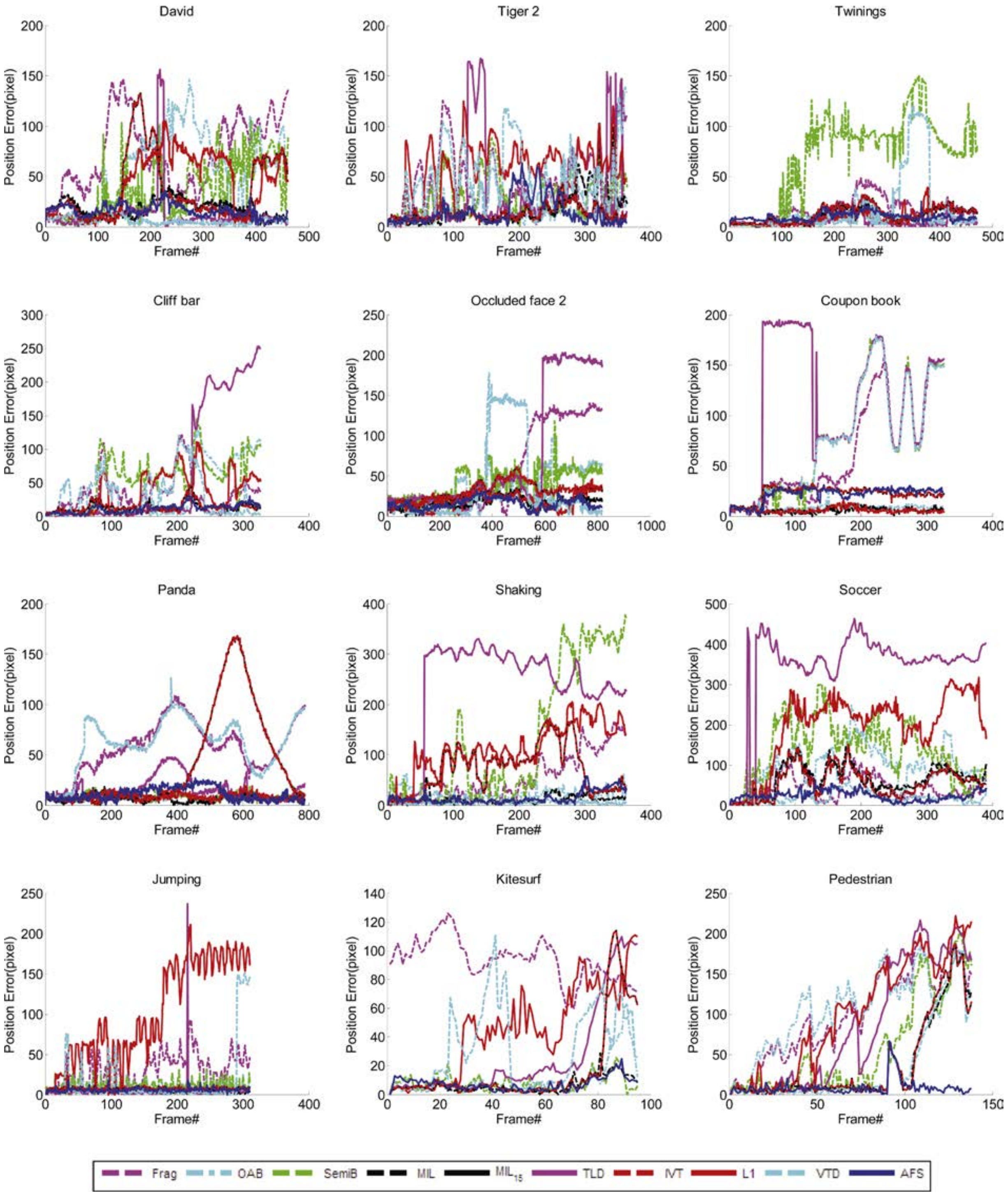

Fig. 14. Error plots of test sequences.

features are localized, which are effective in handling appearance changes due to pose variation; second, the discriminative appearance models are updated in an online manner, which take into account the difference between the target and the background over time and thereby avoid the drift problem throughout these two sequences. In the *Pedestrian* sequence, there is also camera motion. Most trackers drift to the background except for the MIL, MIL$_{15}$, and our AFS trackers from frame #1 to #100. The reason is that the localized Haar-like features are less sensitive to appearance changes caused by pose variation. Nonetheless, the MIL and MIL$_{15}$ trackers drift to the background in latter frames (see frame #120 and #135 in Fig. 8) while only our AFS tracker can perform well throughout the sequence.

3) *Illumination Change and Pose Variation:* We use the *Soccer*, *Kitesurf*, and *Shaking* sequences to evaluate the performance of AFS in handling illumination change and pose variation. In the *Soccer* sequence, there is also severe occlusion besides illumination change. Only our AFS tracker performs well throughout this sequence while the other trackers drift

from the target at some frames as shown in Fig. 9. There is also out-of-plane rotation in the *Kitesurf* sequence. As shown in Fig. 10, only AFS, SemiB, and MIL$_{15}$ trackers work well on this sequence while the other trackers drift to the background in the last frames. In the *Shaking* sequence shown in Fig. 11, the target undergoes large illumination and pose variations. All the trackers except for AFS, VTD, and MIL drift from the target quickly. The discriminative appearance model in AFS finds the most informative features to account for the appearance changes of the target and background over time, and therefore, it achieves favorably accurate and stable tracking results.

4) *Occlusion and Motion Blur:* Figs. 12 and 13 evaluate the AFS tracker when the targets undergo occlusion and motion blur. In the *Occluded face 2* sequence, there is pose variation besides partial occlusion. Although the Frag tracker is specially designed to handle partial occlusion by a part-based model, it cannot perform well on this sequence because of the large scale appearance changes due to the severe pose variation and occlusion. The OAB and SemiB trackers drift to the background when the heavy occlusion occurs at frame #730 in Fig. 12. After that frame, the OAB and SemiB trackers are unable to redetect the target. Although the IVT and L1 methods are able to track the object throughout the sequence, their results are inaccurate at frames #730 and #760, and both the two trackers are snapped to cap area. The reason is that they are generative models that do not take into account the useful information from the background. Both AFS and MIL trackers achieve good results because of the following two reasons. First, the localized Haar-like features are robust to partial occlusion [2]. Second, both trackers use an online update criterion that takes into account the appearance changes of the target and the background. In the *Jumping* sequence shown in Fig. 13, there is severe motion blur, which makes it difficult to distinguish the appearance of the target. Our tracker still performs well while the L1 method drifts to the background quickly. It can be explained by the fact that the global intensity features used in L1 method have limited discriminative capability to separate target from background when the appearance of the target changes much due to severe motion blur.

C. Quantitative Evaluation

We use two commonly used criteria to quantitatively assess the performance of the trackers: the tracking success rate and the center location error using the manually labeled ground truth. We employ the PASCAL [34] overlap criterion to determine whether a tracking result is a success. Given the ground truth bounding box ROI_g and the tracked bounding box ROI_t, the score is defined as $score = \frac{area(R_g \cap R_t)}{area(R_g \cup R_t)}$. If $score \geq 0.5$, the tracking result is considered as a success. Table II shows the success rates of competing methods. Our AFS tracker achieves the best or second best performance in all the test sequences. Fig. 14 illustrates the tracking results in terms of center location error, which is defined as the Euclidian distance between the center locations of the tracked target and the ground truth. Overall, our AFS tracker performs favorably against the other state-of-the-art trackers.

V. Conclusion

In this paper, we proposed a robust tracker based on an online discriminative appearance model. In order to design a robust appearance model, we developed an online active feature selection approach via minimizing a Fishier information criterion. We showed that the features selected by our proposed online AFS boosting algorithm were much more informative and discriminative than those selected by online MIL boosting algorithm, which maximized a likelihood loss function. The AFS appearance model can well handle large appearance changes. Numerous experimental results and evaluations on challenging video sequences demonstrated that our AFS tracker outperforms other state-of-the-art algorithms in terms of efficiency, accuracy, and robustness.

Appendix A
Deviation of (11)

In (11), the conditional probability of the instance $\boldsymbol{x}_{ij}$ is given by (3), which is a logistic regression function$p(y_{ij} = 1|\boldsymbol{x}_{ij}) = \sigma(\boldsymbol{\alpha}^T \boldsymbol{h}(\boldsymbol{x}_{ij}))$. Thus, we have

$$\frac{\partial}{\partial \boldsymbol{\alpha}} p(y_{ij} = 1|\boldsymbol{x}_{ij}) = \boldsymbol{h}(\boldsymbol{x}_{ij})(1 - p(y_{ij} = 1|\boldsymbol{x}_{ij}))p(y_{ij} = 1|\boldsymbol{x}_{ij}) \tag{A-1}$$

Next, we compute$\frac{\partial}{\partial \boldsymbol{\alpha}} p(y_i = 1|X_i)$. There is

$$\begin{aligned}
&\tfrac{\partial}{\partial \boldsymbol{\alpha}} p(y_i = 1|X_i) = \tfrac{\partial}{\partial \boldsymbol{\alpha}} \left(1 - \textstyle\prod_j (1 - p(y_{ij} = 1|\boldsymbol{x}_{ij}))\right) \\
&= -\tfrac{\partial}{\partial \boldsymbol{\alpha}} \textstyle\prod_j (1 - p(y_{ij} = 1|\boldsymbol{x}_{ij})) \\
&= -\left[\tfrac{\partial}{\partial \boldsymbol{\alpha}} \log \textstyle\prod_j (1 - p(y_{ij} = 1|\boldsymbol{x}_{ij}))\right] \textstyle\prod_j (1 - p(y_{ij} = 1|\boldsymbol{x}_{ij})) \\
&= -\left[\textstyle\sum_j \tfrac{\partial}{\partial \boldsymbol{\alpha}} \log(1 - p(y_{ij} = 1|\boldsymbol{x}_{ij}))\right] \textstyle\prod_j (1 - p(y_{ij} = 1|\boldsymbol{x}_{ij})) \\
&= \textstyle\prod_j (1 - p(y_{ij} = 1|\boldsymbol{x}_{ij})) \textstyle\sum_j \tfrac{1}{1-p(y_{ij}=1|\boldsymbol{x}_{ij})} \tfrac{\partial}{\partial \boldsymbol{\alpha}} p(y_{ij} = 1|\boldsymbol{x}_{ij})
\end{aligned}$$

Using (A-1), we have

$$\begin{aligned}
&\tfrac{\partial}{\partial \boldsymbol{\alpha}} p(y_i = 1|X_i) \\
&= \textstyle\sum_j \boldsymbol{h}(\boldsymbol{x}_{ij}) p(y_{ij} = 1|\boldsymbol{x}_{ij}) \textstyle\prod_j (1 - p(y_{ij} = 1|\boldsymbol{x}_{ij})) \\
&= \textstyle\sum_j \boldsymbol{h}(\boldsymbol{x}_{ij}) p(y_{ij} = 1|\boldsymbol{x}_{ij})(1 - p(y_i = 1|X_i))
\end{aligned} \tag{A-2}$$

Using (A-2), we have

$$\begin{aligned}
&\tfrac{\partial^2}{\partial \boldsymbol{\alpha}^2} p(y_i = 1|X_i) \\
&= \textstyle\sum_j \boldsymbol{h}(\boldsymbol{x}_{ij}) p(y_{ij} = 1|\boldsymbol{x}_{ij}) \begin{bmatrix} (1 - p(y_{ij} = 1|\boldsymbol{x}_{ij})) \\ (1 - p(y_i = 1|X_i))\boldsymbol{h}(\boldsymbol{x}_{ij})^T \end{bmatrix} \\
&- \textstyle\sum_j \boldsymbol{h}(\boldsymbol{x}_{ij}) p(y_{ij} = 1|\boldsymbol{x}_{ij})^2 (1 - p(y_i = 1|X_i))\boldsymbol{h}(\boldsymbol{x}_{ij})^T
\end{aligned} \tag{A-3}$$

$$\begin{aligned}
&\tfrac{\partial^2}{\partial \boldsymbol{\alpha}^2} \log p(y_i = 1|X_i, \boldsymbol{\alpha}) \\
&= \tfrac{\partial}{\partial \boldsymbol{\alpha}} \left(\tfrac{1}{p(y_i=1|X_i)}\right) \left(\tfrac{\partial}{\partial \boldsymbol{\alpha}} p(y_i = 1|X_i)\right)^T \\
&+ \tfrac{1}{p(y_i=1|X_i)} \tfrac{\partial^2}{\partial \boldsymbol{\alpha}^2} p(y_i = 1|X_i). \qquad \text{(A-4)} \\
&= -\tfrac{\partial}{\partial \boldsymbol{\alpha}} p(y_i = 1|X_i) \tfrac{1}{p(y_i=1|X_i)^2} \left(\tfrac{\partial}{\partial \boldsymbol{\alpha}} p(y_i = 1|X_i)\right)^T \\
&+ \tfrac{1}{p(y_i=1|X_i)} \tfrac{\partial^2}{\partial \boldsymbol{\alpha}^2} p(y_i = 1|X_i)
\end{aligned}$$

We then compute the components in (11), which are related to the positive and negative bags, respectively. For the positive bags, using (A-1)–(A-4), we have

$$\begin{aligned}
\mathrm{tr}^{+} &= \sum_i \mathrm{tr}\left(y_i p(y_i|X_i,\boldsymbol{\alpha})\tfrac{\partial^2}{\partial\boldsymbol{\alpha}^2}\log p(y_i|X_i,\boldsymbol{\alpha})\right)\\
&= \sum_i \left(\begin{array}{l}
-\frac{(1-p(y_i=1|X_i,\boldsymbol{\alpha}))^2}{p(y_i=1|X_i,\boldsymbol{\alpha})}\\
\sum_j p(y_{ij}=1|\boldsymbol{x}_{ij},\boldsymbol{\alpha})^2\boldsymbol{h}(\boldsymbol{x}_{ij})^T\boldsymbol{h}(\boldsymbol{x}_{ij})\\
+(1-p(y_i=1|X_i,\boldsymbol{\alpha}))\\
\sum_j \left(\begin{array}{l} p(y_{ij}=1|\boldsymbol{x}_{ij},\boldsymbol{\alpha})-\\ 2p(y_{ij}=1|\boldsymbol{x}_{ij},\boldsymbol{\alpha})^2\end{array}\right)\boldsymbol{h}(\boldsymbol{x}_{ij})^T\boldsymbol{h}(\boldsymbol{x}_{ij})
\end{array}\right)\\
&= \sum_i \left(\begin{array}{l}
p(y_i=1|X_i,\boldsymbol{\alpha})\sum_j\left[\begin{array}{l} p(y_{ij}=1|\boldsymbol{x}_{ij},\boldsymbol{\alpha})\\ \left(\begin{array}{l} p(y_{ij}=1|\boldsymbol{x}_{ij},\boldsymbol{\alpha})\\ -1\end{array}\right)\end{array}\right]\boldsymbol{h}(\boldsymbol{x}_{ij})^T\boldsymbol{h}(\boldsymbol{x}_{ij})\\
+\sum_j\left[\begin{array}{l} p(y_{ij}=1|\boldsymbol{x}_{ij},\boldsymbol{\alpha})\\ \left(1-\frac{1}{p(y_i=1|X_i,\boldsymbol{\alpha})}\right)\end{array}\right]\boldsymbol{h}(\boldsymbol{x}_{ij})^T\boldsymbol{h}(\boldsymbol{x}_{ij}).
\end{array}\right)
\end{aligned} \tag{A-5}$$

For the negative bags, we first have $p(y_i = 0|X_i) = 1 - p(y_i = 1|X_i)$, and then

$$\begin{aligned}
\mathrm{tr}^{-} &= \sum_i \mathrm{tr}\left(p(y_i=0|X_i,\boldsymbol{\alpha})\tfrac{\partial^2}{\partial\boldsymbol{\alpha}^2}\log p(y_i=0|X_i,\boldsymbol{\alpha})\right)\\
&= \sum_i \mathrm{tr}\left((1-p(y_i=1|X_i,\boldsymbol{\alpha}))\tfrac{\partial^2}{\partial\boldsymbol{\alpha}^2}\log(1-p(y_i=1|X_i,\boldsymbol{\alpha}))\right)\\
&= \sum_i \mathrm{tr}\left(\left(\begin{array}{l}
\frac{1}{(1-p(y_i=1|X_i,\boldsymbol{\alpha}))}\frac{\partial}{\partial\boldsymbol{\alpha}}p(y_i=1|X_i,\boldsymbol{\alpha})\\
\left(\frac{\partial}{\partial\boldsymbol{\alpha}}p(y_i=1|X_i,\boldsymbol{\alpha})\right)^T\\
-\frac{\partial^2}{\partial\boldsymbol{\alpha}^2}p(y_i=1|X_i,\boldsymbol{\alpha})
\end{array}\right)\right).
\end{aligned}$$

Using (A-1)–(A-3), we have

$$\mathrm{tr}^{-} = \sum_i\left(\begin{array}{l}
p(y_i=0|X_i,\boldsymbol{\alpha})\\
\sum_j\left[\begin{array}{l} p(y_{ij}=0|\boldsymbol{x}_{ij},\boldsymbol{\alpha})\\ (1-p(y_{ij}=0|\boldsymbol{x}_{ij},\boldsymbol{\alpha}))\end{array}\right]\boldsymbol{h}(\boldsymbol{x}_{ij})^T\boldsymbol{h}(\boldsymbol{x}_{ij})
\end{array}\right). \tag{A-6}$$

Finally, with Eq. (A-5) and Eq. (A-6), we have

$$\begin{aligned}
&\mathrm{tr}(I(\boldsymbol{\alpha}))\\
&= \mathrm{tr}\left(\begin{array}{l}
\sum_i\left(\begin{array}{l} y_i p(y_i|X_i,\boldsymbol{\alpha})\frac{\partial^2}{\partial\boldsymbol{\alpha}^2}\log p(y_i|X_i,\boldsymbol{\alpha})\\ +(1-y_i)p(y_i|X_i,\boldsymbol{\alpha})\frac{\partial^2}{\partial\boldsymbol{\alpha}^2}\log p(y_i|X_i,\boldsymbol{\alpha})\end{array}\right)\\
+\delta I_m
\end{array}\right)\\
&= \mathrm{tr}^{+} + \mathrm{tr}^{-} + m\delta
\end{aligned}$$

which is equation (11). ■

REFERENCES

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Sur.*, vol. 38, no. 4, pp. 1–44, Dec. 2006.

[2] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, Aug. 2011.

[3] B. Settles, "Active learning literature survey," *Tech. Rep. 1648*, University of Wisconsin Madison, 2009.

[4] M. Black and A. Jepson, "EigenTracking: Robust matching and tracking of articulated objects using a view-based representation," in *Proc. Eur. Conf. Comput. Vision.*, 1996, pp. 329–342.

[5] A. Jepson, D. Fleet, and T. Maraghi, "Robust online appearance models for visual tracking," *IEEE Trans. Pattern. Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1296–1311, Oct. 2003.

[6] J. Ho, K. Lee, M. Yang, and D. Kriegman, "Visual tracking using learned linear subspace," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, Jun.–Jul. 2004, pp. 782–789.

[7] D. Ross, J. Lim, R. Lin, and M. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vis.*, vol. 77, no. 1, pp. 125–141, 2008.

[8] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, Jun. 2006, pp. 798–805.

[9] X. Mei and H. Ling, "Robust visual tracking using L1 minimization," in *Proc. IEEE Conf. Comput. Vision.*, Jun. 2009, pp. 1436–1443.

[10] J. Kwon and K. Lee, "Visual tracking decomposition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, Jun. 2010, pp. 1269–1276.

[11] L. Sun and G. Liu, "Visual object tracking based on combination of local description and global representation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 4, pp. 408–420, Apr. 2011.

[12] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 564–575, May 2003.

[13] S. Avidan, "Support vector tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 8, pp. 1064–1072, Aug. 2004.

[14] S. Avidan, "Ensemble tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 2, pp. 261–271, Feb. 2007.

[15] R. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1631–1643, Oct. 2005.

[16] J. Zhu, Y. Lao, and Y. Zheng, "Object tracking in structured environments for video surveillance applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 2, pp. 223–235, Feb. 2010.

[17] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via online boosting," in *Proc. British Mach. Vis. Conf.*, 2006, pp. 47–56.

[18] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised online boosting for robust tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 234–247.

[19] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N learning: Bootstrapping binary classifiers by structural constraints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, Jun. 2010, pp. 49–56.

[20] W. Peng and H. Qiao, "Online appearance model learning and generation for adaptive visual tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 2, pp. 156–169, Feb. 2011.

[21] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 864–877.

[22] E. Candes and T. Tao, "Near optimal signal recovery from random projections and universal encoding strategies," *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.

[23] T. Dietterich, R. Lathrop, and T. Perez, "Solving the multiple instance problem with axis-parallel rectangles," *Artif. Intell.*, vol. 89, nos. 1–2, pp. 31–71, 1997.

[24] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, Dec. 2001, pp. 511–518.

[25] M. Cover and J. Thomas, *Elements of Information Theory*. New York, NY, USA: Wiley, 1991.

[26] T. Zhang and F. Oles, "A probability analysis on the value of unlabeled data for classification problems," in *Pro. Int. Conf. Mach. Learn.*, 2000, pp. 1191–1198

[27] B. Settles, M. Craven, and S. Ray, "Multiple-instance active learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1289–1296.

[28] D. Zhang, F. Wang, Z. Shi, and C. Zhang, "Interaction localized content based image retrieval with multiple-instance active learning," *Pattern Recogn.*, vol. 43, no. 2, pp. 478–484, Feb. 2010.

[29] X. Liao, Y. Xue, and L. Carin, "Logistic regression with an auxiliary data source," in *Pro. Int. Conf. Mach. Learn.*, 2005, pp. 505–512.

[30] R. Horn and C. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1985.

[31] P. Viola, J. Platt, and C. Zhang, "Multiple instance boosting for object detection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 1417–1426.

[32] J. Friedman, T. Hastie, and R. Tibshirani, “Additive logistic regression: A statistical view of boosting,” *Ann. Stat.,* vol. 28, no. 2, pp. 337–407, 2000.

[33] C. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[34] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (VOC) challenge,” *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

Kaihua Zhang received the B.S. degree in technology and science of electronic information from the Ocean University of China, Qingdao, China, in 2006 and the master’s degree in signal and information processing from the University of Science and Technology of China, Hefei, China, in 2009. He is currently pursuing the Ph.D. degree at the Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong.

His research interests include segmentation by level set method and visual tracking by detection.

Lei Zhang (M’04) received the B.S. degree from Shenyang Institute of Aeronautical Engineering, Shenyang, China, in 1995, and the M.S. and Ph.D. degrees in automatic control theory and engineering from Northwestern Polytechnical University, Xi’an, China, in 1998 and 2001, respectively.

From 2001 to 2002, he was a Research Associate in the Deptartment of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong. From 2003 to 2006, he was a Post-Doctoral Fellow with the Deptartment of Electrical and Computer Engineering, McMaster University, Ontario, Canada. In 2006, he joined the Deptarment of Computing, Hong Kong Polytechnic University, as an Assistant Professor. Since September 2010, he has been an Associate Professor in the same department. His research interests include image and video processing, biometrics, computer vision, pattern recognition, multisensor data fusion, and optimal estimation theory.

Dr. Zhang is an Associate Editor of IEEE TRANSACTION ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and *Image and Vision Computing Journal*. Dr. Zhang was a recipient of the Faculty Merit Award in Research and Scholarly Activities in 2010 and 2012, and the Best Paper Award of SPIE VCIP2010. More information can be found in his homepage http://www4.comp.polyu.edu.hk/∼cslzhang/.

Ming-Hsuan Yang (M’92–SM’06) received the Ph.D. degree in computer science from the University of Illinois at Urbana-Champaign, Urbana, IL, USA, in 2000.

He is an Assistant Professor of electrical engineering and computer science at the University of California, Merced, CA, USA. Prior to joining University of California, Merced in 2008, he was a Senior Research Scientist with the Honda Research Institute where he worked in vision problems related to humanoid robots. He has co-authored the book entitled, *Face Detection and Gesture Recognition for Human-Computer Interaction* (Kluwer Academic, 2001) and edited a special issue on the *Face Recognition for Computer Vision and Image Understanding* in 2003, and a special issue on Real World Face Recognition for the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE.

Dr. Yang was as an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE from 2007 to 2011, and is an Associate Editor of the *Image and Vision Computing*. He was a recipient of the NSF CAREER award in 2012, the Senate Award for Distinguished Early Career Research at University of California, Merced in 2011, and the Google Faculty Award in 2009. He is a Senior Member of the ACM.

Qinghua Hu (M’11) received the B.S., M.S., and Ph.D. degrees from Harbin Institute of Technology, Harbin, China, in 1999, 2002, and 2008, respectively.

He was an Associate Professor with Harbin Institute of Technology from 2008 to 2011. He is currently a Full Professor with the School of Computer Science and Technology, Tianjin University. He has published over 70 journal and conference papers in the areas of pattern recognition and fault diagnosis. His research interests include intelligent modeling, data mining, knowledge discovery for classification, and regression.

Dr. Hu is a PC Co-Chair of RSCTC 2010 and serves as Referee for a many journals and conferences.