Fast Compressive Tracking

Kaihua Zhang, Lei Zhang, and Ming-Hsuan Yang

Abstract—It is a challenging task to develop effective and efficient appearance models for robust object tracking due to factors such as pose variation, illumination change, occlusion, and motion blur. Existing online tracking algorithms often update models with samples from observations in recent frames. Despite much success has been demonstrated, numerous issues remain to be addressed. First, while these adaptive appearance models are data-dependent, there does not exist sufficient amount of data for online algorithms to learn at the outset. Second, online tracking algorithms often encounter the drift problems. As a result of self-taught learning, misaligned samples are likely to be added and degrade the appearance models. In this paper, we propose a simple yet effective and efficient tracking algorithm with an appearance model based on features extracted from a multiscale image feature space with data-independent basis. The proposed appearance model employs non-adaptive random projections that preserve the structure of the image feature space of objects. A very sparse measurement matrix is constructed to efficiently extract the features for the appearance model. We compress sample images of the foreground target and the background using the same sparse measurement matrix. The tracking task is formulated as a binary classification via a naive Bayes classifier with online update in the compressed domain. A coarse-to-fine search strategy is adopted to further reduce the computational complexity in the detection procedure. The proposed compressive tracking algorithm runs in real-time and performs favorably against state-of-the-art methods on challenging sequences in terms of efficiency, accuracy and robustness.

Index Terms—Visual Tracking, Random Projection, Compressive Sensing.

1 Introduction

Despite that numerous algorithms have been proposed in the literature, object tracking remains a challenging problem due to appearance change caused by pose, illumination, occlusion, and motion, among others. An effective appearance model is of prime importance for the success of a tracking algorithm that has attracted much attention in recent years [2]–[16].

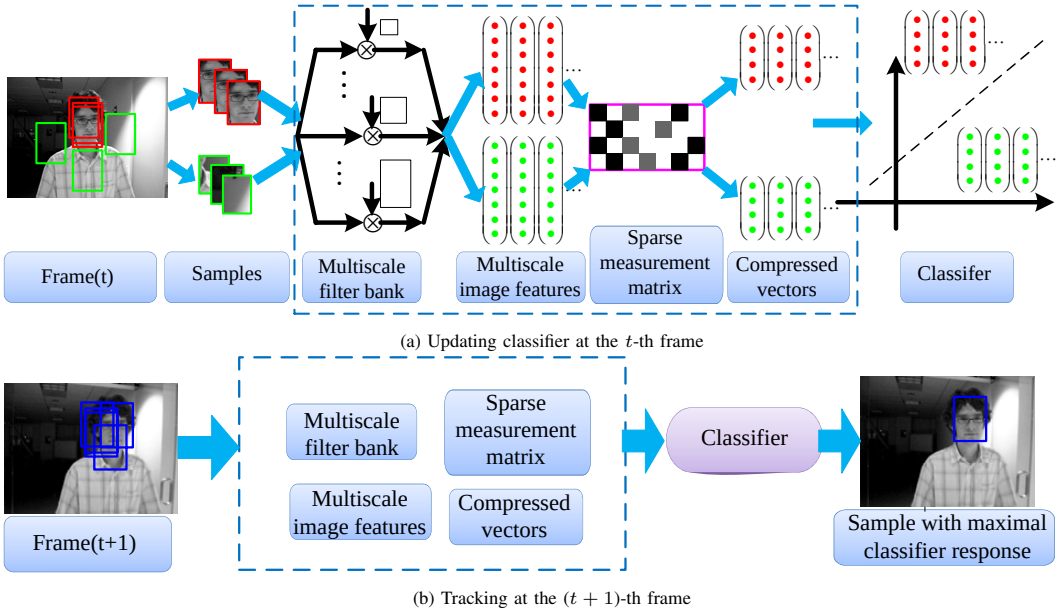
Numerous effective representation schemes have been proposed for robust object tracking in recent years. One commonly adopted approach is to learn a low-dimensional subspace (e.g., eigenspace [7], [17]), which can adapt online to object appearance change. Since this approach is data-dependent, the computational complexity is likely to increase significantly because it needs eigen-decompositions. Moreover, the noisy or misaligned samples are likely to degrade the subspace basis, thereby causing these algorithms to drift away the target objects gradually. Another successful approach is to extract discriminative features from a high-dimensional space. Since object tracking can be posed as a binary classification task which separates object from its local background, a discriminative appearance model plays an important role for its success. Online boosting methods [6], [10] have been proposed to extract discriminative features for object tracking. Alternatively, high-dimensional features can be projected to a low-dimensional space from which a classifier can be constructed.

The compressive sensing (CS) theory [18], [19] shows that if the dimension of the feature space is sufficiently high, these features can be projected to a randomly chosen low-dimensional space which contains enough information to reconstruct the original high-dimensional features. The dimensionality reduction method via random projection (RP) [20], [21] is data-independent, non-adaptive and information-preserving. In this paper, we propose an effective and efficient tracking algorithm with an appearance model based on features extracted in the compressed domain [1]. The main components of the proposed compressive tracking algorithm are shown by Figure 1. We use a very sparse measurement matrix that asymptotically satisfies the restricted isometry property (RIP) in compressive sensing theory [18], thereby facilitating efficient projection from the image feature space to a low-dimensional compressed subspace. For tracking, the positive and negative samples are projected (i.e., compressed) with the same sparse measurement matrix and discriminated by a simple naive Bayes classifier learned online. The proposed compressive tracking algorithm runs at real-time and performs favorably against state-of-the-art trackers on challenging sequences in terms of efficiency, accuracy and robustness.

The rest of this paper is organized as follows. We first review the most relevant work on online object tracking in Section 2. The preliminaries of compressive sensing and random projection are introduced in Section 3. The proposed algorithm is detailed in Section 4, and the experimental results are presented in Section 5 with comparisons to state-of-the-art methods on challenging sequences. We conclude with remarks on our future work in Section 6.

- *Early results of this work were presented in [1].*

- *Kaihua Zhang is with the School of Information and Control, Nanjing University of Information Science & Technology, Nanjing, China. E-mail: zhkhua@gmail.com.*
- *Lei Zhang is with the Department of Computing, the Hong Kong Polytechnic University, Hong Kong. E-mail: cslzhang@comp.polyu.edu.hk.*
- *Ming-Hsuan Yang is with School of Engineering, University of California, Merced, CA, 95344. E-mail: mhyang@ucmerced.edu.*



(a) Updating classifier at the t-th frame

(b) Tracking at the $(t+1)$-th frame

Fig. 1: Main components of the proposed compressive tracking algorithm.

2 Related Work

Recent surveys of object tracking can be found in [22]–[24]. In this section, we briefly review the most relevant literature of online object tracking. In general, tracking algorithms can be categorized as either generative [2], [3], [7], [9], [11], [12], [25]–[29] or discriminative [4]–[6], [8], [10], [13], [16], [30] based on their appearance models.

Generative tracking algorithms typically learn a model to represent the target object and then use it to search for the image region with minimal reconstruction error. Black et al. [2] learn an off-line subspace model to represent the object of interest for tracking. Reference templates based on color histogram [31], [32], integral histogram [25] have been used for tracking. In [3] Jepson et al. present a Gaussian mixture model with an online expectation maximization algorithm to handle object appearance variations during tracking. Ho et al. [17] propose a tracking method using a set of learned subspace model to deal with appearance change. Instead of using pre-trained subspace, the IVT method [7] learns an appearance model online to adapt appearance change. Kwon et al. [9] combine multiple observation and motion models in a modified particle filtering framework to handle large appearance and motion variation. Recently, sparse representation has been used in the ℓ_1-tracker where an object is modeled by a sparse linear combination of target and trivial templates [12]. However, the computational complexity of the ℓ_1-tracker is rather high, thereby limiting its applications in real-time scenarios. Li et al. [11] further extend it by using the orthogonal matching pursuit algorithm for solving the optimization problems efficiently, and Bao et al. [27] improve the efficiency via accelerated proximal gradient approach. A representation based on distribution of pixels at multiple layers is proposed to describe object appearance for tracking [29]. Oron et al. [28] propose a joint model of appearance and spatial configuration of pixels which estimates the amount of local distortion of the target object, thereby well handling rigid and nonrigid deformations. Recently, Zhang et al. [26] propose a multi-task approach to jointly learn the particle representations for robust object tracking. Despite much demonstrated success of these online generative tracking algorithms, several problems remain to be solved. First, numerous training samples cropped from consecutive frames are required in order to learn an appearance model online. Since there are only a few samples at the outset, most tracking algorithms often assume that the target appearance does not change much during this period. However, if the appearance of the target changes significantly, the drift problem is likely to occur. Second, these generative algorithms do not use the background information which is likely to improve tracking stability and accuracy.

Discriminative algorithms pose the tracking problem as a binary classification task with local search and determine the decision boundary for separating the target object from the background. Avidan [4] extends the optical flow approach with a support vector machine classifier for object tracking, and Collins et al. [5] demonstrate that the most discriminative features can be learned online to separate the target object from the background. In [6] Grabner et al. propose an online boosting algorithm to select features for tracking. However, these trackers [4]–[6] use one positive sample (i.e., the current tracker location) and a few negative samples when updating the classifier. As the appearance model is updated with noisy and potentially misaligned examples, this often leads to the tracking drift problem. An online semi-supervised boosting method is proposed by Grabner at al. [8] to alleviate the drift problem in which only the samples in the first frame are labeled and all the other samples are unlabeled. Babenko et al. [10] formulate online tracking within the multiple instance learning framework where samples are considered within positive and negative bags or sets. A semi-supervised learning approach [33] is developed in which positive and

negative samples are selected via an online classifier with structural constraints. Wang et al. [30] present a discriminative appearance model based on superpixels which is able to handle heavy occlusions and recovery from drift. In [13], Hare et al. use an online structured output support vector machine (SVM) for robust tracking which can mitigate the effect of wrong labeling samples. Recently, Henriques et al. [16] introduce a fast tracking algorithm which exploits the circulant structure of the kernel matrix in SVM classifier that can be efficiently computed by the fast Fourier transform algorithm.

3 PRELIMINARIES

We present some preliminaries of compressive sensing which are used in the proposed tracking algorithm.

3.1 Random projection and compressive sensing

In random projection, a random matrix $\mathbf{R} \in \mathbb{R}^{n\times m}$ whose rows have unit length projects data from the high-dimensional feature space $\mathbf{x} \in \mathbb{R}^m$ to a lower-dimensional space $\mathbf{v} \in \mathbb{R}^n$

$$\mathbf{v} = \mathbf{R}\mathbf{x}, \tag{1}$$

where $n \ll m$. Each projection $\mathbf{v}$ is essentially equivalent to a compressive measurement in the compressive sensing encoding stage. The compressive sensing theory [19], [34] states that if a signal is K-sparse (i.e., the signal is a linear combination of only K basis [35]), it is possible to near perfectly reconstruct the signal from a small number of random measurements. The encoder in compressive sensing (using (1)) correlates signal with noise (using random matrix $\mathbf{R}$) [19], thereby it is a universal encoding which requires no prior knowledge of the signal structure. In this paper, we adopt this encoder to construct the appearance model for visual tracking.

Ideally, we expect $\mathbf{R}$ provides a stable embedding that approximately preserves the salient information in any K-sparse signal when projecting from $\mathbf{x} \in \mathbb{R}^m$ to $\mathbf{v} \in \mathbb{R}^n$. A necessary and sufficient condition for this stable embedding is that it approximately preserves distances between any pairs of K-sparse signals that share the same K basis. That is, for any two K-sparse vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ sharing the same K basis,

$$(1-\epsilon)\|\mathbf{x}_1-\mathbf{x}_2\|_{\ell_2}^2 \leq \|\mathbf{R}\mathbf{x}_1-\mathbf{R}\mathbf{x}_2\|_{\ell_2}^2 \leq (1+\epsilon)\|\mathbf{x}_1-\mathbf{x}_2\|_{\ell_2}^2. \tag{2}$$

The restricted isometry property [18], [19] in compressive sensing shows the above results. This property is achieved with high probability for some types of random matrix $\mathbf{R}$ whose entries are identically and independently sampled from a standard normal distribution, symmetric Bernoulli distribution or Fourier matrix. Furthermore, the above result can be directly obtained from the Johnson-Lindenstrauss (JL) lemma [20].

Lemma 1. *(Johnson-Lindenstrauss lemma) [20]: Let Q be a finite collection of d points in $\mathbb{R}^m$. Given $0 < \epsilon < 1$ and $\beta > 0$, let n be a positive integer such that*

$$n \geq \left(\frac{4+2\beta}{\epsilon^2/2-\epsilon^3/3}\right)\ln(d). \tag{3}$$

Let $\mathbf{R} \in \mathbb{R}^{n\times m}$ be a random matrix with $\mathbf{R}(i,j) = r_{ij}$, where

$$r_{ij} = \begin{cases} +1 & \textit{with probability } \frac{1}{2} \\ -1 & \textit{with probability } \frac{1}{2}. \end{cases} \tag{4}$$

or

$$r_{ij} = \sqrt{3} \times \begin{cases} +1 & \textit{with probability } \frac{1}{6} \\ 0 & \textit{with probability } \frac{2}{3} \\ -1 & \textit{with probability } \frac{1}{6}. \end{cases} \tag{5}$$

Then, with probability exceeding $1 - d^{-\beta}$, the following statement holds: For every $\mathbf{x}_1, \mathbf{x}_2 \in Q$,

$$(1-\epsilon)\|\mathbf{x}_1-\mathbf{x}_2\|_{\ell_2}^2 \leq \frac{1}{\sqrt{n}}\|\mathbf{R}\mathbf{x}_1-\mathbf{R}\mathbf{x}_2\|_{\ell_2}^2 \leq (1+\epsilon)\|\mathbf{x}_1-\mathbf{x}_2\|_{\ell_2}^2. \tag{6}$$

Baraniuk et al. [36] prove that any random matrix satisfying the Johnson-Lindenstrauss lemma also holds true for the restricted isometry property in compressive sensing. Therefore, if the random matrix $\mathbf{R}$ in (1) satisfies the JL lemma, $\mathbf{x}$ can be reconstructed with minimum error from $\mathbf{v}$ with high probability if $\mathbf{x}$ is K-sparse (e.g., audio or image signals). This strong theoretical support motivates us to analyze the high-dimensional signals via their low-dimensional random projections. In the proposed algorithm, a very sparse matrix is adopted that not only asymptotically satisfies the JL lemma, but also can be efficiently computed for real-time tracking.

3.2 Very sparse random measurement matrix

A typical measurement matrix satisfying the restricted isometry property is the random Gaussian matrix $\mathbf{R} \in \mathbb{R}^{n\times m}$ where $r_{ij} \sim \mathcal{N}(0,1)$ (i.e., zero mean and unit variance), as used in recent work [11], [37], [38]. However, as the matrix is dense, the memory and computational loads are very expensive when m is large. In this paper, we adopt a very sparse random measurement matrix with entries defined as

$$r_{ij} = \sqrt{\rho} \times \begin{cases} 1 & \text{with probability } \frac{1}{2\rho} \\ 0 & \text{with probability } 1-\frac{1}{\rho} \\ -1 & \text{with probability } \frac{1}{2\rho}. \end{cases} \tag{7}$$

Achlioptas [20] proves that this type of matrix with $\rho = 1$ or 3 satisfies the Johnson-Lindenstrauss lemma (i.e., (4) and (5)). This matrix is easy to compute which requires only a uniform random generator. More importantly, when $\rho = 3$, it is sparse where two thirds of the computation can be avoided. In addition, Li et al. [39] show that for $\rho = o(m)$ ($\mathbf{x} \in \mathbb{R}^m$), the random projections are almost as accurate as the conventional random projections where $r_{ij} \sim \mathcal{N}(0,1)$. Therefore, the random matrix (7) with $\rho = o(m)$ asymptotically satisfies the JL lemma. In this work, we set $\rho = o(m) = m/(a\log_{10}(m)) = m/(10a) \sim m/(6a)$ with a fixed constant a because the dimensionality m is typically in the order of 10^6 to 10^{10}. For each row of $\mathbf{R}$, only about $c = (\frac{1}{2\rho} + \frac{1}{2\rho}) \times m = a\log_{10}(m) \leq 10a$ nonzero entries need to be computed. We observe that good results can be obtained by fixing $a = 0.4$ in our experiments. Therefore, the computational complexity is only $o(cn)$ ($n = 100$ in this work) which is very low. Furthermore, only the nonzero entries of $\mathbf{R}$ need to be stored which makes the memory requirement also very light.

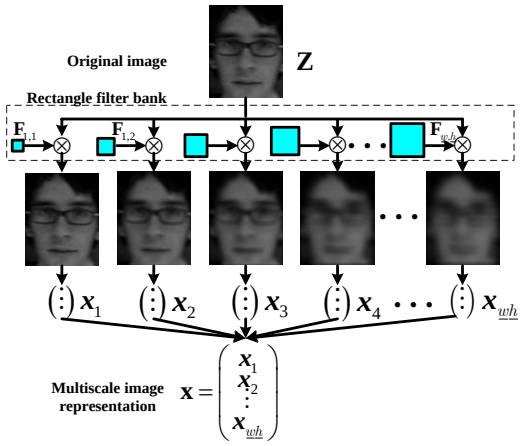


Fig. 2: Illustration of multiscale image representation.

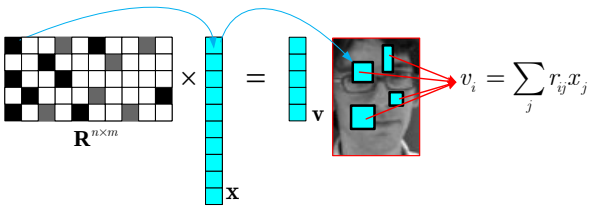


Fig. 3: Graphical representation of compressing a high-dimensional vector $\mathbf{x}$ to a low-dimensional vector $\mathbf{v}$. In the matrix $\mathbf{R}$, dark, gray and white rectangles represent negative, positive, and zero entries, respectively. The blue arrows illustrate that one of nonzero entries of one row of $\mathbf{R}$ sensing an element in $\mathbf{x}$ is equivalent to a rectangle filter convolving the intensity at a fixed position of an input image.

4 Proposed Algorithm

In this section, we present the proposed compressive tracking algorithm in details. The tracking problem is formulated as a detection task and the main steps of the proposed algorithm are shown in Figure 1. We assume that the tracking window in the first frame is given by a detector or manual label. At each frame, we sample some positive samples near the current target location and negative samples away from the object center to update the classifier. To predict the object location in the next frame, we draw some samples around the current target location and determine the one with the maximal classification score.

4.1 Image representation

To account for large scale change of object appearance, a multiscale image representation is often formed by convolving the input image with a Gaussian filter of different spatial variances [40]. The Gaussian filters in practice have to be truncated which can be replaced by rectangle filters. Bay et al. [41] show that this replacement does not affect the performance of the interest point detectors but can significantly speed up the detectors via integral image method [42].

For each sample $\mathbf{Z} \in \mathbb{R}^{w \times h}$, its multiscale representation (as illustrated in Figure 2) is constructed by convolving $\mathbf{Z}$ with a set of rectangle filters at multiple scales $\{\mathbf{F}_{1,1}, \ldots, \mathbf{F}_{w,h}\}$ defined by

$$\mathbf{F}_{w,h}(x,y) = \frac{1}{wh} \times \begin{cases} 1, & 1 \leq x \leq w,\ 1 \leq y \leq h \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where w and h are the width and height of a rectangle filter, respectively.

Then, we represent each filtered image as a column vector in $\mathbb{R}^{wh}$ and concatenate these vectors as a very high-dimensional multiscale image feature vector $\mathbf{x} = (x_1, ..., x_m)^\top \in \mathbb{R}^m$ where $m = (wh)^2$. The dimensionality m is typically in the order of 10^6 to 10^{10}. We adopt a sparse random matrix $\mathbf{R}$ in (7) to project $\mathbf{x}$ onto a vector $\mathbf{v} \in \mathbb{R}^n$ in a low-dimensional space. The random matrix $\mathbf{R}$ needs to be computed only once off-line and remains fixed throughout the tracking process. For the sparse matrix $\mathbf{R}$ in (7), the computational load is very light. As shown in Figure 3, we only need to store the nonzero entries in $\mathbf{R}$ and the positions of rectangle filters in an input image corresponding to the nonzero entries in each row of $\mathbf{R}$. Then, $\mathbf{v}$ can be efficiently computed by using $\mathbf{R}$ to sparsely measure the rectangular features which can be efficiently computed using the integral image method [42].

4.2 Analysis of compressive features

4.2.1 Relationship to the Haar-like features

As shown in Figure 3, each element v_i in the low-dimensional feature $\mathbf{v} \in \mathbb{R}^n$ is a linear combination of spatially distributed rectangle features at different scales. Since the coefficients in the measurement matrix can be positive or negative (via (7)), the compressive features compute the relative intensity difference in a way similar to the generalized Haar-like features [10] (See Figure 3). The Haar-like features have been widely used for object detection with demonstrated success [10], [42], [43]. The basic types of these Haar-like features are typically designed for different tasks [42], [43]. There often exist a very large number of Haar-like features which make the computational load very heavy. This problem is alleviated by boosting algorithms for selecting important features [42], [43]. Recently, Babenko et al. [10] adopt the generalized Haar-like features where each one is a linear combination of randomly generated rectangle features, and use online boosting to select a small set of them for object tracking. In this work, the large set of Haar-like features are compressively sensed with a very sparse measurement matrix. The compressive sensing theories ensure that the extracted features of our algorithm preserve almost all the information of the original image, and hence is able to correctly classify any test image because the dimension of the feature space is sufficiently large (10^6 to 10^{10}) [37]. Therefore, the projected features can be classified in the compressed domain efficiently and effectively without the curse of dimensionality.

4.2.2 Scale invariant property

It is easy to show that the low-dimensional feature $\mathbf{v}$ is scale invariant. As shown in Figure 3, each feature in $\mathbf{v}$ is a linear combination of some rectangle filters convolving the input image at different positions. Therefore, without loss of

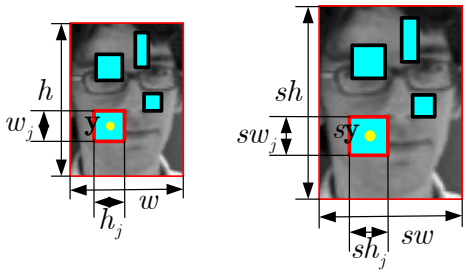


Fig. 4: Illustration of scale invariant property of low-dimensional features. From the left figure to the right one, the ratio is s. Red rectangle represents the j-th rectangle feature at position $\mathbf{y}$.

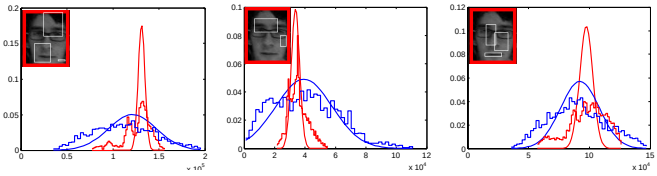


Fig. 5: Probability distributions of three different features in a low-dimensional space. The red stair represents the histogram of positive samples while the blue one represents the histogram of negative samples. The red and blue lines denote the corresponding estimated distributions by the proposed incremental update method.

generality, we only need to show that the j-th rectangle feature x_j in the i-th feature v_i in $\mathbf{v}$ is scale invariant. From Figure 4, we have

$$\begin{aligned} x_j(s\mathbf{y}) &= \mathbf{F}_{sw_j,sh_j}(s\mathbf{y}) \otimes \mathbf{Z}(s\mathbf{y}) \\ &= \mathbf{F}_{sw_j,sh_j}(\mathbf{a}) \otimes \mathbf{Z}(\mathbf{a})|_{\mathbf{a}=s\mathbf{y}} \\ &= \frac{1}{s^2 w_i h_i} \int_{\mathbf{u}\in\Omega_s} \mathbf{Z}(\mathbf{a}-\mathbf{u}) d\mathbf{u} \\ &= \frac{1}{s^2 w_i h_i} \int_{\mathbf{u}\in\Omega} \mathbf{Z}(\mathbf{y}-\mathbf{u}) |s^2| d\mathbf{u} \\ &= \frac{1}{w_i h_i} \int_{\mathbf{u}\in\Omega} \mathbf{Z}(\mathbf{y}-\mathbf{u}) d\mathbf{u} \\ &= \mathbf{F}_{w_j,h_j}(\mathbf{y}) \otimes \mathbf{Z}(\mathbf{y}) \\ &= x_j(\mathbf{y}), \end{aligned}$$

where $\Omega = \{(u_1, u_2) | 1 \leq u_1 \leq w_i, 1 \leq u_2 \leq h_i\}$ and $\Omega_s = \{(u_1, u_2) | 1 \leq u_1 \leq sw_i, 1 \leq u_2 \leq sh_i\}$.

4.3 Classifier construction and update

We assume all elements in $\mathbf{v}$ are independently distributed and model them with a naive Bayes classifier [44],

$$\begin{aligned} H(\mathbf{v}) &= \log \left(\frac{\prod_{i=1}^{n} p(v_i|y=1)p(y=1)}{\prod_{i=1}^{n} p(v_i|y=0)p(y=0)} \right) \\ &= \sum_{i=1}^{n} \log \left(\frac{p(v_i|y=1)}{p(v_i|y=0)} \right), \end{aligned} \quad (9)$$

where we assume uniform prior, $p(y=1) = p(y=0)$, and $y \in \{0, 1\}$ is a binary variable which represents the sample label.

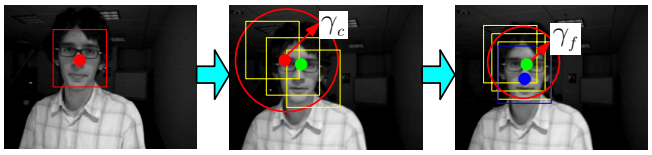


Fig. 6: Coarse-to-fine search for new object location. Left: object center location (denoted by red solid circle) at the t-th frame. Middle: coarse-grained search with a large radius γ_c and search step Δ_c based on the previous object location. Right: fine-grained search with a small radius $\gamma_f < \gamma_c$ and search step $\Delta_f < \Delta_c$ based on the coarse-grained search location (denoted by green solid circle). The final object location is denoted by blue solid circle.

Diaconis and Freedman [45] show that random projections of high dimensional random vectors are almost always Gaussian. Thus, the conditional distributions $p(v_i|y=1)$ and $p(v_i|y=0)$ in the classifier $H(\mathbf{v})$ are assumed to be Gaussian distributed with four parameters $(\mu_i^1, \sigma_i^1, \mu_i^0, \sigma_i^0)$,

$$p(v_i|y=1) \sim \mathcal{N}(\mu_i^1, \sigma_i^1), \quad p(v_i|y=0) \sim \mathcal{N}(\mu_i^0, \sigma_i^0), \quad (10)$$

where μ_i^1 (μ_i^0) and σ_i^1 (σ_i^0) are mean and standard deviation of the positive (negative) class. The scalar parameters in (10) are incrementally updated by

$$\begin{aligned} \mu_i^1 &\leftarrow \lambda \mu_i^1 + (1-\lambda)\mu^1 \\ \sigma_i^1 &\leftarrow \sqrt{\lambda(\sigma_i^1)^2 + (1-\lambda)(\sigma^1)^2 + \lambda(1-\lambda)(\mu_i^1 - \mu^1)^2}, \end{aligned} \quad (11)$$

where $\lambda > 0$ is a learning parameter, $\sigma^1 = \sqrt{\frac{1}{n}\sum_{k=0|y=1}^{n-1}(v_i(k) - \mu^1)^2}$ and $\mu^1 = \frac{1}{n}\sum_{k=0|y=1}^{n-1} v_i(k)$. Parameters μ_i^0 and σ_i^0 are updated with similar rules. The above equations can be easily derived by maximum likelihood estimation [46]. Figure 5 shows the probability distributions for three different features of the positive and negative samples cropped from a few frames of a sequence for clarity of presentation. It shows that a Gaussian distribution with online update using (11) is a good approximation of the features in the projected space where samples can be easily separated.

Because the variables are assumed to be independent in our classifier, the n-dimensional multivariate problem is reduced to the n univariate estimation problem. Thus, it requires fewer training samples to obtain accurate estimation than estimating the covariance matrix in the multivariate estimation. Furthermore, several densely sampled positive samples surrounding the current tracking result are used to update the distribution parameters, which is able to obtain robust estimation even when the tracking result has some drift. In addition, the useful information from the former accurate samples is also used to update the parameter distributions, thereby facilitating the proposed algorithm to be robust to misaligned samples. Thus, our classifier performs robustly even when the misaligned or the insufficient number of training samples are used.

4.4 Fast compressive tracking

The aforementioned classifier is used for local search. To reduce the computational complexity, a coarse-to-fine sliding

Algorithm 1 (Scaled) Fast Compressive Tracking

Input: the t-th image frame

1: Coarsely sample a set of image patches in $D^{\gamma_c} = \{\mathbf{Z}|\|\mathbf{l}(\mathbf{Z})-\mathbf{l}_{t-1}\| < \gamma_c\}$ where $\mathbf{l}_{t-1}$ is the tracking location at the $(t-1)$-th frame by shifting a number of pixels Δ_c, and extract the features with low dimensionality.

2: Use classifier H in (9) to each feature vector $\mathbf{v}(\mathbf{Z})$ and find the tracking location $\mathbf{l}'_t$ with the maximal classifier response.

3: Finely sample a set of image patches in $D^{\gamma_f} = \{\mathbf{Z}|\|\mathbf{l}(\mathbf{Z})-\mathbf{l}'_t\| < \gamma_f\}$ by shifting a number of pixels Δ_f, and extract the features with low dimensionality.

4: Use classifier H in (9) to each feature vector $\mathbf{v}(\mathbf{Z})$ and find the tracking location $\mathbf{l}_t$ with the maximal classifier response. (For multiscale tracking, update the tracking location and scale every fifth frame as $(\mathbf{l}_t(\mathbf{Z}), s) = \arg\max_{\mathbf{v}^s(\mathbf{Z})\in\mathcal{F}} H(\mathbf{v}^s(\mathbf{Z}))$).

5: Sample two sets of image patches $D^{\alpha} = \{\mathbf{Z}|\|\mathbf{l}(\mathbf{Z})-\mathbf{l}_t\| < \alpha\}$ and $D^{\zeta,\beta} = \{\mathbf{Z}|\zeta < \|\mathbf{l}(\mathbf{Z})-\mathbf{l}_t\| < \beta\}$ with $\alpha < \zeta < \beta$, and extract the features with these two sets of samples.

6: Update the classifier parameters according to (11).

Output: Tracking location $\mathbf{l}_t$ (and scale s for multiscale tracking) and classifier parameters

window search strategy is adopted (See Figure 6). The main steps of our algorithm are summarized in Algorithm 1. First, we search the object location based on the previous object location by shifting the window with a large number of pixels Δ_c within a large search radius γ_c. This generates fewer windows than locally exhaustive search method (e.g., [10]) but the detected object location may be slightly inaccurate but close to the accurate object location. Based on the coarse-grained detected location, fine-grained search is carried out with a small number of pixels Δ_f within a small search radius γ_f. For example, we set $\gamma_c = 25$, $\Delta_c = 4$, and $\gamma_f = 10$, $\Delta_f = 1$ in all the experiments. If we use the fine-grained locally exhaustive method with $\gamma_c = 25$ and $\Delta_f = 1$, the total number of search windows is about 1,962 (i.e., $\pi\gamma_c^2$). However, using this coarse-to-fine search strategy, the total number of search windows is about 436 (i.e., $\pi\gamma_c^2/16 + \pi\gamma_f^2$), thereby significantly reducing computational cost.

4.4.1 Multiscale fast compressive tracking

At each location in the search region, three image patches are cropped at different scale s: current ($s = 1$), small ($s = 1-\delta$) and large scale ($s = 1+\delta$), to account for appearance variation due to fast scale change. The template of each rectangle feature for patch with scale s is multiplied by ratio s (See Figure 4). Therefore, the feature $\mathbf{v}^s$ for each patch with scale s can be efficiently extracted by using the integral image method [42]. Since the low-dimensional features for each image patch are scale invariant, we have $\mathbf{v}_t^s = \arg\max_{\mathbf{v}\in\mathcal{F}} H(\mathbf{v}) \approx \mathbf{v}_{t-1}$, where $\mathbf{v}_{t-1}$ is the low-dimensional feature vector that represents the object in the $(t-1)$-th frame, and $\mathcal{F}$ is the set of low-dimensional features extracted from image patches at different scales. The classifier is updated with cropped positive and negative samples based on the new object location and scale. The above procedures can be easily integrated into Algorithm 1: the scale is updated every fifth frame in the fine-grained search procedure (See Step 4 in Algorithm 1), which is a tradeoff between computational efficiency and effectiveness of handling appearance change caused by fast scale change.

4.5 Discussion

We note that simplicity is the prime characteristic of the proposed algorithm in which the proposed sparse measurement matrix $\mathbf{R}$ is independent of training samples, thereby resulting in an efficient method. In addition, the proposed algorithm achieves robust performance as discussed below.

Difference with related work. It should be noted that the proposed algorithm is different from recent work based on sparse representation [12] and compressive sensing [11]. First, both algorithms are generative models that encode an object sample by sparse representation of templates using ℓ_1-minimization. Thus the training samples cropped from the previous frames are stored and updated, but this is not required in the proposed algorithm due to the use of a data-independent measurement matrix. Second, the proposed algorithm extracts a linear combination of generalized Haar-like features and other trackers [12] [11] use sparse representations of holistic templates which are less robust as demonstrated in the experiments. Third, both tracking algorithms [12] [11] need to solve numerous time-consuming ℓ_1-minimization problems although one method has been recently proposed to alleviate the problem of high computational complexity [27]. In contrast, the proposed algorithm is efficient as only matrix multiplications are required.

The proposed method is different from the MIL tracker [10] as it first constructs a feature pool in which each feature is randomly generated as a weighted sum of pixels in 2 to 4 rectangles. A subset of most discriminative features are then selected via an MIL boosting method to construct the final strong classifier. However, as the adopted measurement matrix of the proposed algorithm satisfies the JL lemma, the compressive features can preserve the ℓ_2 distance of the original high-dimensional features. Since each feature that represents a target or background sample is assumed to be independently distributed with a Gaussian distribution, the feature vector for each sample is modeled as a mixture of Gaussian (MoG) distribution. The MoG distribution is essentially a mixture of weighted ℓ_2 distances of Gaussian distributions. Thus, the ℓ_2 distance between the target and background distributions is preserved in the compressive feature space, and the proposed algorithm can obtain favorable results without further learning the discriminative features from the compressive feature space.

Discussion with the online AdaBoost method [6]. The reasons that our method performs better than the OAB method can be attributed to the following factors. First, to reduce the computational complexity, the feature pool size designed by the OAB method is small (less than 250 according to the default setting in [6] which may contain insufficient

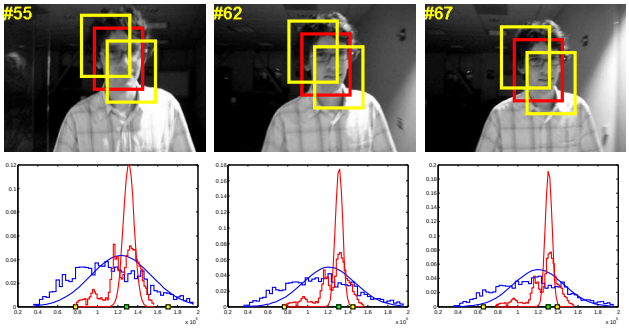


Fig. 7: Illustration of the proposed algorithm in dealing with ambiguity in detection. Top row: three positive samples. The sample in red rectangle is the most "correct" positive sample while other two in yellow rectangles are less "correct" positive samples. Bottom row: the probability distributions for a feature extracted from positive and negative samples. The green markers denote the feature extracted from the most "correct" positive sample while the yellow markers denote the feature extracted from the two less "correct" positive samples. The red and blue stairs as well as lines denote the estimated distributions of positive and negative samples as shown in Figure 5.

discriminative features. However, our compressive features can preserve the intrinsic discriminative strength of the original high-dimensional multiscale features, i.e., large (between 10^6 and 10^{10}) feature space . Therefore, our compressive features have better discriminative capability than the Haar-like features used by the OAB method. Second, the proposed method uses several positive samples (patches close to the tracking result at any frame) for online update of the appearance model which alleviates the errors introduced by inaccurate tracking locations, whereas the OAB method only uses one positive sample (i.e., the tracking result). When the tracking location is not accurate, the appearance model of the OAB method will not be updated properly and thereby cause drift.

Random projection vs. principal component analysis. For visual tracking, dimensionality reduction algorithms such as principal component analysis (PCA) and its variations have been widely used in generative approaches [2], [7]. These methods need to update the appearance models frequently for robust tracking. However, these methods are usually sensitive to heavy occlusion due to the holistic representation schemes although some robust schemes have been proposed [47]. Furthermore, it is not clear whether the appearance models can be updated correctly with new observations (e.g., without alignment errors to avoid tracking drift). In contrast, the proposed algorithm does not suffer from the problems with online self-taught learning approaches [48] as the proposed model with the measurement matrix is data-independent. It has been shown that for image and text applications, favorable results are achieved by methods with random projection than principal component analysis [21].

Robustness to ambiguity in detection. The tracking-by-detection methods often encounter the inherent ambiguity problems as shown in Figure 7. Recently Babenko et al. [10] introduce online multiple instance learning schemes to alleviate the tracking ambiguity problem. The proposed algorithm is robust to the ambiguity problem as illustrated in Figure 7. While the target appearance changes over time, the most "correct" positive samples (e.g., the sample in the red rectangle in Figure 7) are similar in most frames. However, the less "correct" positive samples (e.g., samples in yellow rectangles of Figure 7) are much more different as they contain some background pixels which vary much more than those within the target object. Thus, the distributions for the features extracted from the most "correct" positive samples are more concentrated than those from the less "correct" positive samples. This in turn makes the features from the most "correct" positive samples much more stable than those from the less "correct" positive samples (e.g., on the bottom row of Figure 7, the features denoted by red markers are more stable than those denoted by yellow markers). The proposed algorithm is able to select the most "correct" positive sample because its probability is larger than those of the less "correct" positive samples (See the markers in Figure 7). In addition, the proposed measurement matrix is data-independent and no noise is introduced by mis-aligned samples.

Robustness to occlusion. Each feature in the proposed algorithm is spatially localized (See Figure 3) which is less sensitive to occlusion than methods based on holistic representations. Similar representations, e.g., local binary patterns [49], Haar-like features [6], [10], have been shown to be effective in handling occlusion. Furthermore, features are randomly sampled at multiple scales by the proposed algorithm in a way similar to [10], [50] which have demonstrated robust results for dealing with occlusion.

Dimensionality of projected space. Bingham and Mannila [21] show that in practice the bound of the Johnson-Lindenstrauss lemma (i.e., (3)) is much higher than that suffices to achieve good results on image and text data. In [21], the lower bound for n when $\epsilon = 0.2$ is $1,600$ but $n = 50$ is sufficient to generate good results for image and text analysis. In the experiments, with 100 samples (i.e., $d = 100$), $\epsilon = 0.2$ and $\beta = 1$, the lower bound for n is approximately $1,600$. Another bound derived from the restricted isometry property in compressive sensing [18] is much tighter than that from the Johnson-Lindenstrauss lemma, where $n \geq \kappa\beta \log(m/\beta)$ and κ and β are constants. For $m = 10^6, \kappa = 1$, and $\beta = 10$, it is expected that $n \geq 50$. We observe that good results can be obtained when $n = 100$ in the experiments.

Robustness to preserve important features. With the setting in this work, $d = 100$ and $\beta = 1$, the probability that preserves the pair-wise distances in the JL lemma (See Lemma 1) exceeds $1 - d^{-\beta} = 99\%$. Assume that there exists only one important feature that can separate the foreground object from the background. Since each compressed feature is assumed to be generated from an identical and independent distribution, it is reasonable to assume that each feature contains or looses the piece of important information with the same probability, i.e., $p_i(y = 1) = p_i(y = 0) = 50\%, i = 1, \ldots, n$, where $y = 1$ indicates the feature contains the piece of important information while $y = 0$ otherwise. Therefore, the probability that the only important feature being lost is less

TABLE 1: Summary of all evaluated tracking algorithms.

Trackers	Object representation	Adaptive appearance model[1]	Approach	Classifier
Frag [25]	local intensity histogram	-	generative	-
IVT [7]	holistic image intensity	incremental principal component analysis	generative	-
VTD [9]	hue, saturation, intensity and edge template	sparse principal component analysis	generative	-
L1T [12], CS [11]	holistic image intensity	sparse representation	generative	-
DF [29]	multi-layer distribution fields	-	generative	-
MTT [26]	holistic image intensity	multi-task learning	generative	-
OAB [6]	Haar-like, HOG, and LBP features	online boosting	discriminative	boosting
SemiB [8]	Haar-like features	online semi-supervised boosting	discriminative	boosting
MIL [10]	Haar-like features	online multiple instance learning	discriminative	boosting
TLD [33]	Haar-like features	-	discriminative	cascaded
Struck [13]	Haar-like features	-	discriminative	structured SVM
CST [16]	holistic image intensity	-	discriminative	SVM
SCM [51]	holistic image intensity and local histograms	sparse representation	hybrid	-
ASLA [52]	local image patches	sparse representation	generative	-
CT [1], **FCT**, **SFCT**	Haar-like features	-	discriminative	naive Bayes

[1]For the discriminative trackers, online feature selection methods are adopted to refine appearance models where features including histogram of oriented gradients (HOG) and local binary pattern (LBP) are used.

than $p = d^{-\beta} \times \prod_{i=1}^{n} p_i(y=0) = 1\% \times 0.5^{100} \approx 0$ when a failure happens.

5 EXPERIMENTS

The proposed algorithm is termed as fast compressive tracker (FCT) with one fixed scale, and scaled FCT (SFCT), with multiple scales in order to distinguish from the compressive tracker (CT) proposed by our conference paper [1]. The FCT and SFCT methods demonstrate superior performance over the CT method in terms of accuracy and efficiency (See results in Table 2 and Table 3), which validates the effectiveness of the scale invariant features and coarse-to-fine search strategy. Furthermore, the proposed algorithm is evaluated with other 15 state-of-the-art methods on 20 challenging sequences among which 14 are publicly available and 6 are collected on our own (i.e., *Biker*, *Bolt*, *Chasing*, *Goat*, *Pedestrian*, and *Shaking 2* in Table 2). The 15 evaluated trackers are the compressive sensing (CS) tracker [11], the fragment tracker (Frag) [25], online AdaBoost method (OAB) [6], Semi-supervised tracker (SemiB) [8], incremental visual tracker (IVT) [7], MIL tracker [10], visual tracking decomposition (VTD) algorithm [9], ℓ_1-tracker (L1T) [12], TLD tracker [33], distribution field (DF) tracker [29], multi-task tracker (MTT) [26], Struck (Struck) method [13], , circulant structure tracker (CST) [16], sparsity-based collaborative model (SCM) tracker [51] and adaptive structural local sparse appearance (ASLA) tracker [52]. Table 1 summarizes the characteristics of the evaluated tracking algorithms. Most of the compared discriminative algorithms rely on either refined features (via feature selection such as OAB, SemiB, MIL) or strong classifiers (SVM classifier such as Struck and CST) for object tracking. For the TLD method, it uses a detector integrated with a cascade of three classifiers (i.e., patch variance, random ferns, and nearest neighbor classifiers) for tracking. While the proposed tracking algorithm uses Haar-like features (via random projection) and simple naive Bayes classifier, it achieves favorable results against other methods.

It is worth noticing that the most challenging sequences from the existing works are used for evaluation. All parameters in the proposed algorithm are *fixed for all the experiments* to demonstrate the robustness and stability of the proposed method. To fairly verify the effectiveness of the scale invariant compressive feature and the coarse-of-fine search strategy, the dimensionality of the compressive feature space for the CT method [1] is set to 100 as the FCT and SFCT. For other evaluated trackers, we use the source or binary codes provided by the authors with *default* parameters. Note that these settings are different in our conference paper [1] in which we either use the tuned parameters from the source codes or empirically set them for best results. Therefore, the results of some baseline methods are different. For fair comparisons, all the evaluated trackers are initialized with the same parameters (e.g., initial locations, number of particles and search range). The proposed FCT algorithm runs at 149 frame per second (FPS) with a MATLAB implementation on an i7 Quad-Core machine with 3.4 GHz CPU and 32 GB RAM. In addition, the SFCT algorithm runs 135 frames per second. Both run faster than the CT algorithm (80 FPS) [1], illustrating the efficiency of coarse-to-fine search scheme. The CS algorithm [11] runs 40 FPS, which is much less efficient than our proposed algorithms because of its solving a time-consuming ℓ_1-minimization problem. The source codes, videos, and data sets are available at http://www4.comp.polyu.edu.hk/~cslzhang/FCT/FCT.htm.

5.1 Experimental setup

Given a target location at the current frame, the search radius for drawing positive samples α is set to 4 which generates 45 positive samples. The inner ζ and outer radii β for the set $D^{\zeta,\beta}$ that generates negative samples are set to 8 and 30, respectively. In addition, 50 negative samples are randomly selected from the set $D^{\zeta,\beta}$. The search radius γ_c for the set D^{γ_c} to coarsely detect the object location is 25 and the shifting step Δ_c is 4. The radius γ_f for set D^{γ_f} to fine-grained search is set to 10 and the shifting step Δ_f is set to 1. The scale change parameter δ is set to 0.01. The dimensionality of projected space n is set to 100, and the learning parameter λ is set to 0.85.

TABLE 2: Success rate (SR)(%). **Bold** fonts indicate the best performance while the *italic* fonts indicate the second best ones. The total number of evaluated frames is $8,762$.

Sequence	SFCT	FCT	CT	CS	Frag	OAB	SemiB	IVT	MIL	VTD	L1T	TLD	DF	MTT	Struck	CST	SCM	ASLA
Animal	*99*	92	96	4	3	17	51	4	83	96	6	37	6	87	96	**100**	98	96
Biker	**85**	35	*84*	5	3	66	39	10	1	15	3	2	6	9	9	9	5	10
Bolt	**99**	*94*	90	5	41	0	18	17	92	0	2	0	1	1	8	92	1	1
Cliff bar	95	**99**	89	24	24	66	24	47	71	53	24	63	26	55	44	*96*	41	40
Chasing	88	79	47	67	21	71	62	*91*	65	70	72	76	70	**96**	85	**96**	64	63
Coupon book	**99**	*98*	97	17	26	*98*	23	*98*	*98*	17	16	31	34	39	*98*	81	32	71
David indoor	**99**	*98*	94	8	8	32	46	*98*	71	*98*	83	*98*	51	41	33	66	30	34
Dark car	*75*	36	53	6	0	14	19	54	48	25	46	67	**78**	59	18	48	47	57
Football	*77*	76	74	35	26	31	17	64	*77*	**83**	35	59	56	67	62	69	17	7
Goat	75	*77*	26	26	14	46	43	37	27	39	24	48	44	39	59	**89**	57	37
Occluded face	*98*	**99**	**99**	39	54	49	41	96	97	79	96	87	78	88	*97*	**99**	76	93
Panda	**91**	84	*90*	1	9	83	71	11	80	7	63	34	13	11	43	15	29	71
Pedestrian	*82*	**83**	13	1	0	1	3	0	1	3	4	0	7	4	1	1	5	1
Skating	*96*	**97**	83	7	11	68	39	8	21	*96*	65	37	19	10	84	9	76	61
Shaking 1	72	84	80	9	25	39	30	1	83	*93*	3	15	84	2	48	36	54	**98**
Shaking 2	**97**	88	55	12	34	74	46	39	41	80	36	56	*95*	93	53	43	86	82
Sylvester	*83*	77	69	57	34	65	66	45	77	33	40	**89**	32	67	80	*83*	76	82
Tiger 1	49	52	50	62	19	24	29	8	34	*78*	18	40	36	25	**87**	42	31	14
Tiger 2	61	**72**	48	11	12	36	16	19	44	13	11	24	*65*	34	62	37	2	24
Twinnings	72	*98*	70	41	73	**99**	23	49	83	75	82	91	82	77	95	86	89	63
Average SR	**86**	*82*	73	29	33	56	43	49	68	51	47	57	50	50	64	66	51	58

TABLE 3: Center location error (CLE)(in pixels) and average frame per second (FPS). **Bold** fonts indicate the best performance while the *italic* fonts indicate the second best ones. The total number of evaluated frames is $8,762$.

Sequence	SFCT	FCT	CT	CS	Frag	OAB	SemiB	IVT	MIL	VTD	L1T	TLD	DF	MTT	Struck	CST	SCM	ASLA
Animal	**13**	*15*	16	271	100	62	26	207	32	16	122	125	252	17	19	*15*	16	**13**
Biker	**6**	12	**6**	176	107	*10*	14	111	44	86	89	166	76	68	95	53	227	109
Bolt	**8**	10	*9*	152	44	227	102	60	**8**	146	261	286	277	293	148	12	200	210
Cliff bar	*7*	**6**	8	69	34	33	56	37	14	31	40	70	52	25	46	*7*	99	49
Chasing	9	10	12	9	56	9	44	*5*	13	23	9	47	31	**4**	*5*	**4**	61	47
Coupon book	*5*	**4**	7	175	62	9	74	**4**	6	74	75	81	23	72	6	21	73	23
David indoor	*8*	11	14	72	73	57	37	**6**	19	**6**	17	*8*	56	125	64	18	150	57
Dark car	*7*	9	10	89	116	11	11	8	9	20	8	13	**6**	*7*	9	8	45	8
Football	*8*	13	14	43	144	37	58	10	13	**6**	39	15	33	9	26	17	200	207
Goat	20	*18*	103	137	140	71	77	94	109	92	88	103	86	99	22	**9**	75	94
Occluded face	**11**	*12*	16	29	57	36	39	14	17	36	17	24	22	19	15	13	24	20
Panda	**6**	**6**	10	157	56	8	9	58	*7*	61	9	16	64	47	11	46	156	9
Pedestrian	*7*	**6**	70	78	160	91	86	84	71	74	76	211	90	76	72	104	210	93
Skating	16	14	21	207	176	74	76	144	136	**9**	87	204	174	78	15	*10*	42	72
Shaking 1	13	*10*	14	119	55	22	134	122	12	**6**	72	232	*10*	115	24	21	47	*10*
Shaking 2	*14*	15	46	255	119	18	124	109	58	41	113	144	**7**	16	48	84	18	27
Sylvester	*9*	*9*	14	84	47	12	14	138	*9*	66	50	**8**	56	18	10	**8**	10	*9*
Tiger 1	*13*	23	25	48	39	42	38	45	27	**8**	37	24	30	61	**8**	25	146	49
Tiger 2	16	**10**	17	84	36	22	30	44	18	47	48	40	13	24	*11*	22	230	36
Twinnings	11	10	15	44	15	**7**	70	23	11	19	11	*8*	12	12	9	11	9	29
Average CLE	**9**	*10*	18	96	60	31	48	56	22	42	45	65	48	57	24	23	87	45
Average FPS	135	*149*	80	40	6	22	11	33	38	6	0.5	28	13	5	20	**362**	1	7

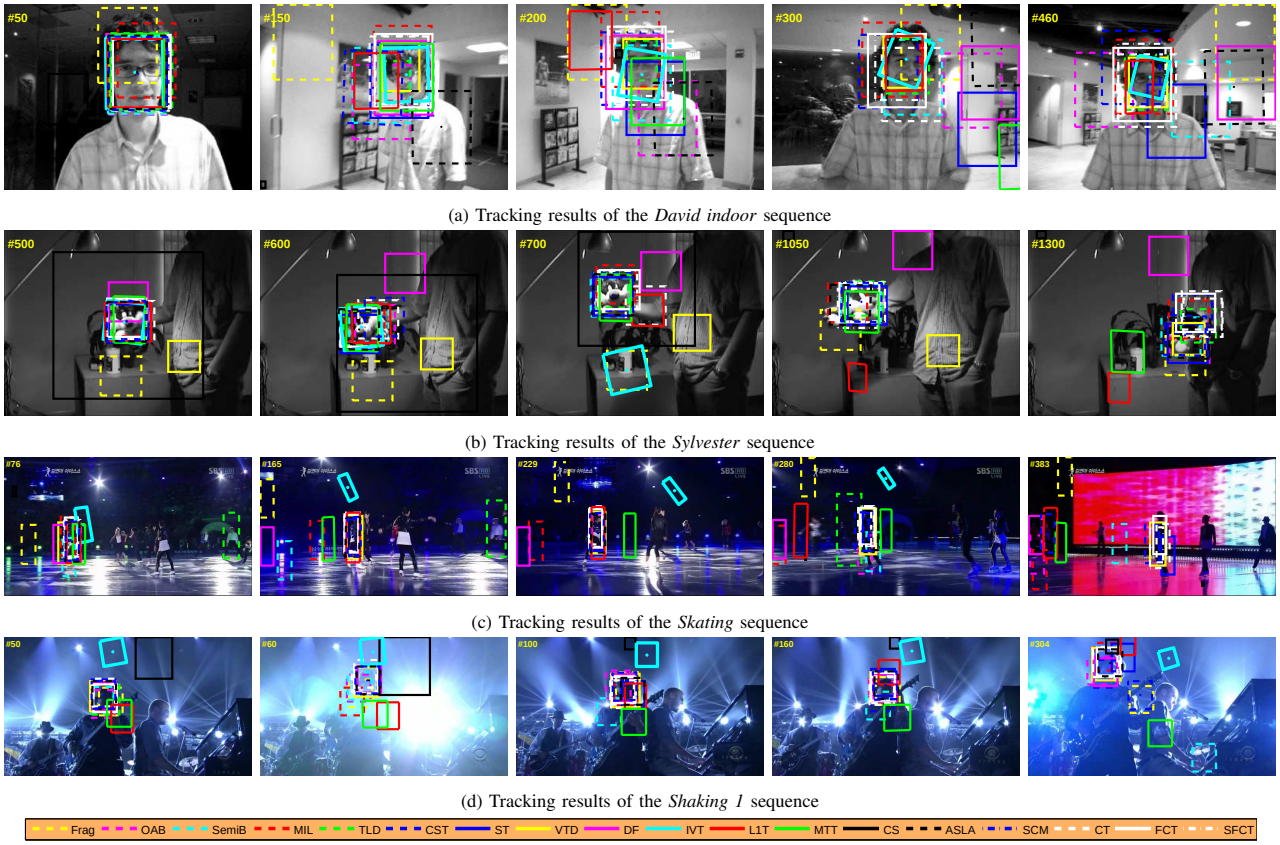
5.2 Evaluation criteria

Two metrics are used to evaluate the proposed algorithm with 15 state-of-the-art trackers in which gray scale videos are used except color images are used for the VTD method. The first metric is the success rate which is used in the PASCAL VOC challenge [53] defined as, $score = \frac{area(ROI_T \bigcap ROI_G)}{area(ROI_T \bigcup ROI_G)}$, where ROI_T is the tracking bounding box and ROI_G is the ground truth bounding box. If the $score$ is larger than 0.5 in one frame, the tracking result is considered as a success. Table 2 shows the tracking results in terms of success rate. The other is the center location error which is defined as the Euclidean distance between the central locations of the tracked objects and the manually labeled ground truth. Table 3 shows the average tracking errors of all methods. The proposed SFCT and FCT algorithms achieve the best or second best results in most sequences based on both success rate and center location error. Furthermore, the proposed trackers run faster than all the other algorithms except for the CST method which uses the fast Fourier transform. In addition, the SFCT algorithm performs better than the FCT algorithm for most sequences, and both achieve much better results than the CT algorithm in terms of both success rate and center location error, verifying the effectiveness of using scale invariant compressive features.

5.3 Tracking results

5.3.1 Pose and illumination change

For the *David indoor* sequence shown in Figure 8(a), the appearance changes gradually due to illumination and pose variation when the person walks out of the dark meeting room. The IVT, VTD, TLD, CT, FCT and SFCT algorithms perform well on this sequence. The IVT method uses a PCA-based appearance model which has been shown to account for appearance change caused by illumination variation well. The VTD method performs well due to the use of multiple observation models constructed from different features. The TLD approach works well because it maintains a detector which uses Haar-like features during tracking. In the *Sylvester* sequence shown in Figure 8(b), the object undergoes large pose and illumination change. The MIL, TLD, Struck, CST, ASLA, FCT and SFCT methods perform well on this sequence with lower tracking errors than other methods. The IVT, L1T, MTT, and DF methods do not perform well on this sequence as these methods use holistic features which are less effective for large scale pose variations. In Figure 8(c), the target object in the *Skating* sequence undergoes occlusion ($\#165$), shape deformation ($\#229, \#280$), and severe illumination change ($\#383$). Only the VTD, Struck, CT, FCT and SFCT methods perform well on this sequence. The VTD method performs well as



(a) Tracking results of the *David indoor* sequence

(b) Tracking results of the *Sylvester* sequence

(c) Tracking results of the *Skating* sequence

(d) Tracking results of the *Shaking 1* sequence

Fig. 8: Screenshots of some sample tracking results when there are pose variations and severe illumination changes.

it constructs multiple observation models which account for some different object appearance variations over time. The Struck method achieves low tracking errors as it maintains a fixed number of support vectors from the former frames which contain different aspects of the object appearance over time. However, the Struck method drifts away from the target after frame #350 in the *Skating* sequence due to several reasons. When the stage light changes drastically and the pose of the performer changes rapidly as she performs, only the VTD, CT, FCT and SFCT methods are able to track the object reliably. The proposed trackers are robust to pose and illumination changes as object appearance can be modeled well by random projections (based on the Johnson-Lindenstrauss lemma) and the classifier with online update is used to separate foreground and background samples. Furthermore, the features used in the proposed algorithms are similar to generalized Haar-like features which have been shown to be robust to pose and orientation change [10].

5.3.2 Occlusion and pose variation

The target object in the *Occluded face* sequence in Figure 9(a) undergoes in-plane pose variation and heavy occlusion. Overall, the MIL, L1T, Struck, CST, CT, FCT and SFCT algorithms perform well on this sequence. In the *Panda* sequence (Figure 9(b)), the target undergoes out-of-plane pose variation and shape deformation. Table 2 and Table 3 show that only the proposed SFCT method outperforms the other methods on this sequence in terms of success rate and center location error. The OAB and MIL methods work well on this sequence as they select the most discriminative Haar-like features for object representation which can well handle pose variation and shape deformation. Although the Struck method uses the Haar-like features to represent objects, no feature selection mechanism is employed and hence it is less effective in handling large pose variation and shape deformation. Due to the same reasons, the Struck method fails to track the target object stably in the *Bolt* sequence (Figure 9(c)). In the *Bolt* sequence, several objects appear in the same scene with rapid appearance change due to shape deformation and fast motion. Only the MIL, CST, CT, FCT and SFCT algorithms track the object stably. The CS, IVT, VTD, L1T, DF, MTT and ASLA methods do not perform well as generative models are less effective to account for appearance change caused by large shape deformation (e.g., background pixels are mistakenly considered as foreground ones), thereby making the algorithms drift away to similar objects. In the *Goat* sequence, the object undergoes pose variation, occlusion, and shape deformation. As shown in Figure 9(d), the proposed FCT and SFCT algorithms perform well, and the Struck as well as CST methods achieve relatively high success rate and low center location error. The CT algorithm fails to track the target after frame #100. In the *Pedestrian* sequence shown in Figure 9(e), the target object undergoes heavy occlusion (e.g., #50). In addition, it is challenging to track the target object due to low resolution.

(a) Tracking results of the *Occluded face*

(b) Tracking results of the *Panda* sequence

(c) Tracking results of the *Bolt* sequence

(d) Tracking results of the *Goat* sequence

(e) Tracking results of the *Pedestrian* sequence

Fig. 9: Screenshots of some sample tracking results when there are severe occlusion and pose variations.

Except the FCT and SFCT algorithms, all the other methods lose track of the target in numerous frames.

The proposed FCT and SFCT algorithms handle occlusion and pose variation well as the adopted scale invariant appearance model is discriminatively learned from target and background with data-independent measurement, thereby alleviating the influence of background pixels (See also Figure 9(c)). Furthermore, the FCT and SFCT algorithms perform well on objects with non-rigid shape deformation and camera view change in the *Panda*, *Bolt* and *Goat* sequences (Figure 9(b), (c), and (d)) as the adopted appearance model is based on spatially local scale invariant features which are less sensitive to non-rigid shape deformation.

5.3.3 Rotation and abrupt motion

The target object in the *Chasing* sequence (Figure 10(a)) undergoes abrupt movements with 360 degree out-of-plane rotation. The IVT, MTT, Struck, CST and CT methods perform well on this sequence. The CS method cannot handle scale changes well as illustrated by frames #430 and #530. The images of the *Shaking 2* sequence (Figure 10(b)) are blurry due to fast motion of the subject. The DF, MTT and SFCT methods achieve favorable performance on this sequence in terms of both success rate and center location error. However, the MTT method drifts away from the target object after frame #270. When the out-of-plane rotation and abrupt motion both occur in the *Tiger 1*, *Tiger 2* and *Biker* sequences (Figure 10(c), (d)), most algorithms fail to track the target objects well. The proposed SFCT and FCT algorithms outperform most of the other methods in most metrics (accuracy, success rate and speed). The *Twinings* and *Animal* sequences contain objects undergoing out-of-plane rotation and abrupt motion, respectively. Similarly, the proposed trackers perform well in terms of all metrics.

Fig. 10: Screenshots of some sample tracking results when there are rotation and abrupt motion.

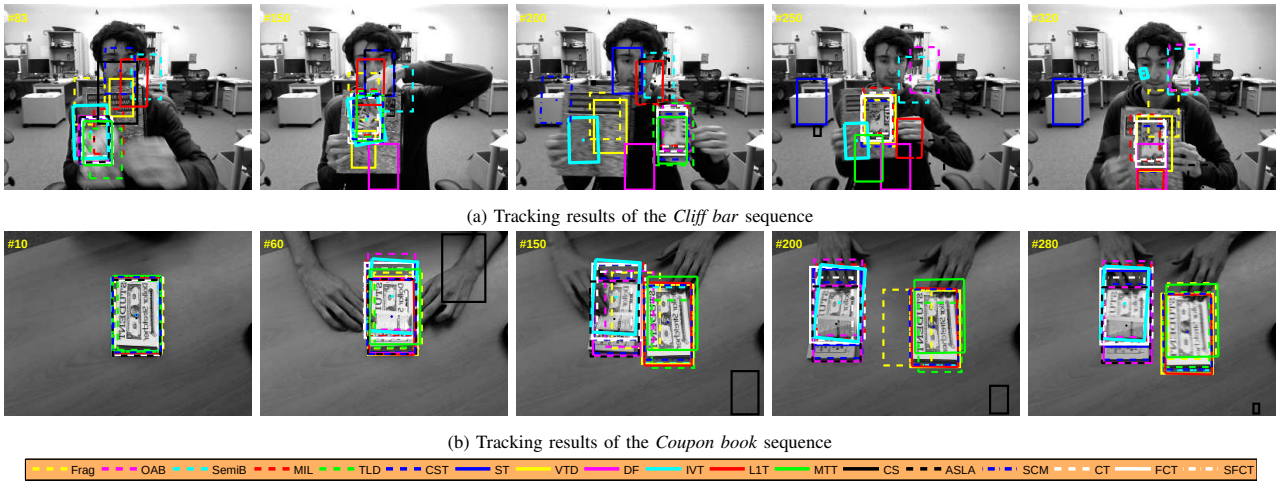


Fig. 11: Screenshots of some sample tracking results with background clutter.

5.3.4 Background clutter

The object in the *Cliff bar* sequence changes in scale, orientation and the surrounding background has similar texture (Figure 11(a)). As the Frag, IVT, VTD, L1T, DF, CS, MTT and ASLA methods use generative appearance models that do not exploit the background information, it is difficult to keep track of the objects correctly. The object in the *Coupon book* sequence undergoes significant appearance change at the 60-th frame and then the other coupon book appears in the scene. The CS method drifts to the background after frame #60. The Frag, SemiB, VTD, L1T, TLD, MTT and SCM methods drift away to track the other coupon book (#150, #200, #280 in

Figure 11(b)) while the proposed SFCT and FCT algorithms successfully track the correct one. The proposed algorithms are able to track the right objects accurately in these sequences as it extracts discriminative scale invariant features for the most "correct" positive sample (i.e., the target object) online with classifier update for foreground and background separation (See Figure 7).

6 Concluding Remarks

In this paper, we propose a simple yet robust tracking algorithm with an appearance model based on non-adaptive random projections that preserve the structure of original image space. A very sparse measurement matrix is adopted to efficiently compress features from the foreground targets and background ones. The tracking task is formulated as a binary classification problem with online update in the compressed domain. Numerous experiments with state-of-the-art algorithms on challenging sequences demonstrate that the proposed algorithm performs well in terms of accuracy, robustness, and speed.

Our future work will focus on applications of the developed algorithm for object detection and recognition under heavy occlusion. In addition, we will explore efficient detection modules for persistent tracking (where objects disappear and reappear after a long period of time).

Acknowledgements

We would like to thank valuable comments from the reviewers and associate editor. K. Zhang and L. Zhang are supported in part by the Hong Kong Polytechnic University ICRG Grant (G-YK79). M.-H. Yang is supported in part by the NSF CAREER Grant 1149783 and NSF IIS Grant 1152576.

References

[1] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Proceedings of European Conference on Computer Vision*, pp. 864–877, 2012.

[2] M. Black and A. Jepson, "Eigentracking: Robust matching and tracking of articulated objects using a view-based representation," *International Journal of Computer Vision*, vol. 26, no. 1, pp. 63–84, 1998.

[3] A. Jepson, D. Fleet, and T. El-Maraghi, "Robust online appearance models for visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1296–1311, 2003.

[4] S. Avidan, "Support vector tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 1064–1072, 2004.

[5] R. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1631–1643, 2005.

[6] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *Proceedings of British Machine Vision Conference*, pp. 47–56, 2006.

[7] D. Ross, J. Lim, R. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1, pp. 125–141, 2008.

[8] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *Proceedings of European Conference on Computer Vision*, pp. 234–247, 2008.

[9] J. Kwon and K. Lee, "Visual tracking decomposition.," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1269–1276, 2010.

[10] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1619–1632, 2011.

[11] H. Li, C. Shen, and Q. Shi, "Real-time visual tracking using compressive sensing," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1305–1312, 2011.

[12] X. Mei and H. Ling, "Robust visual tracking and vehicle classification via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2259–2272, 2011.

[13] S. Hare, A. Saffari, and P. Torr, "Struck: Structured output tracking with kernels," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 263–270, 2011.

[14] Y. Wu, J. Cheng, J. Wang, H. Lu, J. Wang, H. Ling, E. Blasch, and L. Bai, "Real-time probabilistic covariance tracking with efficient model update," *IEEE Transactions on Image Processing*, vol. 21, no. 5, pp. 2824–2837, 2012.

[15] Y. Wu, B. Shen, and H. Ling, "Visual tracking via online non-negative matrix factorization," *IEEE Transactions on Circuit and Systems for Video Technology*, no. 3, pp. 374–383, 2014.

[16] J. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proceedings of European Conference on Computer Vision*, pp. 702–715, 2012.

[17] J. Ho, K. Lee, M.-H. Yang, and D. Kriegman, "Visual tracking using learned linear subspaces," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. I–782, 2004.

[18] E. Candes and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.

[19] E. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?," *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[20] D. Achlioptas, "Database-friendly random projections: Johnson-lindenstrauss with binary coins," *Journal of Computer and System Sciences*, vol. 66, no. 4, pp. 671–687, 2003.

[21] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: applications to image and text data," in *International Conference on Knowledge Discovery and Data Mining*, pp. 245–250, 2001.

[22] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Computing Surveys*, vol. 38, no. 4, 2006.

[23] M.-H. Yang and J. Ho, "Toward robust online visual tracking," in *Distributed Video Sensor Networks* (B. Bhanu, C. Ravishankar, A. Roy-Chowdhury, H. Aghajan, and D. Terzopoulos, eds.), pp. 119–136, Springer, 2011.

[24] S. Salti, A. Cavallaro, and L. Di Stefano, "Adaptive appearance modeling for video tracking: Survey and evaluation," *IEEE Transactions on Image Processing*, vol. 21, no. 10, pp. 4334–4348, 2012.

[25] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 798–805, 2006.

[26] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via multi-task sparse learning," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2042–2049, 2012.

[27] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust l1 tracker using accelerated proximal gradient approach," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1830–1837, 2012.

[28] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan, "Locally orderless tracking," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1940–1947, 2012.

[29] L. Sevilla-Lara and E. Learned-Miller, "Distribution fields for tracking," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1910–1917, 2012.

[30] S. Wang, H. Lu, F. Yang, and M.-H. Yang, "Superpixel tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1323–1330, 2011.

[31] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, 2003.

[32] C. Shen, J. Kim, and H. Wang, "Generalized kernel-based visual tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 1, pp. 119–130, 2010.

[33] Z. Kalal, J. Matas, and K. Mikolajczyk, "Pn learning: Bootstrapping binary classifiers by structural constraints," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 49–56, 2010.

[34] D. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[35] R. Baraniuk, "Compressive sensing," *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 118–121, 2007.

[36] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, "A simple proof of the restricted isometry property for random matrices," *Constructive Approximation*, vol. 28, no. 3, pp. 253–263, 2008.
[37] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.
[38] L. Liu and P. Fieguth, "Texture classification from random features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 574–586, 2012.
[39] P. Li, T. Hastie, and K. Church, "Very sparse random projections," in *International Conference on Knowledge Discovery and Data Mining*, pp. 287–296, 2006.
[40] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
[41] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.
[42] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 511–518, 2001.
[43] S. Li and Z. Zhang, "Floatboost learning and statistical face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1112–1123, 2004.
[44] A. Ng and M. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes," in *Advances in Neural Information Processing Systems*, pp. 841–848, 2002.
[45] P. Diaconis and D. Freedman, "Asymptotics of graphical projection pursuit," *The Annals of Statistics*, pp. 793–815, 1984.
[46] K. Zhang and H. Song, "Real-time visual tracking via online weighted multiple instance learning," *Pattern Recognition*, vol. 46, no. 1, pp. 397–411, 2013.
[47] F. D. la Torre and M. J. Black, "Robust principal component analysis for computer vision," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 362–369, 2001.
[48] R. Raina, A. Battle, H. Lee, B. Packer, and A. Ng, "Self-taught learning: transfer learning from unlabeled data," in *International Conference on Machine Learning*, pp. 759–766, 2007.
[49] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.
[50] A. Leonardis and H. Bischof, "Robust recognition using eigenimages," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 99–118, 2000.
[51] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1838–1845, 2012.
[52] J. Xu, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1822–1829, 2012.
[53] M. Everingham, L. Gool, C. Williams, J. Winn, and A. Zisserman, "The pascal visual object class (voc)challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

Kaihua Zhang is a professor in the School of Information and Control, Nanjing University of Information Science & Technology, Nanjing, China. He received the B.S. degree in Technology and Science of Electronic Information from Ocean University of China (OUC) in 2006, the M.S. degree in Signal and Information Processing from the University of Science and Technology of China (USTC) in 2009 and Ph.D degree from the Department of Computing in the Hong Kong Polytechnic University in 2013. From Aug. 2009 to Aug. 2010, he worked as a Research Assistant in the Department of Computing, The Hong Kong Polytechnic University. His research interests include image segmentation, level sets, and visual tracking.

Lei Zhang received the B.Sc. degree in 1995 from Shenyang Institute of Aeronautical Engineering, Shenyang, P.R. China, the M.Sc. and Ph.D degrees in Control Theory and Engineering from Northwestern Polytechnical University, Xi'an, P.R. China, respectively in 1998 and 2001. From 2001 to 2002, he was a research associate in the Dept. of Computing, The Hong Kong Polytechnic University. From Jan. 2003 to Jan. 2006 he worked as a Postdoctoral Fellow in the Dept. of Electrical and Computer Engineering, McMaster University, Canada. In 2006, he joined the Dept. of Computing, The Hong Kong Polytechnic University, as an Assistant Professor. Since Sept. 2010, he has been an Associate Professor in the same department. His research interests include Image and Video Processing, Computer Vision, Pattern Recognition and Biometrics, etc. Dr. Zhang has published about 200 papers in those areas. Dr. Zhang is currently an Associate Editor of IEEE Trans. on CSVT and Image and Vision Computing. He was awarded the 2012-13 Faculty Award in Research and Scholarly Activities. More information can be found in his homepage http://www4.comp.polyu.edu.hk/~cslzhang/.

Ming-Hsuan Yang is an associate professor in Electrical Engineering and Computer Science at University of California, Merced. He received the PhD degree in computer science from the University of Illinois at Urbana-Champaign in 2000. Prior to joining UC Merced in 2008, he was a senior research scientist at the Honda Research Institute working on vision problems related to humanoid robots. He coauthored the book Face Detection and Gesture Recognition for Human-Computer Interaction (Kluwer Academic 2001) and edited special issue on face recognition for Computer Vision and Image Understanding in 2003, and a special issue on real world face recognition for IEEE Transactions on Pattern Analysis and Machine Intelligence. Yang served 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 International Journal of Computer Vision, Image and Vision Computing and Journal of Artificial Intelligence Research. He received the NSF CAREER award in 2012, the Senate Award for Distinguished Early Career Research at UC Merced in 2011, and the Google Faculty Award in 2009. He is a senior member of the IEEE and the ACM.