Robust Object Tracking via Sparse Collaborative Appearance Model

Wei Zhong, Huchuan Lu, *Senior Member, IEEE*, and Ming-Hsuan Yang, *Senior Member, IEEE*

***Abstract*— In this paper, we propose a robust object tracking algorithm based on a sparse collaborative model that exploits both holistic templates and local representations to account for drastic appearance changes. Within the proposed collaborative appearance model, we develop a sparse discriminative classifier (SDC) and sparse generative model (SGM) for object tracking. In the SDC module, we present a classifier that separates the foreground object from the background based on holistic templates. In the SGM module, we propose a histogram-based method that takes the spatial information of each local patch into consideration. The update scheme considers both the most recent observations and original templates, thereby enabling the proposed algorithm to deal with appearance changes effectively and alleviate the tracking drift problem. Numerous experiments on various challenging videos demonstrate that the proposed tracker performs favorably against several state-of-the-art algorithms.**

***Index Terms*— Object tracking, collaborative model, sparse representation, feature selection, occlusion handling.**

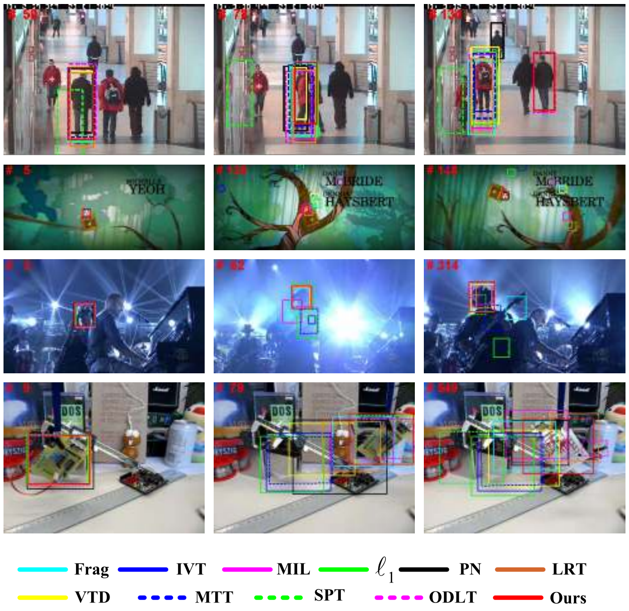

Fig. 1. Challenging factors for object tracking: heavy occlusions (*caviar*), rotation (*panda*), illumination changes (*shaking*) and cluttered background (*board*).

I. Introduction

The goal of object tracking is to estimate the states of a target object in an image sequence. It plays a critical role in numerous vision applications such as motion analysis, activity recognition, visual surveillance and intelligent user interfaces. While much progress has been made in recent years, it is still a challenging problem to develop a robust algorithm for complex and dynamic scenes due to large appearance changes caused by varying illumination, camera motion, occlusions, pose variation and shape deformation (See Fig. 1).

For visual tracking, an appearance model is used to represent the target object and verify predictions in each frame. A motion model is applied to predict the likely states of an object (e.g., Kalman filter [1] and particle filter [2], [3]). In this paper, we mainly focus on the appearance model since it is usually the most crucial component of any tracking algorithm.

Several factors need to be considered for an effective appearance model. First, an object can be represented by different features such as intensity [4], color [2], texture [5], superpixels [6], and Haar-like features [7]–[10]. Meanwhile, the representation schemes can be based on holistic templates [1], [11], [12] or local histograms [13], [14]. In this work, we use intensity values to represent objects due to its simplicity and efficiency. Furthermore, our approach exploits the strength of holistic templates to distinguish the target from the background, and the effectiveness of local patches in handling partial occlusion.

Second, a generative or discriminative appearance model is needed to effectively verify state predictions. For generative methods, tracking is formulated as searching for the most similar region to the target object within a neighborhood [1], [4], [13], [15]–[17]. For discriminative approaches, tracking is posed as a binary classification problem which aims to design a classifier for distinguishing the target object from the background [5], [7]–[10], [18]–[20]. In addition, several algorithms have been proposed to exploit the advantages of both

Manuscript received June 5, 2013; revised November 24, 2013; accepted January 23, 2014. Date of publication March 24, 2014; date of current version April 22, 2014. The work of W. Zhong and H. Lu was supported in part by the Natural Science Foundation of China under Grants 61071209 and 61272372, and in part by Joint Foundation of China Education Ministry and China Mobile Communication Corporation under Grant MCM20122071. The work of M.-H. Yang was supported in part by the National Science Foundation (NSF) CAREER under Grant 1149783 and in part by NSF Information and Intelligent Systems under Grant 1152576. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Nikolaos V. Boulgouris.

W. Zhong and H. Lu are with the School of Information and Communication Engineering, Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116024, China (e-mail: zhongwei.dut@gmail.com; lhchuan@dlut.edu.cn).

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

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/TIP.2014.2313227

generative and discriminative models [21]–[24]. We develop a simple yet robust collaborative model that makes use of the generative model to account for appearance changes and the discriminative classifier to effectively separate the foreground target from the background.

The third issue is concerned with online update schemes of tracking algorithms to account for appearance variations of the target object and the background. Although numerous update approaches have been proposed [1], [4], [5], [7], [15], straightforward and frequent updates of tracking results may gradually result in drifts due to accumulated errors, especially when occlusion occurs. To address this problem, Babenko *et al.* [9] propose an online boosting algorithm within the multiple instance learning (MIL) framework to resolve ambiguities of object locations and thereby reduce tracking drifts. Kalal *et al.* [10] develop a bootstrapping classifier in which the structure of unlabeled data is exploited via positive and negative constraints to select potential samples for update. To capture appearance variations and reduce tracking drifts, we propose a method that takes occlusions into consideration for appearance update.

In this paper, we propose a robust object tracking algorithm with an effective and adaptive appearance model.[1] Within the proposed tracking algorithm, the collaboration of the generative model and the discriminative classifier leads to a more flexible and robust likelihood function to verify the state predictions. The proposed model is adaptively updated with consideration of occlusions to account for appearance variations and alleviate drifts. Numerous experiments on various challenging sequences show that the proposed algorithm performs favorably against the state-of-the-art methods.

II. Related Work and Context

There is a rich literature in object tracking and here we discuss the most related work and put the proposed algorithm within proper context (See [26], [27] for recent surveys).

Sparse representation has recently been applied to visual tracking [15]–[17], [20]. Mei and Ling [15] present a visual tracking algorithm based on a generative sparse representation of templates. In spite of demonstrated success, there are still several issues to be addressed. First, this tracking method handles occlusion via ℓ_1 minimization of target and trivial templates with a particle filter at the expense of high computational cost. Second, the trivial templates can be used to model any image region from the target object or the background. Therefore, the reconstruction errors of image regions from the occluded target object and the background may be both small. As the sample with minimal reconstruction error is regarded as the target location, ambiguities are likely to accumulate and cause tracking failure.

Liu *et al.* [16] propose a tracking method which selects a sparse and discriminative set of features to improve efficiency and robustness. As the number of discriminative features is fixed, this method is less effective for object tracking in dynamic and complex scenes. In [17], an algorithm based on histograms of local sparse representation for object tracking is proposed where the target object is located via mode seeking (using the mean shift algorithm) of voting maps constructed by reconstruction errors. That is, this algorithm operates under the premise that the most likely target object location has minimal reconstruction error based on sparse representation. In contrast to the generative approaches based on sparse representation [15], [17] which do not differentiate foreground patches from the background ones, we propose a weighting method to ensure that occluded patches are not used to account for appearance changes of the target object, thereby resulting a more robust model. Furthermore, geometric information between local patches has not been well exploited [15], [17] whereas the proposed algorithm exploits both local histograms and a holistic template set to encode structural information.

Occlusion is one of the most challenging problems in object tracking. Adam *et al.* [13] propose a fragments-based method to handle occlusions where the target object is located by a voting map formed by comparing histograms of the candidate patches and the corresponding templates. However, the template is not updated and thus this approach is sensitive to large appearance variations. We develop an effective method which estimates and rejects possible occluded patches to improve robustness of the proposed appearance model when occlusions occur. In addition, the proposed model is adaptively updated with consideration of the occlusion rate to better account for appearance changes.

III. Proposed Tracking Algorithm

Visual tracking has been commonly formulated within the Bayesian filtering framework in which the goal is to determine a posteriori probability, $p(\mathbf{x}_t|\mathbf{z}_{1:t})$, of the target state by

$$p(\mathbf{x}_t|\mathbf{z}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1})d\mathbf{x}_{t-1}, \quad (1)$$

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}) \propto p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{z}_{1:t-1}), \quad (2)$$

where $\mathbf{x}_t$ is the object state, and $\mathbf{z}_t$ is the observation at time t. Let $\mathbf{x}_t = \left[l_x, l_y, \theta, s, \alpha, \phi\right]^\top$, where $l_x, l_y, \theta, s, \alpha, \phi$ denote x, y translations, rotation angle, scale, aspect ratio, and skew respectively. We assume that the affine parameters are independent and modeled by six scalar Gaussian distributions. The motion model $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ predicts the state at t based on the immediate previous state, and the observation model $p(\mathbf{z}_t|\mathbf{x}_t)$ describes the likelihood of observing $\mathbf{z}_t$ at state $\mathbf{x}_t$. The particle filter is an effective realization of Bayesian filtering, which predicts the state regardless of the underlying distribution. The optimal state is obtained by the maximum a posteriori estimation (MAP) over a set of N samples,

$$\hat{\mathbf{x}}_t = \arg_{\mathbf{x}_t^i} \max p(\mathbf{z}_t|\mathbf{x}_t^i)p(\mathbf{x}_t^i|\mathbf{x}_{t-1}), \quad (3)$$

where $\mathbf{x}_t^i$ is the i-th sample at frame t. In the next two sections, we present a tracking algorithm within the particle filter framework. We improve the motion model $p(\mathbf{x}_t^i|\mathbf{x}_{t-1})$ as an efficient two-step particle filter, and we present an effective and robust observation model $p(\mathbf{z}_t|\mathbf{x}_t^i)$ based on the collaboration of discriminative and generative models.

[1] Preliminary results of this work are presented in [25].

IV. Motion Model

Using a particle filter, the samples at frame t can be drawn by a Gaussian function with mean $\mathbf{x}_{t-1}$ and variance σ^2:

$$p(\mathbf{x}_t^i|\mathbf{x}_{t-1}) = G(\mathbf{x}_{t-1}, \sigma^2). \tag{4}$$

The use of more samples is likely to improve the tracking robustness at the expense of increasing computational cost. We note the tracking result is the MAP estimation over the samples which can be modeled well with mode seeking, and propose a motion model:

$$p(\mathbf{x}_t^i|\mathbf{x}_{t-1}) = w_t^i G(\mathbf{x}_{t-1}, \sigma^2), \tag{5}$$

where w_t^i is the weight of the i-th sample at frame t computed by the corresponding sparse coefficients using the template set.

At time t, the sample set $\mathbf{X}=\left\{\mathbf{x}_\mathrm{t}^\mathrm{i}\right\}_{\mathrm{i}=1,\ldots,\mathrm{N}}$ is obtained by the Gaussian function using Eq. 4. The template set $\mathbf{T}=\{\mathbf{t}^\mathrm{j}\}_{\mathrm{j}=1,\ldots,\mathrm{m}}$ is composed of $m-1$ tracking results in the latest $m-1$ frames and the template in the first frame. Given the sample set $\mathbf{X}$, the sparse coefficients γ_j of each $\mathbf{t}^j$ of the template set $\mathbf{T}$ are computed by

$$\min_{\boldsymbol{\gamma}_j} \|\mathbf{t}^j - \mathbf{X}\boldsymbol{\gamma}_j\|_2^2 + \lambda_1\|\boldsymbol{\gamma}_j\|_1, \quad \text{s.t. } \boldsymbol{\gamma}_j \succeq 0, \quad j = 1, \ldots, m, \tag{6}$$

where each column of $\mathbf{X}$ is a sample at time t and λ_1 is a weight parameter. The sample set $\mathbf{X}$ forms an over-complete dictionary, and the sparsity constraints force to select the samples that are highly correlated to the templates. That is, the samples that do not model the templates well are not considered as good candidates for the tracking results. We note that this formulation is different from the ℓ_1 tracking method [15] which requires solving N ℓ_1-minimization problems. In contrast, the proposed method requires solving m ℓ_1-minimization problems ($m \ll N$), thereby reducing the computational complexity significantly.

A constraint term, $\boldsymbol{\gamma}_j \succeq 0$, is introduced to make sure the sparse coefficients are nonnegative. In this context, their amplitudes reflect the similarity between the templates and the samples in terms of appearance. For example, if $\boldsymbol{\gamma}_{ji}$=0 ($\boldsymbol{\gamma}_{ji}$ is the i-th element of the vector $\boldsymbol{\gamma}_j$), it indicates the sample $\mathbf{x}^i$ is quite different from $\mathbf{t}^j$, and unlikely to be the tracking result. If $\sum\limits_j \boldsymbol{\gamma}_{ji}$=0, it indicates the sample $\mathbf{x}^i$ is not similar with all the templates. Thus, we set the weight w_t^i by

$$w_t^i = \begin{cases} 0, & \text{if } \sum\limits_j \boldsymbol{\gamma}_{ji}=0 \\ 1, & \text{otherwise.} \end{cases} \tag{7}$$

In this motion model, the samples obtained by Eq. 6 form a sample set $\mathbf{X}' \in \mathbb{R}^{K\times n}$, which is a subset of the sample set $\mathbf{X} \in \mathbb{R}^{K\times N}$. As illustrated in Section VI-C, the number of particles n obtained by Eq. 6 is less than the original number N. As the number of samples is much smaller, this weighting scheme facilitates the tracking speed without losing accuracy.

V. Observation Model

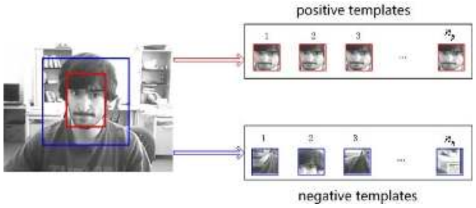

Fig. 2. Positive and negative templates in the SDC module.

Most tracking methods use rectangular image regions to represent targets, and thus background pixels are inevitably included as part of the foreground objects. Consequently, classifiers based on local representations may be significantly affected when background patches are considered as positive ones for update. On the other hand, the holistic appearance encoded by a target template is more distinct than the local appearance of local patches. Thus, holistic templates are more effective for discriminative models to separate foreground objects from the background. In addition, local representations are more amenable for generative models because of flexibility. In this work, we present a collaborative observation model that integrates a discriminative classifier based on holistic templates and a generative model using local representations.

A. Sparse Discriminative Classifier (SDC)

Motivated by the demonstrated success of sparse representation for vision tasks [14]–[17], [20], [28], [29], we propose a sparse discriminative classifier for object tracking. In the following, we use the vector $\mathbf{x}$ to represent intensity values of a raster scanned image.

1) Templates: The training image set is composed of n_p positive templates and n_n negative templates. Initially, we draw n_p sample images around the target location (e.g., within a radius of a few pixels) and downsample the selected images to a canonical size (32 × 32 in our experiments) with the standard bilinear interpolation filter for efficiency. Each downsampled image is stacked together to form the set of positive templates (See Fig. 2). Similarly, the negative training set is composed of images further away from the target location (e.g., within an annular region some pixels away from the target object as shown in Fig. 2). Thus, the negative training set consists of both the background and images with parts of the target object. This facilitates better object localization as samples containing only partial appearance of the target are treated as the negative samples and the corresponding confidence values are likely to be small.

2) Feature Selection: The above-mentioned gray-scale feature space is rich yet redundant, from which determinative ones that best distinguish the foreground object from the background can be extracted by learning a classifier,

$$\min_{\mathbf{s}} \left\|\mathbf{A}^\top\mathbf{s} - \mathbf{p}\right\|_2^2 + \lambda_2\|\mathbf{s}\|_1, \tag{8}$$

where $\mathbf{A} \in \mathbb{R}^{K\times(n_p+n_n)}$ is composed of n_p positive templates $\mathbf{A}_+$ and n_n negative templates $\mathbf{A}_-$, K is the dimension of the features, and λ_2 is a weight parameter. Each element of the vector $\mathbf{p} \in \mathbb{R}^{(n_p+n_n)\times 1}$ represents the property of each template in the training set $\mathbf{A}$, i.e., $+1$ for positive templates and -1 for negative templates.

The solution of Eq. 8 is the sparse vector $\mathbf{s}$, whose nonzero elements correspond to discriminative features selected from the K-dimensional feature space. The feature selection scheme adaptively chooses suitable number of discriminative features in dynamic environments via the ℓ_1 constraints. We project the features to a subspace via a projection matrix $\mathbf{S}$ which is formed by removing all-zero rows from a diagonal matrix $\mathbf{S}'$ and the elements are determined by

$$S'_{ii} = \begin{cases} 0, & s_i = 0 \\ 1, & \text{otherwise.} \end{cases} \tag{9}$$

Both the training template set and the candidates sampled by a particle filter are projected to the discriminative feature space. Thus, the training template set and candidates in the projected space are $\mathbf{A}' = \mathbf{SA}$ and $\mathbf{x}' = \mathbf{Sx}$.

3) Confidence Measure: The proposed SDC method is developed based on the assumption that a target image region can be better represented by the sparse combination of positive templates while a background patch can be better represented by the span of negative templates. Given a candidate region $\mathbf{x}$, it is represented by the training template set with the coefficients $\boldsymbol{\alpha}$ computed by

$$\min_{\boldsymbol{\alpha}} \left\|\mathbf{x}' - \mathbf{A}'\boldsymbol{\alpha}\right\|_2^2 + \lambda_3\|\boldsymbol{\alpha}\|_1, \tag{10}$$

where $\mathbf{x}'$ is the projected vector of $\mathbf{x}$ and λ_3 is a weight parameter.

A candidate region with smaller reconstruction error using the foreground template set indicates it is more likely to be a target object, and vice versa. Thus, we formulate the confidence value H_c of the candidate $\mathbf{x}$ by

$$H_c = \frac{1}{1+\exp\left(-\left(\varepsilon_b - \varepsilon_f\right)/\sigma\right)}, \tag{11}$$

where $\varepsilon_f = \left\|\mathbf{x}' - \mathbf{A}'_+\boldsymbol{\alpha}'_+\right\|_2^2$ is the reconstruction error of the candidate $\mathbf{x}$ with the foreground template set $\mathbf{A}_+$, and $\boldsymbol{\alpha}_+$ is the corresponding sparse coefficient vector. Similarly, $\varepsilon_b = \left\|\mathbf{x}' - \mathbf{A}'_-\boldsymbol{\alpha}'_-\right\|_2^2$ is the reconstruction error of the candidate $\mathbf{x}$ using the background template set $\mathbf{A}_-$, and $\boldsymbol{\alpha}_-$ is the corresponding sparse coefficient vector. The variable σ is fixed to be a small constant that balances the weight of the discriminative classifier and the generative model presented in Section V-B.

In [30], the reconstruction error is computed based on the target (positive) templates, which is less effective for tracking since both the negative and indistinguishable samples (e.g., patches covering some part of a target object) have large reconstruction errors when computed with the target (positive) template set. Thus, it introduces ambiguities in differentiating whether such patches are from the foreground or background. In contrast, our confidence measure exploits the distinct properties of the foreground and the background in computing the reconstruction errors to better distinguish patches from the positive and negative classes.

B. Sparse Generative Model (SGM)

Motivated by recent advances of sparse coding for image classification [31]–[33] as well as object tracking [17], we present a generative model for object representation that takes local appearance information of patches and occlusions into consideration.

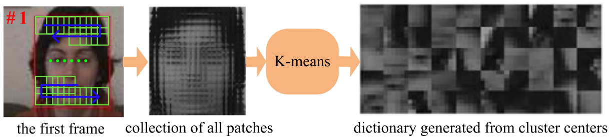
#1
K-means
the first frame
collection of all patches
dictionary generated from cluster centers

Fig. 3. We scan the first frame with overlapped sliding windows. Then the dictionary is generated with cluster centers of all the collected patches.

1) Histogram Generation: For simplicity, we use the gray-scale features to represent the local appearance information of a target object where each image is normalized to 32 × 32 pixels. We use overlapped sliding windows on the normalized images to obtain M patches and each patch is converted to a vector $\mathbf{y}_i \in \mathbb{R}^{G\times 1}$, where G denotes the size of the patch. The sparse coefficient vector $\boldsymbol{\beta}$ of each patch is computed by

$$\min_{\boldsymbol{\beta}_i} \left\|\mathbf{y}_i - \mathbf{D}\boldsymbol{\beta}_i\right\|_2^2 + \lambda_4 \left\|\boldsymbol{\beta}_i\right\|_1, \quad \text{s.t. } \boldsymbol{\beta}_i \succeq 0, \tag{12}$$

where the dictionary $\mathbf{D} \in \mathbb{R}^{G\times J}$ is generated from J cluster centers using the k-means algorithm on the M patches from the first frame (which consists of the most representative patterns of the target object) as Fig. 3, and λ_4 is a weight parameter.

In this work, the sparse coefficient vector $\boldsymbol{\beta}_i \in \mathbb{R}^{J\times 1}$ of each patch is normalized and concatenated to form a histogram by

$$\boldsymbol{\rho} = \left[\boldsymbol{\beta}_1^\top, \boldsymbol{\beta}_2^\top, \ldots, \boldsymbol{\beta}_M^\top\right]^\top, \tag{13}$$

where $\boldsymbol{\rho} \in \mathbb{R}^{(J\times M)\times 1}$ is the proposed histogram for one candidate region, as shown by Fig. 4.

2) Occlusion Handling: In order to deal with occlusions, we modify the constructed histogram to exclude the occluded patches when describing the target object. A patch with large reconstruction error is regarded as occluding part and the corresponding sparse coefficient vector is set to be zero. Thus, a weighted histogram is generated by

$$\boldsymbol{\varphi} = \boldsymbol{\rho} \odot \mathbf{o}, \tag{14}$$

where $\odot$ denotes the element-wise multiplication. Each element of $\mathbf{o}$ is an indicator of occlusion at the corresponding patch and is obtained by

$$o_i = \begin{cases} 1 & \varepsilon_i < \varepsilon_0 \\ 0 & \text{otherwise,} \end{cases} \tag{15}$$

where $\varepsilon_i = \left\|\mathbf{y}_i - \mathbf{D}\boldsymbol{\beta}_i\right\|_2^2$ is the reconstruction error of patch $\mathbf{y}_i$, and ε_0 is a predefined threshold which determines whether the patch is occluded or not. We thus have a sparse histogram $\boldsymbol{\varphi}$ for each candidate region. The proposed representation scheme takes spatial information of local patches and occlusion into account, thereby making it more effective and robust.

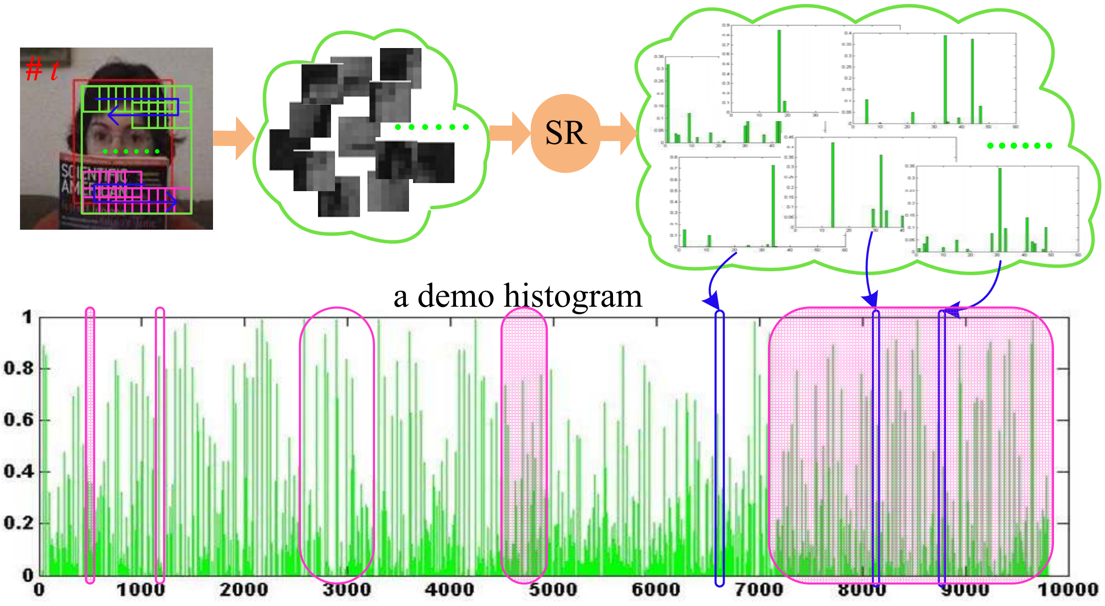

Fig. 4. We scan a candidate region in the t-th frame with overlapped sliding windows. The sparse coefficient vectors of all the patches are concatenated to form the histogram of this candidate region. The histogram segments in magenta are coefficient vectors of the occluded patches. These segments and their counterparts in the template histogram are not taken into account when computing the similarity of this histogram and the template histogram.

3) Similarity Function: We use the histogram intersection function to compute the similarity of histograms between the candidate and the template due to its effectiveness [33] by

$$L_c = \sum\nolimits_{j=1}^{J\times M} \min\left(\boldsymbol{\varphi}_c^j, \boldsymbol{\psi}^j\right), \tag{16}$$

where $\boldsymbol{\varphi}_c$ and $\boldsymbol{\psi}$ are the histograms for the c-th candidate and the template. The histogram of the template $\boldsymbol{\psi}$ is generated by Eqs. 12-14. The patches $\mathbf{D}$ in Eq. 12 are all from the first frame and the template histogram is computed only once for each image sequence. It is updated every several frames and the update scheme is presented in Section V-D.

The vector $\mathbf{o}$ in Eq. 15 reflects the occlusion condition of the corresponding candidate. The comparison between the candidate and the template should be carried out under the same occlusion condition, so the template and the c-th candidate share the same vector $\mathbf{o}_c$ in Eq. 14. For example, when the template is compared with the c-th candidate, the vector $\mathbf{o}$ of the template in Eq. 14 is set to $\mathbf{o}_c$.

C. Collaborative Model

We propose a collaborative model using the SDC and the SGM modules within the particle filter framework. In our tracking algorithm, both the confidence value based on the holistic templates and the similarity measure based on the local patches contribute to an effective and robust probabilistic appearance model. The likelihood function of the c-th candidate region is computed by

$$\begin{aligned} p\left(\mathbf{z}_t \left| \mathbf{x}_t^c\right.\right) &= H_c L_c \\ &= \left(\sum\nolimits_{j=1}^{J\times M} \min(\boldsymbol{\varphi}_{\mathrm{c}}^{\mathrm{j}}, \boldsymbol{\psi}^{\mathrm{j}})\right) \Big/ (1+\exp(-(\varepsilon_b-\varepsilon_f)/\sigma)), \end{aligned} \tag{17}$$

and each tracking result is the candidate with the maximum a posteriori estimation using Eq. 3.

The multiplicative formula is more effective in our tracking scheme compared with the alternative additive operation. The confidence value H_c gives higher weights to the candidates considered as positive samples (i.e., ε_f smaller than ε_b) and penalizes the others. As a result, it can be considered as the weight of the local similarity measure L_c.

D. Update Scheme

Since the appearance of an object often changes significantly during the tracking process, the update scheme is important and necessary. We develop an update scheme in which the SDC and SGM modules are updated independently.

For the SDC module, we update the negative templates every several frames (5 in our experiments) from image regions away (e.g., more than 8 pixels) the current tracking result. The positive templates remain the same in the tracking process. As the SDC module aims to distinguish the foreground from the background, it is important to ensure that the positive and negative templates are all correct and distinct.

For the SGM module, the dictionary $\mathbf{D}$ is fixed during the tracking process. Therefore, the dictionary is not incorrectly updated due to tracking failures or occlusions. In order to capture the appearance changes and recover the object from occlusions, the new template histogram $\boldsymbol{\psi}_n$ is computed by

$$\boldsymbol{\psi}_n = \mu\boldsymbol{\psi}_f + (1-\mu)\,\boldsymbol{\psi}_l \quad \text{if } O_n < O_0, \tag{18}$$

where $\boldsymbol{\psi}_f$ is the histogram representing the manually set tracking result in the first frame and it is generated with the way shown in Fig. 4). The notion $\boldsymbol{\psi}_l$ is the histogram last stored before update, and μ is the weight. The variable O_n denotes the occlusion measure of the tracking result in the new frame. It is computed by the corresponding occlusion

TABLE I

EVALUATED IMAGE SEQUENCES

Sequence	# Frames	Challenging Factors
faceocc1	898	partial occlusion
faceocc2	819	heavy occlusion, in-plane rotation
caviar1	382	partial occlusion, scale changes
caviar2	500	heavy occlusion, scale changes
animal	71	motion blur, background clutter
jumping	313	motion blur
girl	501	out-of-plane rotation, partial occlusion
davidin300	462	out-of-plane rotation, illumination variation, scale changes
panda	241	in-plane rotation, heavy occlusion
sylv	1344	out-of-plane rotation, in-plane rotation
car4	659	illumination variation, scale changes
car11	393	illumination variation, background clutter, scale changes
singer1	321	illumination variation, scale changes
shaking	365	illumination variation, scale changes
board	598	background clutter, out-of-plane rotation, scale changes
stone	593	background clutter, partial occlusion

indication vector $\mathbf{o}_n$ (by Eq. 15) using

$$O_n = \frac{1}{J \times M} \sum\nolimits_{i=1}^{J \times M} \left(1 - o_n^i\right). \tag{19}$$

The appearance model is updated as long as the occlusion measure O_n in this frame is smaller than a predefined constant O_0. This update scheme preserves the first template $\boldsymbol{\psi}_f$ (which is usually correct [10], [23], [34]) and takes the most recent tracking result into account.

VI. EXPERIMENTAL RESULTS

To evaluate the performance of our tracker, we conduct experiments on sixteen challenging image sequences (fourteen are publicly available and two are from our own dataset). These sequences cover most challenging situations in object tracking: heavy occlusion, motion blur, in-plane and out-of-plane rotations, large illumination changes, scale variation as well as complex background (See Table I). With the same initial positions of the targets, we compare with nine state-of-the-art tracking algorithms including the Frag [13], IVT [4], MIL [9], ℓ_1 [15], PN [10], VTD [35], MTT [28], SPT [17], LRT [29] and ODLT [20] methods. We present some representative results in this section. The proposed algorithm is implemented in MATLAB and runs at 3 frames per second on a 3.4 GHz i7-2770M Core PC with 32GB memory. All the MATLAB source codes and datasets are available at http://faculty.ucmerced.edu/mhyang/project/scm.htm.

The parameters of the proposed tracking algorithm are fixed in all experiments. The numbers of positive templates n_p and negative templates n_n are 50 and 200 respectively. All the weight parameters of Eqs. 6, 10 and 12 are set to be 0.01, and the variable λ_2 in Eq. 8 is fixed to be 0.001. In all the experiments, the number of patch M is 196. The row number G and column number J of the dictionary $\mathbf{D}$ in Eq. 12 are 36 and 50. The threshold ε_0 in Eq. 15 is 0.04. The update rate μ is set to be 0.95, and the threshold O_0 in Eq. 18 is 0.8.

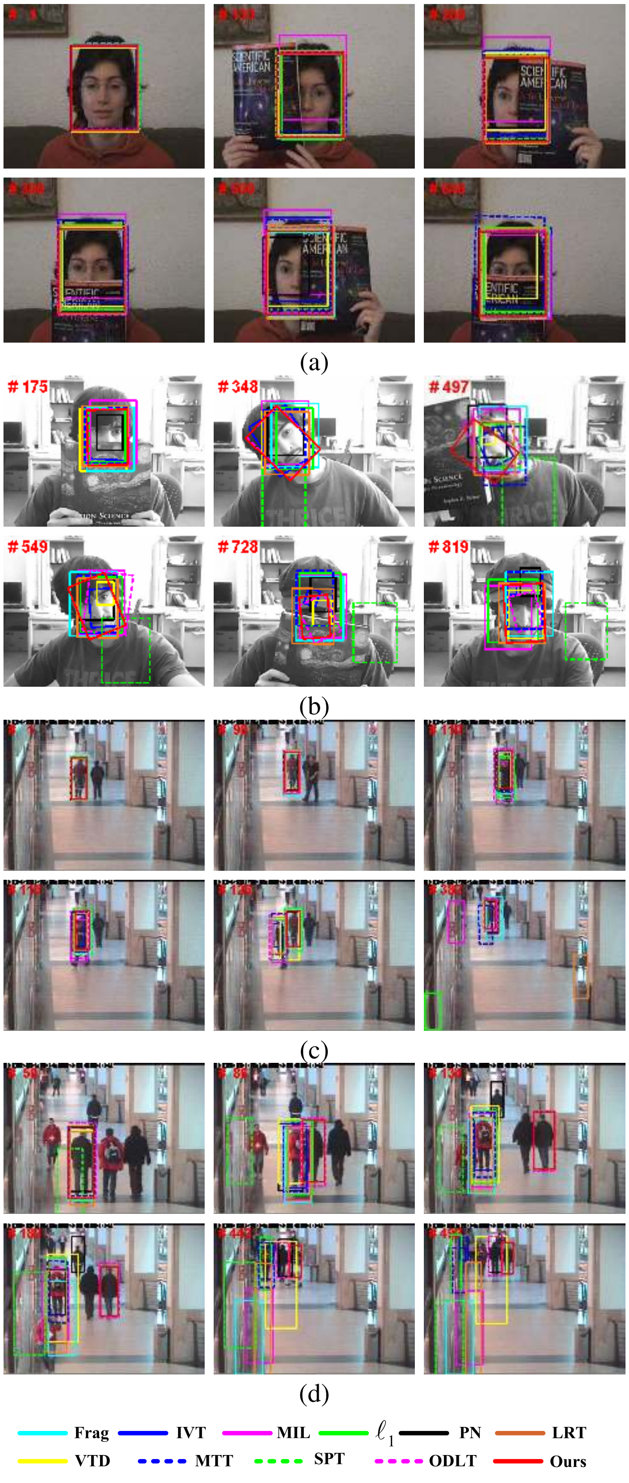

Fig. 5. Sample tracking results of evaluated algorithms on four image sequences with occlusions. (a) *faceocc1*, (b) *faceocc2*, (c) *caviar1*, and (d) *caviar2*.

A. Qualitative Evaluation

Heavy occlusion: Occlusion is one of the most general yet crucial issues in object tracking, and numerous tracking methods [9], [13], [15], [17], [20], [28] as well as the proposed algorithm are developed to deal with this problem. In the *faceoocc1* sequence, the woman occludes her face with a book frequently. As shown in Fig. 5(a), the Frag [13], ℓ_1 [15] and the

LRT [29] methods as well as the proposed algorithm perform better than the other trackers. The Frag method [13] uses the fragments-based histogram with a voting scheme to handle partial occlusions. On the other hand, the ℓ_1 tracking [15] and the LRT [29] methods respectively uses trivial templates and a sparse error matrix to model the occlusions. In the SGM module, we estimate the possible occluded patches and develop a robust histogram which only compares the patches that are not occluded. Thus, this scheme effectively alleviates the effects of occlusions. In the *faceocc2* sequence, the LRT method [29] and the proposed tracker perform better although the face is heavily occluded [frame 175 and 728 of Fig. 5(b)] with in-plane rotations [frame 348 and 497 of Fig. 5(b)]. The MIL tracking algorithm [9] locates the target well but deals with in-plane rotations less effectively. The ℓ_1 tracking method [15] updates the template set with a straightforward scheme and the tracking results are less accurate. In addition, the SPT approach [17] is less effective in dealing with in-plane rotations and drifts away when the man rotates his head (frame 348).

In the *caviar1* and *caviar2* sequences, the targets are occluded heavily and undergo scale changes. In addition, there are numerous objects with similar appearance (color and shape) to the targets. For most template-based trackers, simple update without dealing with occluded regions often leads to drifts [frame 125 in Fig. 5(c)]. In contrast, our tracker achieves stable performance in the entire sequences in spite of large scale changes and heavy occlusions. In the *caviar2* sequence, all the trackers, except the ODLT method [20] and our tracker, fail due to heavy occlusion [frame 134 in Fig. 5(d)]. This can be attributed to our SGM module that reduces the effects of occlusions and only compares the foreground with the stored histograms. Our update scheme does not update the appearance model with occluding patches, thereby alleviating the tracking drift problem.

Motion blur: Fast motion of the target object or the camera leads to blurred image appearance which is difficult to account for in object tracking. Fig. 6(a) shows tracking results on the *animal* sequence in which the object appearance is almost indistinguishable due to motion blurs. Most tracking algorithms fail to follow the target right at the beginning of this sequence. At frame 42, the PN [10] and SPT [17] methods mistakenly locate a similar object instead of the correct target. The reason is that the true target is blurred and it is difficult for the PN detector [10] to distinguish it from the background. The SPT method [17] uses only the foreground information and does not separate the target from the background well. Although the ODLT [20] algorithm uses the local sparse representation as the features to learn a classifier, there is no mechanism to differentiate each patch is from the foreground or the background before appearance updates. That is, discriminative tracking methods based only on local representations may not be effective when motion blurs or heavy occlusions occur. The proposed tracker well handles the situation with similar objects as the SDC module selects the discriminative features to better separate the target from the background. By updating the negative templates online, the proposed algorithm successfully tracks the target object throughout this sequence.

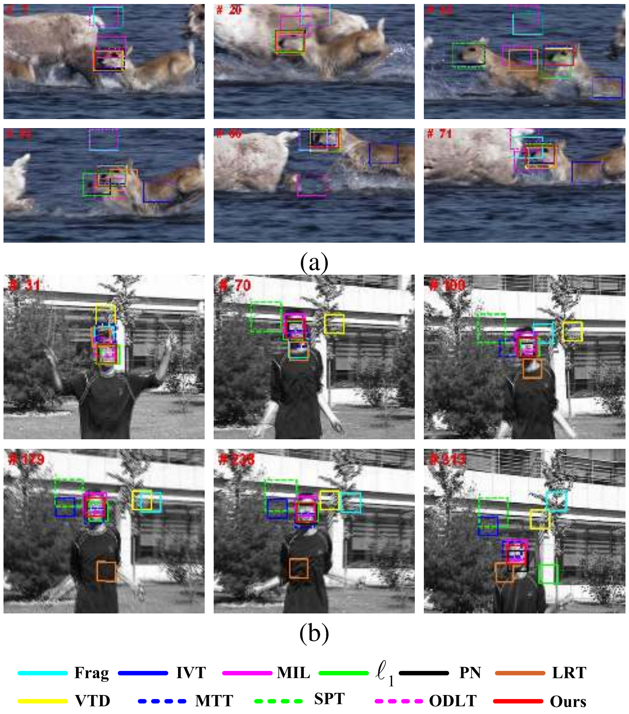

Fig. 6. Sample tracking results of evaluated algorithms on two image sequences with motion blur. (a) *animal* and (b) *jumping*.

The appearance changes caused by motion blurs in the *jumping* sequence [Fig. 6 (b)] are drastic such that the Frag [13] and VTD [35] methods fail before frame 31. The IVT [4] method is able to track the target in some frames (frame 100) but fails when the motion blur occurs (frame 238). Our tracker successfully keeps track of the target object with small errors. The main reason is that we use the SDC module to separate the foreground from the background. Meanwhile, the confidence measure computed by Eq. 11 assigns smaller weights to the candidate from the background. Therefore, the tracking drift problem is alleviated.

Rotation: The *girl* sequence shown in Fig. 7(a) consists of both in-plane and out-of-plane rotations. The PN [10] and VTD [35] methods fail when the girl rotates her head. Compared with other algorithms, our tracker is more robust and accurate (e.g., frame 312 and 430). In the proposed method, the background candidates are assigned with small weights according to Eq. 11, and our tracker does not drift to the background when the girl rotates (frame 111 and 312). In the *davidin300* sequence shown in Fig. 7(b), the target object undergoes out-of-plane rotations (frame 150) and illumination changes (frame 1 and frame 462). The IVT method [4] tracks the target well as the subspace learning method is robust to illumination changes and small pose variations. The LRT tracker [29] performs quite robustly to appearance and illumination changes in this case. This can be attributed to its low rank property and adaptively updated dictionary. The proposed algorithm tracks the target well as our SDC module reduces the weights of the background and increases the weight of the target object. In addition, the update scheme of the SGM module, which exploits the appearance of the first frame, facilitates the tracking process.

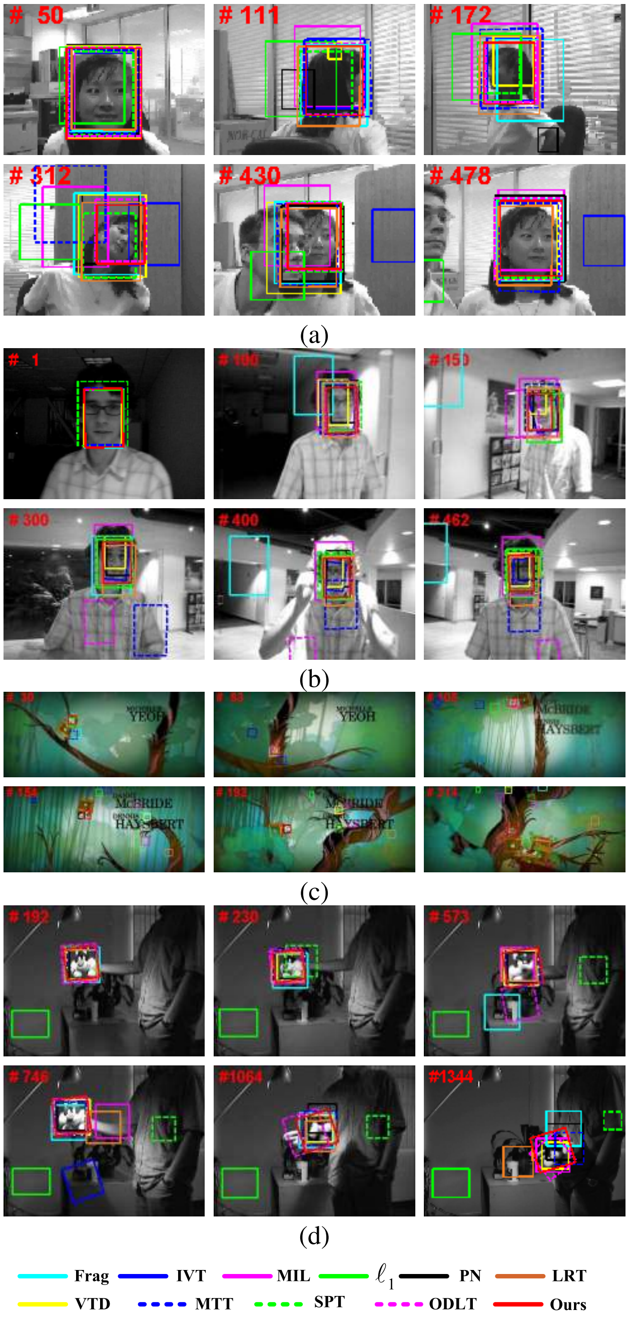

Fig. 7. Sample tracking results of evaluated algorithms on four image sequences with rotations. (a) *girl*, (b) *davidin300*, (c) *panda*, and (d) *sylv*.

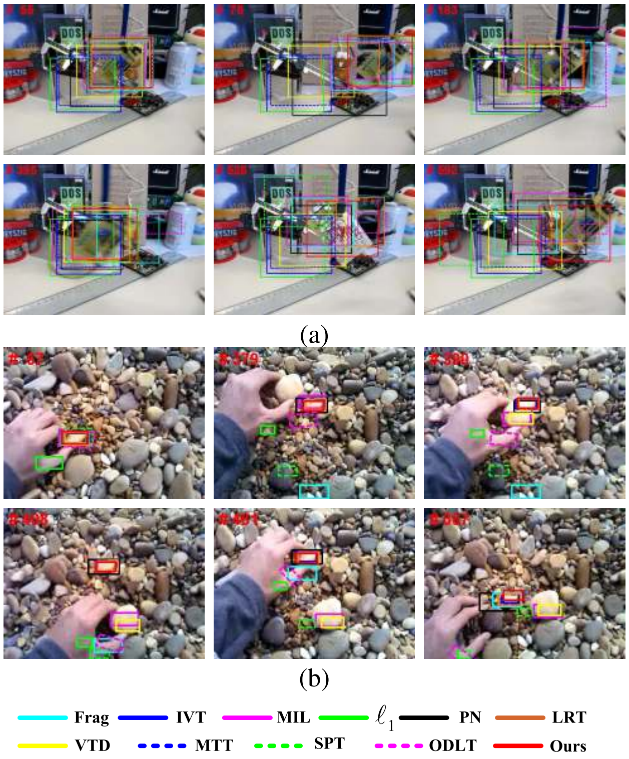

Fig. 8. Sample tracking results of evaluated algorithms on two image sequences with cluttered background. (a) *board* and (b) *stone*.

The target object in the *panda* sequence experiences more and larger in-plane rotations [Fig. 7(c)]. The IVT method [4] fails due to occlusion at frame 53 and fast movement. Most trackers drift after the target undergoes large rotations (frame 154) whereas our method performs well throughout this sequence. As the motion models of most tracking methods are based on translational or similarity transforms, it is difficult to account for complex movements. In contrast, the use of local histograms of the proposed algorithm helps in accounting for appearance changes due to complex motion. In addition, the target object in the *panda* sequence also undergoes occlusions as shown in frame 53 and frame 214. The PN tracking method [10] fails to detect occlusions and loses the target object after frame 214 while our tracker performs well. In the *sylv* sequence [Fig. 7(d)], the target object undergoes frequent in-plane and out-of-plane rotations. The SPT tracking method [17] does not deal with the rotation problems well and drifts away when there is out-of-plane rotation (frame 230). The Frag method [13] does not locate the target well when the target object undergoes large variations (frame 573, 1344). In contrast, our tracker performs well throughout this long sequence.

Complex background: The *board* sequence is challenging as the background is cluttered and the target object undergoes out-of-plane rotations as shown in Fig. 8(a). In frame 55, most trackers fail as holistic representations inevitably include background pixels that may be considered as part of the foreground object by straightforward update schemes. Using fixed templates, the Frag method [13] is able to track the target as long as there is no drastic appearance changes (frame 55 and 183), but fails when the target moves quickly or rotates (frame 78, 395 and 528). Our tracker performs well in this sequence as the target can be differentiated from the cluttered background by the SDC module. In addition, the update scheme uses the newly arrived negative templates that facilitate separation of the foreground object and the background.

The *stone* sequence consists of cluttered images where multiple objects in the background resemble the foreground target. The ℓ_1 tracking method [15] loses track of the target and instead locates one hand region which is similar to the target in terms of appearance. The MIL tracking algorithm [9] drifts

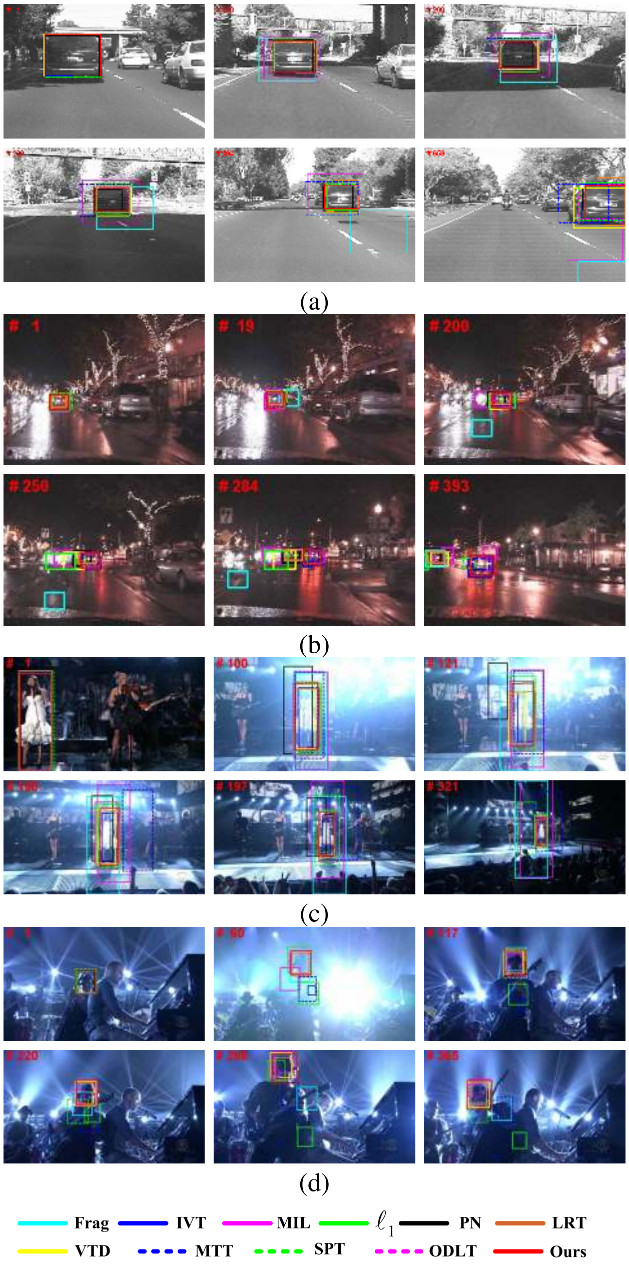

Fig. 9. Sample tracking results of evaluated algorithms on four image sequences with illumination changes. (a) *car4*, (b) *car11*, (c) *singer1*, and (d) *shaking*.

as the Haar-like features are less effective in distinguishing similar objects. The SPT method [17] does not perform well as only the foreground information is used to distinguish the target object from the background. In contrast, our tracker maintains a holistic template set that separates the target from its surroundings with the SDC module.

Illumination changes: Fig. 9 shows the tracking results on sequences with dramatic illumination changes. For the *car4* sequence, the target undergoes large illumination changes when the car passes the overpass (frame 160, 200 and 294). The Frag method [13] does not track the target well when large illumination changes occur as the template set without update is not effective in accounting for significant appearance variations. In addition, the Frag tracker [13] does not deal with scale change well. For the *car11* sequence, there is low contrast between the foreground and the background (frame 284) as well as illumination changes. The Frag method [13] fails at the beginning (frame 19) because it only uses the local information and does not maintain a holistic representation of the target. The IVT tracking method [4] performs well in this sequence which can be attributed to the fact that subspace learning methods are robust to illumination changes. As discriminative features are selected to separate the target from the background in the SDC module, the proposed tracking algorithm performs well in spite of the low contrast between the foreground and the background.

In the *singer1* sequence, the stage light changes drastically as shown in frame 121 and 321 of Fig. 9. The PN tracking method [10] does not track or re-detect the target object well when drastic lighting change occurs (frame 121). The MTT method [28] does not perform well in this sequence. This can be attributed to the fact that generative methods are less effective in differentiating regions with similar appearance to the target object when there is low contrast. On the other hand, the LRT method [29] and the proposed tracking algorithm accurately locates the target object even despite large changes in illumination and scale. In the *shaking* sequence, the target object undergoes large appearance variations due to drastic changes in illumination and motion. The low rank property of the LRT method [29] weaken the influence of nonuniform illumination so that the LRT tracker yields good performance. The proposed SDC module includes regions from the background and those that partially overlap with the target object as negative templates such the confidence values of these candidates computed by Eq. 11 are small. Thus, the proposed method is able to track the target object accurately.

B. Quantitative Evaluation

We evaluate the above-mentioned algorithms using the center location error and overlap ratio [36] based on the ground truth. The overlap ratio is computed by intersection over union based on the tracking result R_T and the ground truth R_G, i.e., $\frac{R_T \cap R_G}{R_T \cup R_G}$.

Figs. 10 and 11 show the center locations as well as the overlap ratios of the evaluated algorithms. For better readability, in each of the two figures, we only demonstrate curves in nine videos rather than the whole sixteen ones. However the selected videos in Figs. 10 and 11 are different so that for each video at least one of the overlap ratio curve and the center location curve is shown. Overall, the proposed algorithm performs well against the other state-of-the-art methods in the sixteen image sequences.

Tables II and III show the average center error and overlapping ratio where the red, blue and green fonts represent the top three tracking results. In Table II, the symbol – denotes that the PN tracking method [10] does not return tracking results in numerous frames which are discarded. We note that the PN method does not return tracking results for a significant number of frames in some sequences (e.g., the *shaking* sequence). Overall, our tracker achieves favorable results against other methods in terms of both metrics.

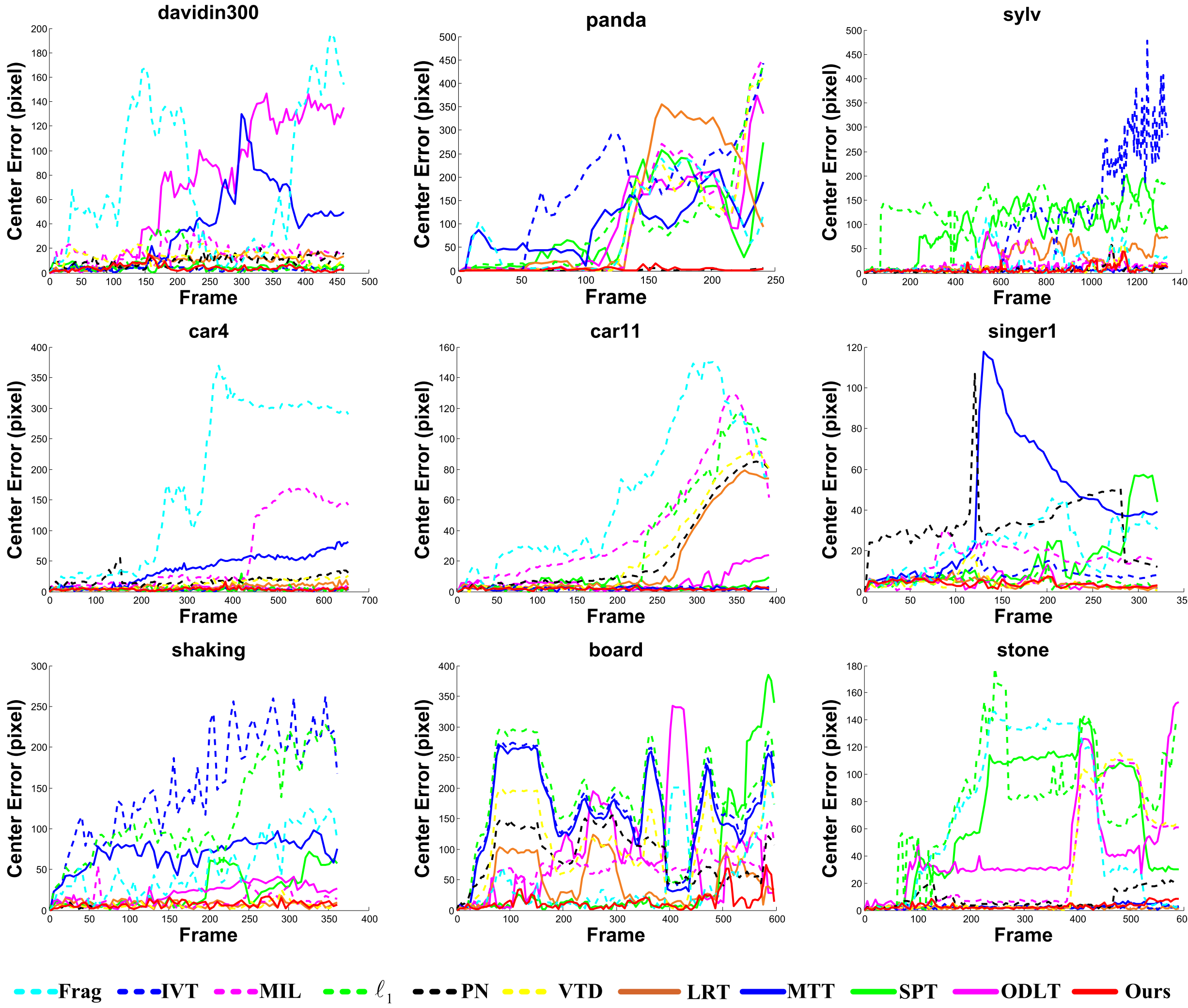

Fig. 10. Quantitative evaluation in terms of center location error (in pixel). The proposed algorithm is compared with nine state-of-the-art methods on nine challenging image sequences.

C. Discussion

SDC vs SGM: Since the proposed tracking algorithm consists of two collaborative modules, we demonstrate the merits of each one and how they complement each other. If the SDC module is used for tracking without the SGM module, the likelihood of Eq. 17 is $p\left(\mathbf{z}_t \left|\mathbf{x}_t^c\right.\right) = H_c$, and similarly if the SGM module is used without the SDC module, the likelihood is $p\left(\mathbf{z}_t \left|\mathbf{x}_t^c\right.\right) = L_c$. The tracking results based on either SDC or SGM are presented in Tables II and III.

In most cases, the collaborative model performs better than or equal to the SDC and SGM module individually although each one performs well against the state-of-the-art methods. As shown in Fig. 12 (and both Tables II and III), either the SDC or SGM based tracking method does not track the target object in the *panda* sequence well whereas the proposed algorithm with the collaborative model performs well. This can be attributed to that the collaborative model exploits the strength of both the SDC and SGM modules via Eq. 17.

In some cases (e.g., the *caviar1*, *davidin300*, *faceocc2*, *shaking* and *stone* sequences), the SDC-based tracking method is less effective but the one based on the SGM module performs well, and the proposed tracking algorithm based on the collaborative model achieves good results. The main reason is the SDC module is developed to separate the foreground object from the background and does not deal with occlusions robustly. In contrast, the local model is designed to account for appearance change due to occlusions and thus SGM-based tracking method performs well.

On the other hand, in some cases (e.g., the *caviar2* and *singer1* sequences), the SGM-based tracking method does not perform as well as the SDC-based algorithm, and the proposed tracking algorithm based on the collaborative model achieves good results. This can be attributed to the fact that the SGM module alone is not effective in explaining objects in cluttered background or with low contrast whereas the SDC module is designed to separate the foreground from the background. The *caviar2* sequence contains several objects with similar appearance in color and shape to the target. In the *singer1* sequence, the target object appears in the scenes with low contrast to the background and large scale change.

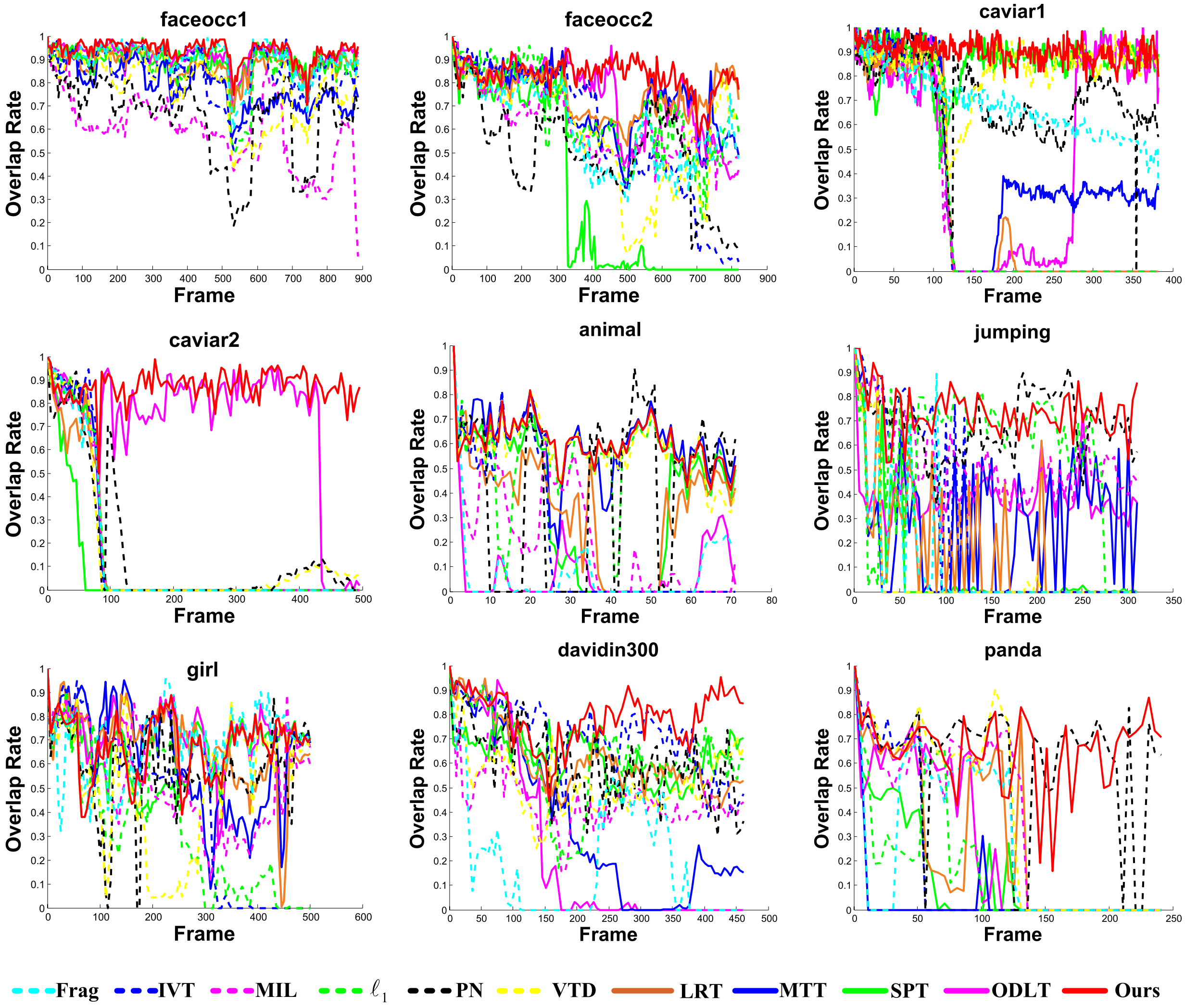

Fig. 11. Quantitative evaluation in terms of overlap ratio [36]. The proposed algorithm is compared with nine state-of-the-art methods on nine challenging image sequences.

TABLE II

Average Center Location Error (in Pixel): Top Three Results Are Shown in Red, Blue and Green Fonts (Symbol – Means That Frames That the PN Method [10] Loses the Target Are Not Counted)

	Frag	IVT	MIL	ℓ_1	PN	VTD	MTT	SPT	ODLT	LRT	Our	SDC	SGM
animal	90.9	127.6	65.8	14.8	24.0–	**12.0**	**9.4**	68.9	94.6	44.0	**9.9**	10.5	10.6
board	**45.4**	165.4	66.7	184.0	90.1	105.0	156.2	**51.9**	89.5	46.5	**13.9**	142.3	13.9
car4	179.8	**2.9**	60.2	4.1	18.8–	12.3	37.2	**3.3**	5.7	7.4	**3.8**	6.9	3.4
car11	64.0	**2.2**	43.5	33.3	25.2	27.1	**1.9**	4.1	6.2	20.8	**1.7**	18.8	1.8
caviar1	5.7	45.3	48.5	120.0	5.6	**4.0**	21.0	**1.8**	16.7	81.5	**1.0**	49.8	0.9
caviar2	115.5	65.7	99.6	65.4	**43.9**	57.9	65.3	130.6	**6.3**	96.7	**2.3**	2.6	5.0
davidin300	76.7	**3.6**	16.2	7.7	9.7	13.6	36.5	**4.9**	70.7	9.5	**4.0**	91.6	3.9
faceocc1	5.7	9.2	32.3	6.6	17.7	11.2	14.1	**5.4**	4.0	**4.7**	**3.3**	5.0	3.2
faceocc2	15.5	10.3	14.1	11.2	18.6	10.5	9.3	58.6	**7.0**	**3.3**	**4.5**	28.9	4.6
girl	18.1	48.5	32.3	62.5	23.2	21.5	23.9	**12.4**	**11.6**	19.1	**10.2**	12.1	9.9
jumping	59.4	37.6	**9.7**	12.8	**3.7**	63.4	21.0	55.8	12.8	48.9	**3.8**	6.7	4.0
shaking	52.1	153.7	11.4	118.0	**4.2**–	**5.9**	69.8	24.2	20.9	7.4	**7.2**	77.2	7.7
singer1	22.1	8.5	15.2	4.6	32.7	**4.1**	41.3	14.5	4.6	**4.2**	**3.8**	3.4	4.0
sylv	20.6	97.8	16.4	122.2	**5.9**	8.8	**6.4**	93.6	15.3	28.5	**7.1**	16.3	8.4
panda	**89.6**	169.6	105.4	95.5	**2.3**–	96.5	98.1	95.5	98.5	121.8	**3.0**	45.3	162.9
stone	65.9	**2.3**	32.4	77.5	8.0	31.4	**2.5**	68.8	42.2	**2.0**	3.1	78.5	2.4

Motion Model: As mentioned in Section IV, the proposed motion model (Eq. 5) is used to select n samples which is adaptively changed according to different scenes. We show the number of samples used by the proposed motion model with the *caviar1* sequence in Fig. 13. Although the number of samples N selected by Eq. 4 using a simple Gaussian model is

TABLE III

AVERAGE OVERLAP RATIO: TOP THREE RESULTS ARE SHOWN IN RED, BLUE AND GREEN

	Frag	IVT	MIL	ℓ_1	PN	VTD	MTT	SPT	ODLT	LRT	Our	SDC	SGM
animal	0.08	0.22	0.22	**0.54**	0.43	**0.57**	**0.60**	0.36	0.08	0.34	**0.60**	0.59	0.60
board	**0.65**	0.14	0.46	0.12	0.34	0.32	0.17	**0.63**	0.28	0.57	**0.78**	0.30	0.79
car4	0.22	**0.92**	0.34	0.84	0.63	0.73	0.52	**0.90**	0.85	0.76	**0.89**	0.85	0.89
car11	0.08	**0.80**	0.17	**0.43**	0.37	0.43	**0.58**	0.48	0.43	0.44	**0.79**	0.54	0.80
caviar1	0.68	0.27	0.25	0.27	0.70	**0.83**	0.45	**0.85**	0.50	0.28	**0.90**	0.26	0.90
caviar2	0.13	0.14	0.13	0.14	**0.16**	0.15	0.14	0.07	**0.71**	0.12	**0.87**	0.83	0.77
davidin300	0.19	**0.71**	0.44	**0.62**	0.60	0.52	0.35	**0.62**	0.25	**0.62**	**0.79**	0.25	0.80
faceocc1	**0.89**	0.84	0.59	0.87	0.64	0.77	0.78	**0.89**	**0.91**	0.88	**0.93**	0.90	0.93
faceocc2	0.60	0.58	0.61	0.67	0.49	0.59	0.71	0.33	**0.73**	**0.76**	**0.83**	0.44	0.83
girl	**0.68**	0.42	0.51	0.32	0.57	0.51	0.62	0.64	**0.71**	**0.68**	**0.67**	0.64	0.68
jumping	0.12	0.27	0.53	**0.56**	**0.69**	0.07	0.29	0.09	0.39	0.17	**0.74**	0.69	0.72
shaking	0.25	0.02	0.65	0.03	0.12	**0.74**	0.06	0.46	0.38	**0.67**	**0.72**	0.19	0.72
singer1	0.34	0.66	0.33	0.70	0.41	**0.79**	0.32	0.52	0.73	**0.80**	**0.84**	0.84	0.85
sylv	0.52	0.34	0.52	0.04	**0.72**	**0.67**	**0.72**	0.12	0.54	0.39	**0.70**	0.55	0.68
panda	0.24	0.16	0.35	0.15	**0.61**	**0.37**	0.03	0.13	0.25	0.26	**0.65**	0.50	0.35
stone	0.15	**0.65**	0.32	0.10	0.41	0.41	**0.62**	0.08	0.07	**0.65**	**0.60**	0.10	0.61

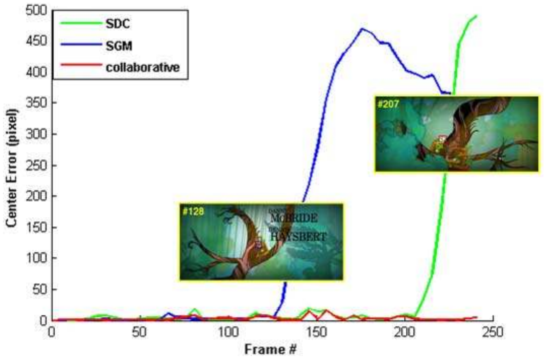

Fig. 12. Performance of tracking methods based on the SDC, SGM, and the collaborative models on the *panda* sequence.

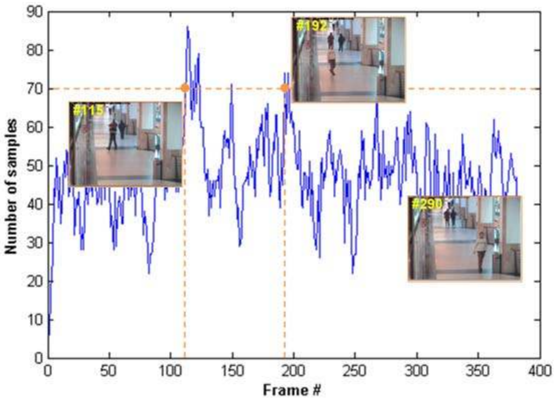

Fig. 13. The curve shows the values of n, the number of samples selected by the proposed particle selecting method (Eq. 6), in all the frames from the *caviar1* sequence.

set to be 600, the average number of samples selected by Eq. 5 of the proposed motion model is about 50. When the target is occluded (e.g., frame 115 and 192), more than 70 samples are used (based on the weights computed by Eq. 7). Overall, the proposed method uses less than 9% of that used with a simple Gaussian model and achieves good tracking results. Besides, as for the images of size 320 × 240, the speed of the algorithm running at a PC with 2.2GHz Pentium Dual core is about 1.25f/s compared with 0.33f/s without the improved motion model.

VII. CONCLUSION

In this paper, we propose and demonstrate an effective and robust tracking method based on the collaboration of generative and discriminative modules. In the proposed tracking algorithm, holistic templates are incorporated to construct a discriminative classifier that can effectively deal with cluttered and complex background. Local representations are adopted to form a robust histogram that considers the spatial information among local patches with an occlusion handling module, which enables our tracker to better handle heavy occlusions. The contributions of these holistic discriminative and local generative modules are integrated in a unified manner. Furthermore, the online update scheme reduces drifts and enhances the proposed method to adaptively account for appearance changes in dynamic scenes. Quantitative and qualitative comparisons with nine state-of-the-art algorithms on sixteen challenging image sequences demonstrate the robustness of the proposed tracking algorithm.

REFERENCES

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

[2] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2002, pp. 661–675.

[3] Y. Li, H. Ai, T. Yamashita, S. Lao, and M. Kawade, "Tracking in low frame rate video: A cascade particle filter with discriminative observers of different life spans," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 10, pp. 1728–1740, Oct. 2008.

[4] D. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 125–141, 2008.

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

[6] S. Wang, H. Lu, F. Yang, and M.-H. Yang, "Superpixel tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1323–1330.

[7] H. Grabner and H. Bischof, "On-line boosting and vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 260–267.

[8] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *Proc. 10th Eur. Conf. Comput. Vis.*, Jan. 2008, pp. 234–247.

[9] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jan. 2009, pp. 983–990.

[10] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, Jul. 2012.

[11] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Int. Joint Conf. Artif. Intell.*, 1981, pp. 121–130.

[12] S. Baker and I. Matthews, "Lucas-Kanade 20 years on: A unifying framework," *Int. J. Comput. Vis.*, vol. 56, no. 3, pp. 221–255, Feb. 2004.

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

[14] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1822–1829.

[15] X. Mei and H. Ling, "Robust visual tracking using ℓ_1 minimization," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Oct. 2009, pp. 1436–1443.

[16] B. Liu, L. Yang, J. Huang, P. Meer, L. Gong, and C. Kulikowski, "Robust and fast collaborative tracking with two stage sparse optimization," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, pp. 624–637.

[17] B. Liu, J. Huang, L. Yang, and C. Kulikowsk, "Robust tracking using local sparse appearance model and k-selection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 1313–1320.

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

[19] F. Tang, S. Brennan, Q. Zhao, and H. Tao, "Co-tracking using semi-supervised support vector machines," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.

[20] Q. Wang, F. Chen, W. Xu, and M.-H. Yang, "Online discriminative object tracking with local sparse representation," in *Proc. IEEE Workshop Appl. Comput. Vis.*, Jan. 2012, pp. 425–432.

[21] Q. Yu, T. B. Dinh, and G. G. Medioni, "Online tracking and reacquisition using co-trained generative and discriminative trackers," in *Proc. 10th ECCV*, 2008, pp. 678–691.

[22] R. Liu, J. Cheng, and H. Lu, "A robust boosting tracker with minimum error bound in a co-training framework," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2009, pp. 1459–1466.

[23] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof, "PROST: Parallel robust online simple tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 723–730.

[24] T. B. Dinh and G. G. Medioni, "Co-training framework of generative and discriminative trackers with partial occlusion handling," in *Proc. IEEE Workshop Appl. Comput. Vis.*, Jan. 2011, pp. 642–649.

[25] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1838–1845.

[26] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surveys*, vol. 38, no. 4, pp. 1–45, 2006.

[27] K. Cannons, "A review of visual tracking," Dept. Comput. Sci., York Univ., Toronto, ON, Canada, Tech. Rep. CSE-2008-07, 2008.

[28] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via multi-task sparse learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2042–2049.

[29] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Low-rank sparse learning for robust visual tracking," in *Proc. ECCV*, 2012, pp. 2042–2049.

[30] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[31] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE Conf. CVPR*, Jun. 2009, pp. 1794–1801.

[32] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. IEEE Conf. CVPR*, Jun. 2010, pp. 3360–3367.

[33] S. Gao, I. W.-H. Tsang, L.-T. Chia, and P. Zhao, "Local features are not lonely—Laplacian sparse coding for image classification," in *Proc. IEEE Conf. CVPR*, Jun. 2010, pp. 3555–3561.

[34] I. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 810–815, Jun. 2004.

[35] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Proc. IEEE Conf. CVPR*, Jun. 2010, pp. 1–10.

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

Wei Zhong received the M.Sc. degree in signal and information processing from the Dalian University of Technology, Dalian, China, in 2012. She is currently a Researcher with China Unicom, Dalian. Her research interests include computer vision and pattern recognition.

Huchuan Lu (SM'12) received the Ph.D. degree in system engineering and the M.Sc. degree in signal and information processing from the Dalian University of Technology (DUT), Dalian, China, in 2008 and 1998, respectively. He joined the faculty in 1998 and is currently a Full Professor with the School of Information and Communication Engineering, DUT. His current research interests include computer vision and pattern recognition with focus on visual tracking, saliency detection, and segmentation. He is also a member of the Association for Computing Machinery and an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, PART B.

Ming-Hsuan Yang (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 currently an Associate Professor of Electrical Engineering and Computer Science with the University of California, Merced, CA, USA. He serves as an Program Chair of the Asian Conference on Computer Vision in 2014, and an Area Chair for the IEEE International Conference on Computer Vision in 2011, the IEEE Conference on Computer Vision and Pattern Recognition in 2008, 2009, and 2014, the European Conference on Computer Vision in 2014, and the Asian Conference on Computer in 2009, 2010, and 2012. He has served as an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE from 2007 to 2011, and currently is an Associate Editor of the *International Journal of Computer Vision, Image and Vision Computing* and the *Journal of Artificial Intelligence Research*. He is a recipient of 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 Association for Computing Machinery.