Online Object Tracking With Sparse Prototypes

Dong Wang, Huchuan Lu, *Member, IEEE*, and Ming-Hsuan Yang, *Senior Member, IEEE*

***Abstract*— Online object tracking is a challenging problem as it entails learning an effective model to account for appearance change caused by intrinsic and extrinsic factors. In this paper, we propose a novel online object tracking algorithm with sparse prototypes, which exploits both classic principal component analysis (PCA) algorithms with recent sparse representation schemes for learning effective appearance models. We introduce ℓ_1 regularization into the PCA reconstruction, and develop a novel algorithm to represent an object by sparse prototypes that account explicitly for data and noise. For tracking, objects are represented by the sparse prototypes learned online with update. In order to reduce tracking drift, we present a method that takes occlusion and motion blur into account rather than simply includes image observations for model update. Both qualitative and quantitative evaluations on challenging image sequences demonstrate that the proposed tracking algorithm performs favorably against several state-of-the-art methods.**

***Index Terms*— Appearance model, ℓ_1 minimization, object tracking, principal component analysis (PCA), sparse prototypes.**

I. Introduction

AS ONE of the fundamental problems in computer vision, object tracking plays a critical role in numerous lines of research such as motion analysis, image compression, and activity recognition. While much progress has been made in the past decades, developing a robust online tracker is still a challenging problem due to difficulties to account for appearance change of a target object, which includes intrinsic (e.g., pose variation and shape deformation) and extrinsic factors (e.g., varying illumination, camera motion, and occlusions).

A tracking method typically consists of three components: an appearance (observation) model which evaluates the likelihood of an observed image patch (associated to a state) belonging to the object class; a dynamic model (or motion model), which aims to describe the states of an object over time (e.g., Kalman filter [1] and particle filter [2], [3]); and a search strategy for finding the likely states in the current frame (e.g., mean shift [1] and sliding window [4]). In this paper, we propose a robust appearance model that considers the effects of occlusion and motion blur. Hence, we only discuss key issues related to appearance models rather than present a detailed review of all components.

In order to develop effective appearance models for robust object tracking, several critical factors need to be considered. The first one is concerned with how objects are represented. Any representation scheme can be categorized based on adopted features (e.g., intensity [5], color [2], texture [6], Haar-like feature [4], [7], super-pixel based feature [8], and sparse coding [9]), and description models (e.g., holistic histogram [1], part-based histogram [10], and subspace representation [5]). Instead of treating the target object as a collection of low-level features, subspace representation methods provide a compact notion of the "thing" being tracked, which facilitates other vision tasks (e.g., object recognition).

Second, representation schemes can be either generative or discriminative. For object tracking, generative methods focus on modeling appearance and formulate the problem as finding the image observation with minimal reconstruction error (e.g., using templates [1], [10] and subspace models [5], [11], [12]). On the other hand, discriminative algorithms aim at determining a decision boundary that distinguishes the target from the background (e.g., using boosting algorithms [4], [6], [7], [13], and support vector machines [14], [15]). It has been shown that discriminative models perform better if the training set size is large [16], while generative models achieve higher generalization when limited data is available [17]. In addition, several algorithms that exploit the advantages of both generative and discriminative models have been proposed [15], [18]–[20]. In this paper, we focus on developing a robust algorithm using a generative appearance model that considers occlusion and motion blur to alleviate tracking drift.

Third, it has been shown that online learning facilitates tracking algorithms by adapting to appearance change of the target and the background. Numerous methods including template update [1], [21], incremental subspace learning [5], [11], [12], [22], [23], and online classifiers [4], [6] have been demonstrated to be effective for object tracking. However, due to straightforward update of appearance models using tracking results, slight inaccuracy can therefore result in incorrectly labeled training examples and degrade the models gradually with drifts. To address this problem, Avidan [6] adopts a simple outlier rejection scheme, and Babenko *et al.* [7] introduce multiple instance learning (MIL) [24] into visual tracking. Alternatively, Grabner *et al.* [13] propose a semi-supervised boosting algorithm to address the online update problem where labeled training examples come from the first frame only, and subsequent instances are regarded as unlabeled data. This strategy is further extended in [25], which introduces constraints for positive and negative examples to exploit the structure of unlabeled data.

Manuscript received October 20, 2011; revised April 24, 2012; accepted April 29, 2012. Date of publication June 5, 2012; date of current version December 20, 2012. The work of D. Wang and H. Lu was supported in part by the Natural Science Foundation (NSF) of China under Grant 61071209. The work of M.-H. Yang was supported in part by the NSF CAREER under Grant 1149783 and NSF IIS under Grant 1152576. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Carlo S. Regazzoni.

D. Wang 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: wangdong.ice@gmail.com; lhchuan@dlut.edu.cn).

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

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.2012.2202677

Notwithstanding the demonstrated success in reducing drifts, considerably few attempts have been made to directly address the occlusion problem, which remains as arguably the most critical factor for causing tracking failures. To deal with this challenging problem, several part-based models have been employed. Adam *et al.* [10] propose a fragment-based tracking method using histograms. Yang *et al.* [26] apply the "bag of words" model from the category-level object recognition [27] to visual tracking. In [21], Mei *et al.* propose a tracking algorithm by casting the problem as determining the most likely patch with a sparse representation of templates. This method is able to model partial occlusion by sparse representation of trivial templates. However the space or time complexity is quite significant. Recently, motivated by the success of the HOG-LBP pedestrian detector [28], Dinh *et al.* [29] propose a complex co-training approach using generative and discriminative trackers that deals with partial occlusion. Although these algorithms perform relatively well in handling partial occlusion, they often fail to track objects in challenging image sequences with drastic appearance change and background clutters.

In this paper, we propose a robust generative tracking algorithm with adaptive appearance model which handles partial occlusion and other challenging factors. Compared with part-based models [10], [26], our algorithm maintains holistic appearance information and therefore provides a compact representation of the tracked target. By exploiting the advantage of subspace representation, our algorithm is able to process higher resolution image observations, and performs more efficiently with favorable results than the existing method based on sparse representation of templates [21]. In comparison to the subspace based tracking algorithms [5], [12], our algorithm is able to deal with heavy occlusion effectively. Different from [29], our algorithm does not require a complex combination of generative and discriminative trackers for handling partial occlusion. Numerous experiments and evaluations on challenging image sequences bear out that the proposed algorithm is efficient and effective for robust object tracking.

II. Related Work and Context

Much work has been done in object tracking and more thorough reviews on this topic can be found in [30]. In this section we discuss the most relevant algorithms and put this work in proper context.

A. Object Tracking With Incremental Subspace Learning

Object tracking via online subspace learning ([5], [11], [12], [22], [23]) has attracted much attention in recent years. The incremental visual tracking (IVT) method [5] introduces an online update approach for efficiently learning and updating a low dimensional PCA subspace representation of the target object. Several experimental results demonstrate that PCA subspace representation with online update is effective in dealing with appearance change caused by in-plane rotation, scale, illumination variation and pose change. However, it has also been shown that the PCA subspace based representation scheme is sensitive to partial occlusion, which can be explained by Eq. 1

$$\mathbf{y} = \mathbf{U}\mathbf{z} + \mathbf{e} \tag{1}$$

where $\mathbf{y}$ denotes an observation vector, $\mathbf{z}$ indicates the corresponding coding or coefficient vector, $\mathbf{U}$ represents a matrix of column basis vectors, and $\mathbf{e}$ is the error term (Figure 1(a)).

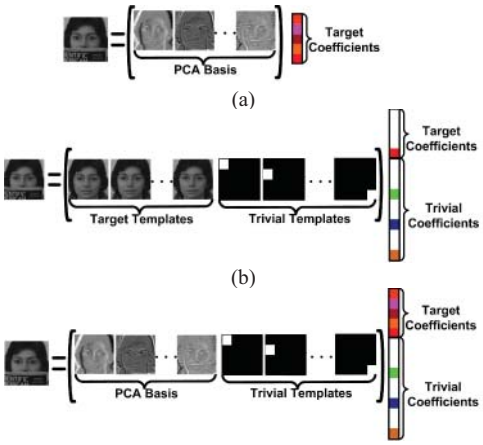

Fig. 1. Motivation of our work. (a) PCA. (b) Sparse representation. (c) Sparse prototypes. Prototypes consist of PCA basis vectors and trivial templates.

In PCA, the underlying assumption is that the error vector $\mathbf{e}$ is Gaussian distributed with small variances (i.e., small dense noise). Therefore, the coding vector $\mathbf{z}$ can be estimated by $\mathbf{z} = \mathbf{U}^\top \mathbf{y}$, and the reconstruction error can be approximated by $\|\mathbf{y} - \mathbf{U}\mathbf{U}^\top \mathbf{y}\|_2^2$. However, this assumption does not hold for object representation in visual tracking when partial occlusion occurs as the noise term cannot be modeled with small variance. Hence, the IVT method is sensitive to partial occlusion. In addition, the IVT method is not equipped with an effective update mechanism since it simply uses new observations for learning new basis vectors without detecting partial occlusion and processing these samples accordingly. In order to account for partial occlusion for object tracking, we model the error term $\mathbf{e}$ with arbitrary but sparse noise.

B. Object Tracking With Sparse Representation

Sparse representation has recently been extensively studied and applied in pattern recognition and computer vision, e.g., face recognition [31], super-resolution [32], and image inpainting [33]. Motivated by [31], Mei *et al.* [21] propose an algorithm (ℓ_1 tracker) by casting the tracking problem as finding the most likely patch with sparse representation and handling partial occlusion with trivial templates by

$$\mathbf{y} = \mathbf{A}\mathbf{z} + \mathbf{e} = [\mathbf{A}\ \mathbf{I}] \begin{bmatrix} \mathbf{z} \\ \mathbf{e} \end{bmatrix} = \mathbf{B}\mathbf{c} \tag{2}$$

where $\mathbf{y}$ denotes an observation vector, $\mathbf{A}$ represents a matrix of templates, $\mathbf{z}$ indicates the corresponding coefficients, and $\mathbf{e}$ is the error term which can be viewed as the coefficients of trivial templates. Figure 1(b) illustrates the sparse representation scheme with trivial templates for object tracking.

By assuming that each candidate image patch is sparsely represented by a set of target and trivial templates, Eq. 2 can

be solved via ℓ_1 minimization [21]

$$\min \frac{1}{2} \|\mathbf{y} - \mathbf{Bc}\|_2^2 + \lambda \|\mathbf{c}\|_1 \tag{3}$$

where $\|.\|_1$ and $\|.\|_2$ denote the ℓ_1 and ℓ_2 norms respectively.

The underlying assumption of this approach is that error $\mathbf{e}$ can be modeled by arbitrary but sparse noise, and therefore it can be used to handle partial occlusion. However, the ℓ_1 tracker has two main drawbacks. First, the computational complexity limits its performance. As it requires solving a series of ℓ_1 minimization problems, it often deals with low-resolution images (e.g., 12×15 patches in [21]) as a tradeoff of speed and accuracy. Such low-resolution images may not capture sufficient visual information to represent objects for tracking. The ℓ_1 tracker is computationally expensive even with further improvements [34]. Second, it does not exploit rich and redundant image properties which can be captured compactly with subspace representations. We present an efficient and effective representation that factors out the part describing the object appearance and the other part for noise.

C. Motivation of This Work

In this paper, we exploit the strength of both subspace learning and sparse representation for modeling object appearance. It can be viewed as introducing ℓ_1 regularization into subspace representation with PCA. For object tracking, we model target appearance with PCA basis vectors, and account for occlusion with trivial templates by

$$\mathbf{y} = \mathbf{Uz} + \mathbf{e} = [\mathbf{U}\ \mathbf{I}] \begin{bmatrix} \mathbf{z} \\ \mathbf{e} \end{bmatrix} \tag{4}$$

where $\mathbf{y}$ denotes an observation vector, $\mathbf{U}$ represents a matrix of column basis vectors, $\mathbf{z}$ indicates the coefficients of basis vectors, and $\mathbf{e}$ is the error term (which can be viewed as the coefficients of trivial templates). In our formulation, the prototypes consist of a small number of PCA basis vectors and a set of trivial templates (See Figure 1(c)). As $\mathbf{e}$ is assumed to be arbitrary but sparse noise, we solve Eq. 4 by

$$\min_{\mathbf{z},\mathbf{e}} \frac{1}{2} \|\mathbf{y} - \mathbf{Uz} - \mathbf{e}\|_2^2 + \lambda \|\mathbf{e}\|_1. \tag{5}$$

We present an algorithm to solve this optimization problem in the next section. Here, we first highlight the difference between the formulations in Eq. 5 and Eq. 3. For the formulation with Eq. 3, the coefficients for both target and trivial templates should be sparse (as illustrated in Figure 1(b)) since the target templates are coherent [35] and coefficients for trivial templates are used to model partial occlusion. However, for our formulation with Eq. 5, coefficients for trivial templates should be sparse while the coefficients for the basis vectors are not sparse as PCA basis vectors are not coherent but orthogonal. As the number of trivial templates is much larger than the number of basis vectors, an observation can be sparsely represented by prototypes. Thus, we need to develop an algorithm to solve Eq. 5 rather than use existing ones.

Our formulation has the following advantages. First, compared with the incremental subspace representation of the IVT tracker [5], our method models partial occlusion explicitly and therefore handles it effectively. Second, compared with the ℓ_1 tracker [21], our algorithm is able to handle high-resolution image patches with less computational complexity by exploiting subspace representation.

Algorithm 1 Algorithm for Computing $\mathbf{z}_{opt}$ and $\mathbf{e}_{opt}$

Input: An observation vector $\mathbf{y}$, orthogonal basis vectors $\mathbf{U}$, and a small constant λ.
1: Initialize $\mathbf{e}_0 = \mathbf{0}$ and $i = 0$
2: Iterate
3: Obtain $\mathbf{z}_{i+1}$ via $\mathbf{z}_{i+1} = \mathbf{U}^\top (\mathbf{y} - \mathbf{e}_i)$
4: Obtain $\mathbf{e}_{i+1}$ via $\mathbf{e}_{i+1} = S_\lambda (\mathbf{y} - \mathbf{Uz}_{i+1})$
5: $i \leftarrow i + 1$
6: Until convergence or termination
Output: $\mathbf{z}_{opt}$ and $\mathbf{e}_{opt}$

III. Object Representation via Orthogonal Basis Vectors and ℓ_1 Regularization

In this section, we propose an algorithm for object representation with sparse prototypes as formulated in Eq. 5. Let the objective function be $L(\mathbf{z}, \mathbf{e}) = \frac{1}{2} \|\mathbf{y} - \mathbf{Uz} - \mathbf{e}\|_2^2 + \lambda \|\mathbf{e}\|_1$, the optimization problem is

$$\begin{aligned} &\min_{\mathbf{z},\mathbf{e}} L(\mathbf{z}, \mathbf{e}) \\ &\text{s.t. } \mathbf{U}^\top \mathbf{U} = \mathbf{I} \end{aligned} \tag{6}$$

where $\mathbf{y} \in R^{d \times 1}$ denotes an observation vector, $\mathbf{U} \in R^{d \times k}$ represents a matrix of orthogonal basis vectors, $\mathbf{z} \in R^{k \times 1}$ indicates the coefficients of basis vectors, $\mathbf{e} \in R^{d \times 1}$ describes the error term, λ is a regularization parameter, and $\mathbf{I} \in R^{d \times d}$ indicates an indentity matrix (where d is the dimension of the observation vector, and k represents the number of basis vectors). As there is no close-form solution for Eq. 6, we propose an iterative algorithm to compute $\mathbf{z}_{opt}$ and $\mathbf{e}_{opt}$.

Lemma 1: Given $\mathbf{e}_{opt}$, $\mathbf{z}_{opt}$ can be obtained from $\mathbf{z}_{opt} = \mathbf{U}^\top (\mathbf{y} - \mathbf{e}_{opt})$.

Proof: If $\mathbf{e}_{opt}$ is given, the problem of Eq. 6 is equivalent to the minimization of $J(\mathbf{z})$, where $J(\mathbf{z}) = \frac{1}{2} \|(\mathbf{y} - \mathbf{e}_{opt}) - \mathbf{Uz}\|_2^2$. This is a simple least squares problem, and the solution can be easily found as $\mathbf{z}_{opt} = \mathbf{U}^\top (\mathbf{y} - \mathbf{e}_{opt})$. ■

Lemma 2: Given $\mathbf{z}_{opt}$, $\mathbf{e}_{opt}$ can be obtained from $\mathbf{e}_{opt} = S_\lambda (\mathbf{y} - \mathbf{Uz}_{opt})$ where $S_\tau(x)$ is a shrinkage operation defined as $S_\tau(x) = \text{sgn}(x) \cdot (|x| - \tau)$.

Proof: If $\mathbf{z}_{opt}$ is given, the minimization of Eq. 6 is equivalent to the minimization of $G(\mathbf{e}) = \frac{1}{2} \|\mathbf{e} - (\mathbf{y} - \mathbf{Uz}_{opt})\|_2^2 + \lambda \|\mathbf{e}\|_1$. This is a convex optimization problem and the global minimum can be found by the shrinkage operator, $\mathbf{e}_{opt} = S_\lambda (\mathbf{y} - \mathbf{Uz}_{opt})$ [36]. ■

By Lemmas 1 and 2, the optimization in Eq. 5 can be solved efficiently. The steps of our algorithm are presented in Table 1.

We note that several approaches for sparse principal component analysis have been proposed [37]–[39]. These methods extend classical PCA algorithms to find sparse factors via non-negative matrix factorization, ℓ_1 penalty, and LASSO approach for generic data sets. Our work focuses on inferencing sparse

coefficients with prototypes (PCA basis vectors and trivial templates as illustrated in Figure 1(c) as well as Eq. 4), whereas sparse PCA (SPCA) aims to learn sparse basis vectors from a given training set. As the proposed algorithm exploits image properties for representing observations with sparse prototypes, it is likely to be more effective for vision applications. The experimental results presented in Section V bear out the motivation of this formulation for object tracking.

IV. Object Tracking via Sparse Prototypes

Object tracking can be considered as a Bayesian inference task in a Markov model with hidden state variables [5]. Given a set of observed images $\mathbf{Y}_t = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_t\}$ at the t-th frame, we estimate the hidden state variable $\mathbf{x}_t$ recursively

$$p(\mathbf{x}_t|\mathbf{Y}_t) \propto p(\mathbf{y}_t|\mathbf{x}_t) \int p(\mathbf{x}_t|\mathbf{x}_{t-1})\, p(\mathbf{x}_{t-1}|\mathbf{Y}_{t-1}) d\mathbf{x}_{t-1} \tag{7}$$

where $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ represents the dynamic (motion) model between two consecutive states, and $p(\mathbf{y}_t|\mathbf{x}_t)$ denotes observation model that estimates the likelihood of observing $\mathbf{y}_t$ at state $\mathbf{x}_t$. The optimal state of the tracked target given all the observations up to t-th frame is obtained by the maximum a posteriori estimation over N samples at time t by

$$\widehat{\mathbf{x}}_t = \arg\max_{\mathbf{x}_t^i} p\left(\mathbf{y}_t^i|\mathbf{x}_t^i\right) p\left(\mathbf{x}_t^i|\mathbf{x}_{t-1}\right), \; i = 1, 2, \ldots, N \tag{8}$$

where $\mathbf{x}_t^i$ indicates the i-th sample of the state $\mathbf{x}_t$, and $\mathbf{y}_t^i$ denotes the image patch predicated by $\mathbf{x}_t^i$. Figure 2 shows the main steps of our tracking algorithm. At the outset, the state of the target object is manually initialized.

A. Dynamic Model

In this paper, we apply an affine image warp to model the target motion between two consecutive frames. The six parameters of the affine transform are used to model $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ of a tracked target. Let $\mathbf{x}_t = \{x_t, y_t, \theta_t, s_t, \alpha_t, \phi_t\}$, where $x_t, y_t, \theta_t, s_t, \alpha_t, \phi_t$ denote x, y translations, rotation angle, scale, aspect ratio, and skew respectively. The state transition is formulated by random walk, i.e., $p(\mathbf{x}_t|\mathbf{x}_{t-1}) = N(\mathbf{x}_t; \mathbf{x}_{t-1}, \Psi)$, where Ψ is a diagonal covariance matrix.

B. Observation Model

If no occlusion occurs, an image observation $\mathbf{y}_t$ can be assumed to be generated from a subspace of the target object spanned by $\mathbf{U}$ and centered at $\boldsymbol{\mu}$. However, it is necessary to account for partial occlusion in an appearance model for robust object tracking. We assume that a centered image observation $\overline{\mathbf{y}}_t$ ($\overline{\mathbf{y}}_t = \mathbf{y}_t - \boldsymbol{\mu}$) of the tracked object can be represented by a linear combination of the PCA basis vectors U and few elements of the identity matrix $\mathbf{I}$ (i.e., trivial templates) (Figure 1(c)), i.e., $\overline{\mathbf{y}}_t = \mathbf{U}\mathbf{z}_t + \mathbf{e}_t$. We note that $\mathbf{U}$ consists of a few basis vectors and $\mathbf{z}_t$ is usually dense. On the other hand, $\mathbf{e}_t$ accounts for noise or occlusion. Some samples drawn by our dynamic model are shown in Figure 3. If there is no occlusion, the most likely image patch can be effectively represented by the PCA basis vectors and coefficients corresponding to trivial templates (referred as trivial coefficients) tend to be zeros (as illustrated by the sample $\mathbf{y}^1$ of Figure 3(b)). On the other hand, a candidate patch that does not correspond to the true target location (e.g., mis-aligned sample) often leads to a dense representation (as illustrated by the samples $\mathbf{y}^2$ and $\mathbf{y}^3$ of Figure 3(b)). If partial occlusion occurs, the most likely image patch can be represented as a linear combination of PCA basis vectors and very few number of trivial templates (as illustrated by $\mathbf{y}^4$ of Figure 3(c)). As shown in Figure 3(c), the trivial coefficients of the sample that best matches the target, $\mathbf{y}^4$, are much sparser than those that do not correspond to the true object location ($\mathbf{y}^5$ and $\mathbf{y}^6$). Based on these observations, we note that the precise localization of the tracked target can be benefited by penalizing the sparsity of trivial coefficients.

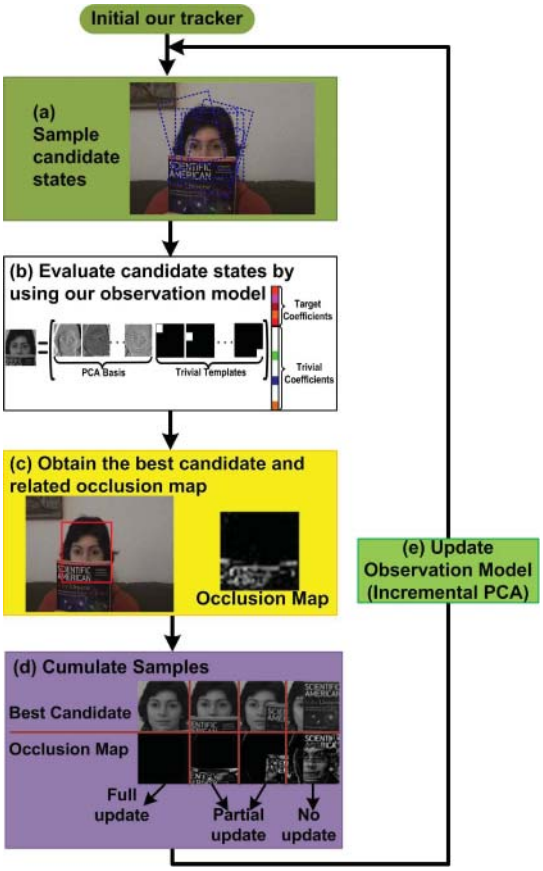

Fig. 2. Our tracking algorithm. It consists of three main parts: dynamic model, observation model, and update module.

For each observation corresponding to a predicted state, we solve the following equation efficiently using the proposed algorithm as summarized in Table 1

$$L\left(\mathbf{z}^i, \mathbf{e}^i\right) = \min_{\mathbf{z}^i, \mathbf{e}^i} \frac{1}{2} \left\| \overline{\mathbf{y}}^i - \mathbf{U}\mathbf{z}^i - \mathbf{e}^i \right\|_2^2 + \lambda \left\| \mathbf{e}^i \right\|_1, \tag{9}$$

and obtain $\mathbf{z}^i$ and $\mathbf{e}^i$, where i denotes the i-th sample of the state $\mathbf{x}$ (without loss of generality, we drop the frame index t). The observation likelihood can be measured by the reconstruction error of each observed image patch

$$p(\overline{\mathbf{y}}^i|\mathbf{x}^i) = \exp\left(-\left\| \overline{\mathbf{y}}^i - \mathbf{U}\mathbf{z}^i \right\|_2^2\right). \tag{10}$$

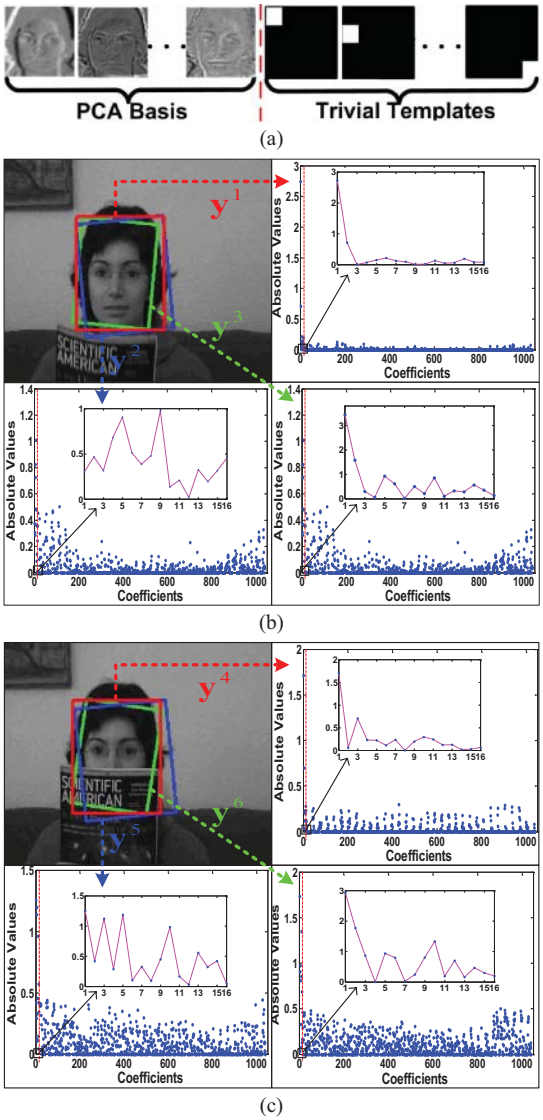

Fig. 3. Coefficients and alignment as well as occlusions. This figure illustrates how the PCA and trivial coefficients indicate whether a sample is aligned on the target when it is unconcluded or occluded. (a) Prototypes consist of PCA basis vectors and trivial templates. (b) Good and bad candidates when no occlusion occurs. (c) Good and bad candidates when partial occlusion occurs. The red bounding box represents a good candidate while the blue and green bounding boxes denote two bad samples. The coefficients of PCA basis vectors are shown with circles on the right. The trivial coefficients are shown with solid lines in the sub-figures pointed by arrows. See text for details.

However, Eq. 10 does not consider occlusion. Thus, we use a mask to factor out non-occluding and occluding parts

$$p\left(\overline{\mathbf{y}}^i|\mathbf{x}^i\right) = \exp\left[-\left(\left\|\mathbf{w}^i \odot \left(\overline{\mathbf{y}}^i - \mathbf{U}\mathbf{z}^i\right)\right\|_2^2 + \beta \sum\left(1 - \mathbf{w}^i\right)\right)\right] \tag{11}$$

where $\mathbf{w}^i = \left[w_1^i, w_2^i, \ldots, w_d^i\right]^\top$ is a vector that indicates the zero elements of $\mathbf{e}^i$, $\odot$ is the Hadamard product (element-wise product), and β is a penalty term (simply set to λ in this study). If the j-th element of $\mathbf{e}^i$ (obtained from Eq. 9), is zero then $w_j^i = 1$, otherwise $w_j^i = 0$. The first part of the exponent accounts for the reconstruction error of unoccluded proportion of the target object, and the second term aims to penalize labeling any pixel as being occluded. The experimental results in Section V demonstrate the effectiveness of our formulation.

TABLE I
EVALUATED IMAGE SEQUENCES

Image sequence	# Frames	Challenging factors
Occlusion 1 [10]	898	partial occlusion
Occlusion 2 [7]	819	partial occlusion in-plane rotation, out-plane rotation
Caviar 1 [41]	382	partial occlusion, scale change
Caviar 2 [41]	500	partial occlusion, scale change
Car 4 [5]	659	illumination variation, scale change
Singer1 [40]	321	illumination variation, scale change
David Indoor [5]	462	illumination variation scale change, out-plane rotation
Car 11 [5]	393	illumination variation scale change, background clutter
Deer [40]	71	abrupt motion, background clutter
Jumping [25]	313	abrupt motion
Lemming [19]	1336	out-plane rotation, scale change occlusion, background clutter
Cliffbar [7]	471	in-plane rotation, scale change background clutter, abrupt motion

C. Update of Observation Model

It is essential to update the observation model for handling appearance change of a target object for visual tracking. The model degrades if some imprecise samples are used for update, thereby causing tracking drift. Instead, we explore the trivial coefficients for occlusion detection since the corresponding templates are used to account for noise. First, each trivial coefficient vector corresponds to a 2D map as a result of reverse raster scan of an image patch. A non-zero element of this map indicates that pixel is occluded (referred as occlusion map). Second, we compute the ratio η, of the number of non-zero pixels and the number of occlusion map pixels. We use two thresholds tr_1 and tr_2 to describe the degree of occlusion (e.g., $tr_1 = 0.1$ and $tr_2 = 0.6$ in this paper). Third, based on the occlusion ratio η, we apply one of the three kinds of operations: full, partial, and no update. If $\eta < tr_1$, we directly update the model with this sample. If $tr_1 \leq \eta \leq tr_2$, it indicates that the target is partially occluded. We then replace the occluded pixels by its corresponding parts of the average observation $\boldsymbol{\mu}$, and use this recovered sample for update. Otherwise if $\eta > tr_2$, it means that a significant part of the target object is occluded, and we discard this sample without update. Figure 2(d) shows several cases regarding three update scenarios. After we cumulate enough samples, we use an incremental principal component analysis method [5] to update our observation model (i.e., PCA basis vectors $\mathbf{U}$ and the average vector $\boldsymbol{\mu}$).

D. Discussion

We note that our tracker is robust since it is able to tackle the presence of potential outliers (e.g., occlusion and mis-alignment) with the proposed observation model and update scheme. For the accurate location of the tracked target, the proposed representation model (Eq. 9) and observation likelihood (Eq. 11) as presented in Section IV-B, enable the

tracker to handle partial occlusion explicitly and facilitate it to choose the well-aligned observation (See Figure 3 for illustrated examples). With the update scheme of the proposed observation model, as presented in Section IV-C, our tracker is able to alleviate the problem caused by inaccurate samples (i.e., update the model by outliers).

V. Experiments

The proposed algorithm is implemented in MATLAB which runs at 2 frames per second on a Pentium 2.0 GHz Dual Core PC with 3 GB memory. For each sequence, the location of the target object is manually labeled in the first frame. For PCA representation, each image observation is normalized to 32×32 pixels and 16 eigenvectors are used in all experiments. In addition, we use 1024 trivial templates. With our formation in Eq. 4, the dimensionality of $\mathbf{z}$ and $\mathbf{e}$ is 16 and 1024 respectively. As a trade-off between computational efficiency and effectiveness, 600 particles are used and our tracker is incrementally updated every 5 frames. The regularization constant λ is set to 0.05 in all experiments. We present some representative results in this section. All the MATLAB source codes and datasets are available on our web sites (http://ice.dlut.edu.cn/lu/publications.html, http://faculty.ucmerced.edu/mhyang/pubs.html).

We use twelve challenging image sequences in the experiments. Table I lists all the evaluated image sequences and only gray scale information is used for experiments. We use the ground truth information when it is provided along with each sequence (e.g., *Occlusion 1*, *Caviar 1*, *Caviar 2*), and label the other videos on our own. We evaluate the proposed tracker against six state-of-the-art algorithms using the source codes provided by the authors for fair comparisons, including the IVT [5], ℓ_1 [21], FragTrack [10], MILTrack [7], VTD [40], and PN algorithms [25]. Both qualitative as well as quantitative evaluations are presented, and more results can be found at http://faculty.ucmerced.edu/mhyang/video/prototype1.avi.

A. Qualitative Evaluation

Heavy occlusion: In the *Occlusion 1* sequence [10], our algorithm, FragTrack and ℓ_1 methods perform better, as shown in Figure 4(a), since these methods take partial occlusion into account. The FragTrack method is able to handle occlusion via the part-based representation with histograms. The proposed and ℓ_1 trackers handle occlusion using sparse representation with trivial templates. For the *Occlusion 2* sequence (Figure 4(b)), our tracker performs best especially when partial occlusion or in-plane rotation occurs (e.g., #0150, #0500, and #0700). In these frames, the FragTrack method performs poorly since it does not handle appearance change caused by pose and occlusion. Although the MILTrack method is able to track the target object, it is not able to estimate the in-plane rotation due to its design. We note that it is not straightforward to extend this method by simply using an affine motion model as a result of the adopted representation with generalized Haar-like features. On the other hand, the ℓ_1 tracker does not perform well in this sequence. This can be explained by that the simple update method of the ℓ_1 tracker takes new image observations for update without factoring out occlusion.

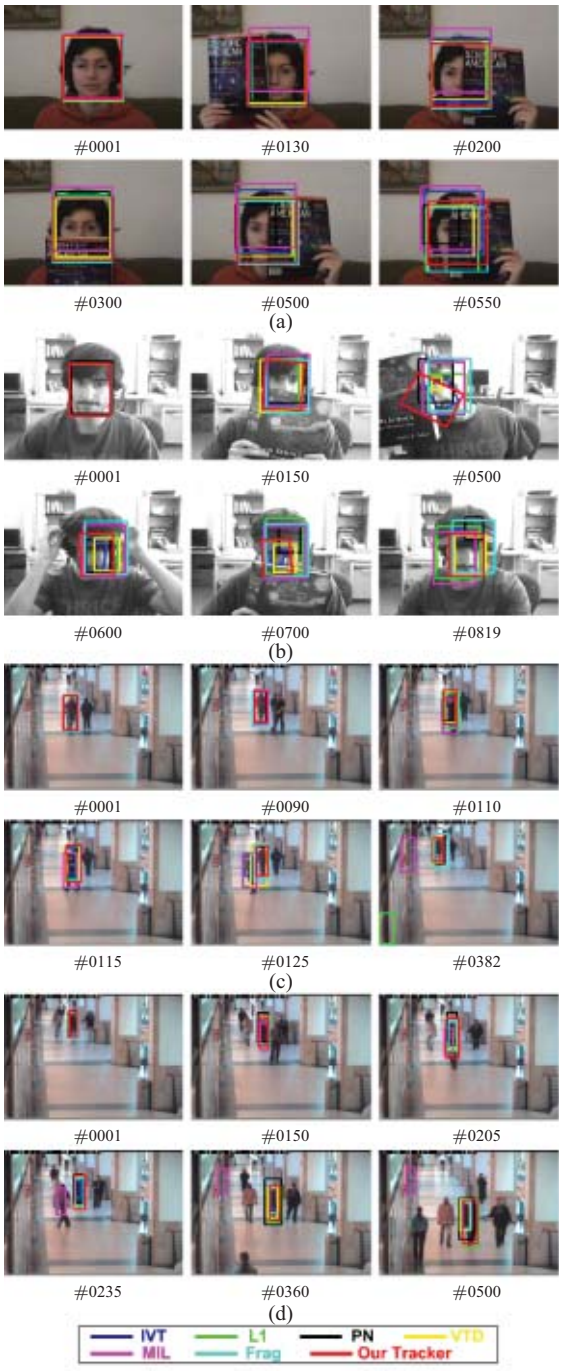

Fig. 4. Qualitative evaluation: objects undergo heavy occlusion and pose change. Similar objects also appear in the scenes. (a) Occlusion 1. (b) Occlusion 2. (c) Caviar 1. (d) Caviar 2.

Figure 4(c)-(d) shows the tracking results of different algorithms in surveillance videos. These videos are challenging as they contain scale change, partial occlusion and similar objects. The MILTrack method does not perform well when the target is occluded by a similar object. As the generalized Haar-like features are used for object representation in the MILTrack method, they are less effective when similar objects occlude each other. The ℓ_1 and IVT trackers drift away from

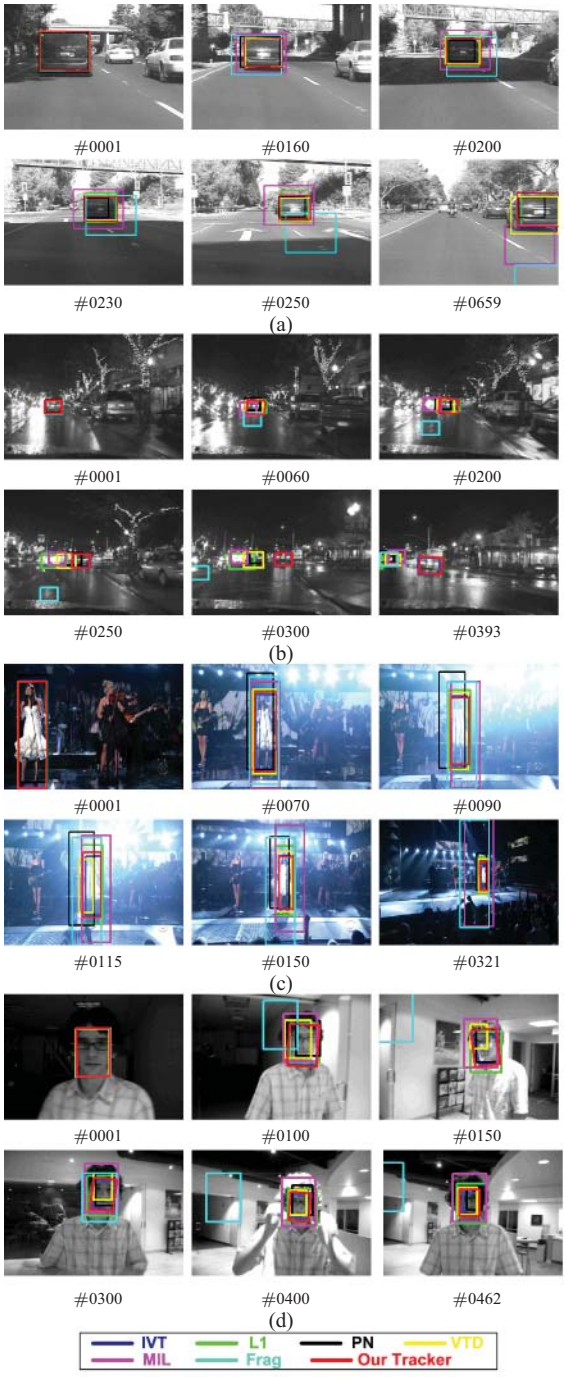

Fig. 5. Qualitative evaluation: object appearance change drastically due to large variation of lighting, pose, scale and low contrast. (a) Car 4. (b) Car 11. (c) Singer 1. (d) David Indoor.

the target after it is occluded by a similar object. In contrast, our algorithm performs well in terms of position and scale even when the target is heavily occluded.

Illumination change: Figure 5 shows results from four challenging sequences with significant change of illumination and scale, as well as pose variation. For the *Car 4* sequence, there is a drastic lighting change when the vehicle goes underneath the overpass or the trees. The target object is small with low contrast and drastic illumination change in

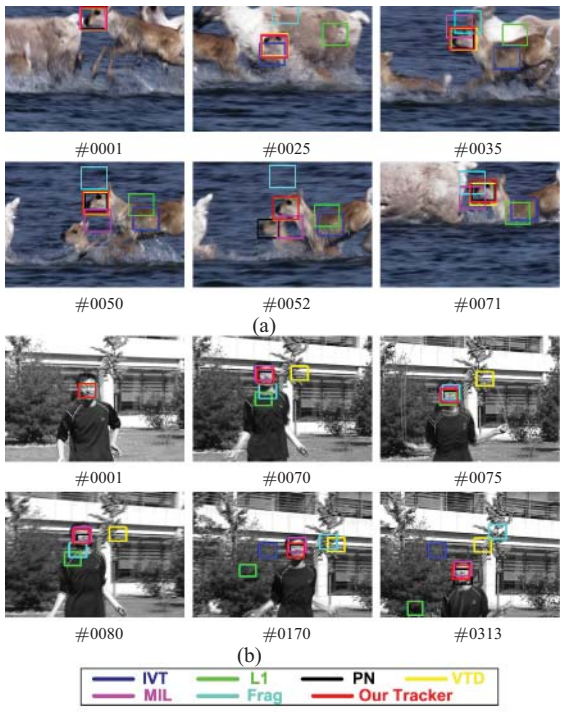

Fig. 6. Qualitative evaluation: significant object appearance change due to fast movement and motion blur in cluttered background. (a) Deer. b) Jumping.

the *Car 11* video. The IVT and proposed algorithms perform well in tracking this vehicle whereas the other methods drift away when drastic illumination variation occurs (e.g., #0200) or when similar objects appear in the scene (e.g., #0300). In the *Singer 1* video, drastic change of illumination and scale makes it difficult to track. Likewise, the appearance of the person changes significantly when he walks from a dark room into areas with spot light in the *David Indoor* video. In addition, appearance change caused by scale and pose as well as camera motion pose great challenges. We note that the IVT and proposed trackers perform better than the other methods. This can be attributed to that appearance change of the object can be well approximated by a subspace at fixed pose [42]. We also note that some trackers do not adapt to scale (e.g., MILTrack) or in-plane rotation (e.g., MILTrack, PN, and FragTrack).

Fast motion: Figure 6 illustrates the tracking results using the *Deer* and *Jumping* sequences. As the objects undergo abrupt motion, it is difficult to predict their locations. In addition, it is rather challenging to account for appearance change caused by motion blur and properly update these appearance models. In the *Deer* video, the VTD method and our tracker perform better than the other algorithms. For the *Jumping* sequence, our tracker performs better than the other methods whereas the MILTrack and PN algorithms are able to track the objects in some frames. We note that the PN algorithm is equipped with a re-initialization mechanism which facilitates object tracking. Due to repetitive motion in the *Jumping* sequence, some trackers may be able to track the object again by chance after failure (e.g., ℓ_1, MILTrack and FragTrack methods from #0070 to #0075). Similarly, some

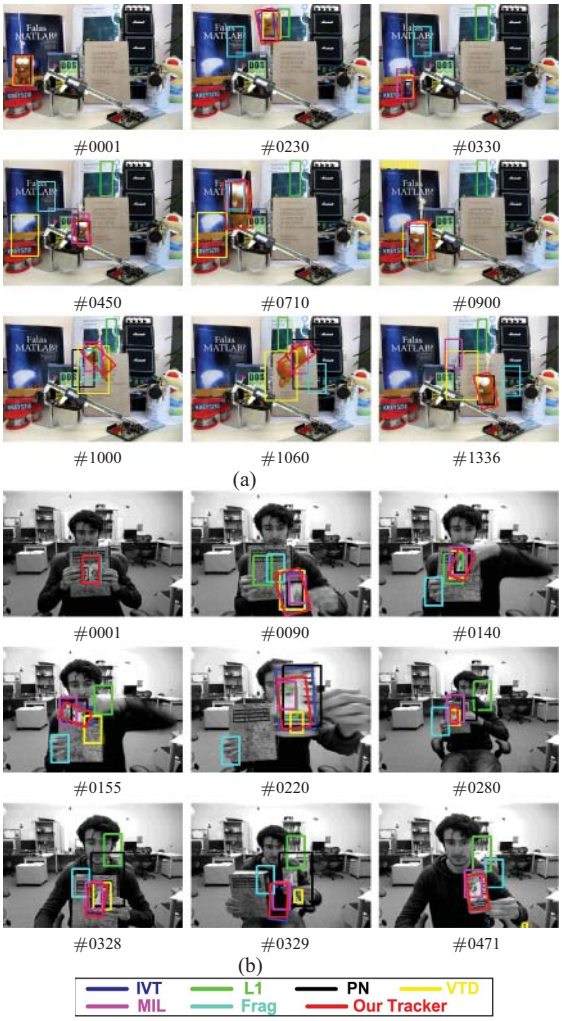

Fig. 7. Qualitative evaluation: objects undergo in- and out-plane rotation, fast motion, scale change in cluttered background. (a) Lemming. (b) Cliffbar.

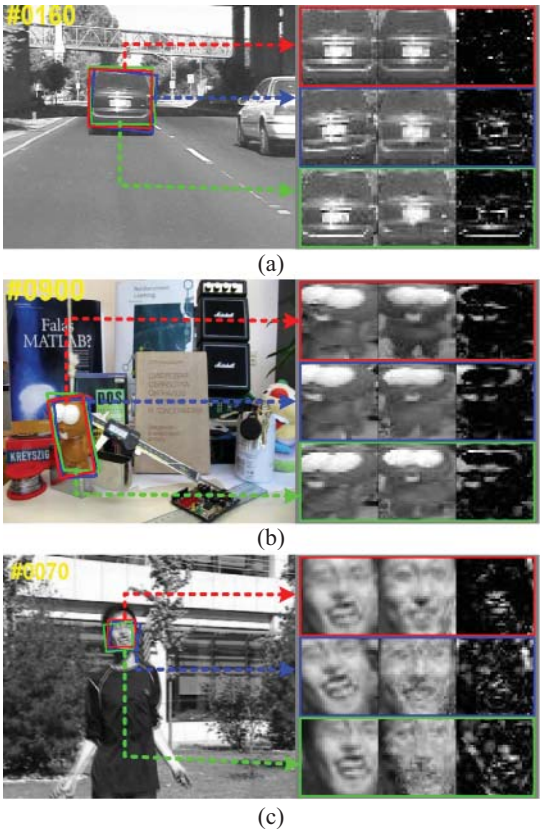

Fig. 8. Some representative cases. The red bounding box represents a good candidate while the blue and green boxes denote two bad samples. For each sample, the original sample image $\mathbf{y}$, the reconstructed image $\mathbf{Uz}+\overline{\mathbf{y}}$, and the error image $\mathbf{e}$ are shown from left to right in the right panel. (a) Illumination variation. (b) Pose change and background clutter. (c) Motion blur.

trackers may be able to capture the object by chance as the object reappears at the same image location due to camera pan (e.g., VTD and PN methods from #0052 to #0071) in the *Deer* video.

Cluttered background: The *Lemming* sequence is challenging for visual tracking as the object undergoes change of scale and pose, as well as heavy occlusion in cluttered background. Figure 7(a) shows that the proposed and PN trackers perform better than the other methods. It is worth noting that our tracker is able to adapt to scale change (e.g., #0001, #0230, #0710, and #1336), in-plane rotation (e.g., #0230, #1000, #1060, and #1336), and occlusion (e.g., #0330, and #0450). The object in the *Cliffbar* clip (Figure 7(b)) undergoes scale change, in-plane rotation, abrupt motion in a cluttered background. In addition, the target and the surrounding region have similar texture. The ℓ_1 and FragTrack methods perform poorly since the surrounding background is similar to the target object (#0090 of Figure 7(b)). The IVT algorithm fails after abrupt motion occurs (e.g., #0328, and #0329) and the PN tracker drifts gradually (e.g., #0280, #0328, and #0329). These results can be attributed to problem with appearance update. Both the MILTrack method and our algorithm are able to track the locations. However, the proposed algorithm adapts better to change of scale (e.g., #0001, #0090, #0280, and #0471) and rotation (e.g., #0001, #0140, #0155, and #0471).

Discussion: In Section IV, we present some justifications why the proposed method is able with partial occlusion effectively by using two representative cases (Figure 3(b) and (c)). We further present more results to illustrate how our algorithm handles other challenging factors. Figure 8 shows three representative tracking results under illumination variation, pose change, background clutter and motion blur. As shown in Figure 8(a) and (b), the best candidates can be well represented by the PCA basis and therefore the error terms are more sparse than those of the misaligned candidates. This can be attributed to the strength of subspace representation. If the tracked target undergoes illumination variation and slight pose change, the appearance variation can be well modeled by a low dimension PCA subspace. Thus, our tracker performs well when the target objects undergo illumination variation and pose change. In addition, we note that accurate locations of the tracked objects can be obtained by penalizing the sparsity of the error term. Our tracker capture the targets accurately when they appear in cluttered backgrounds (Figure 8(b)) or move abruptly (Figure 8(c)). In Figure 8(c) while the well-aligned and mis-aligned candidates are not well reconstructed, the error terms of the mis-aligned ones are larger and the

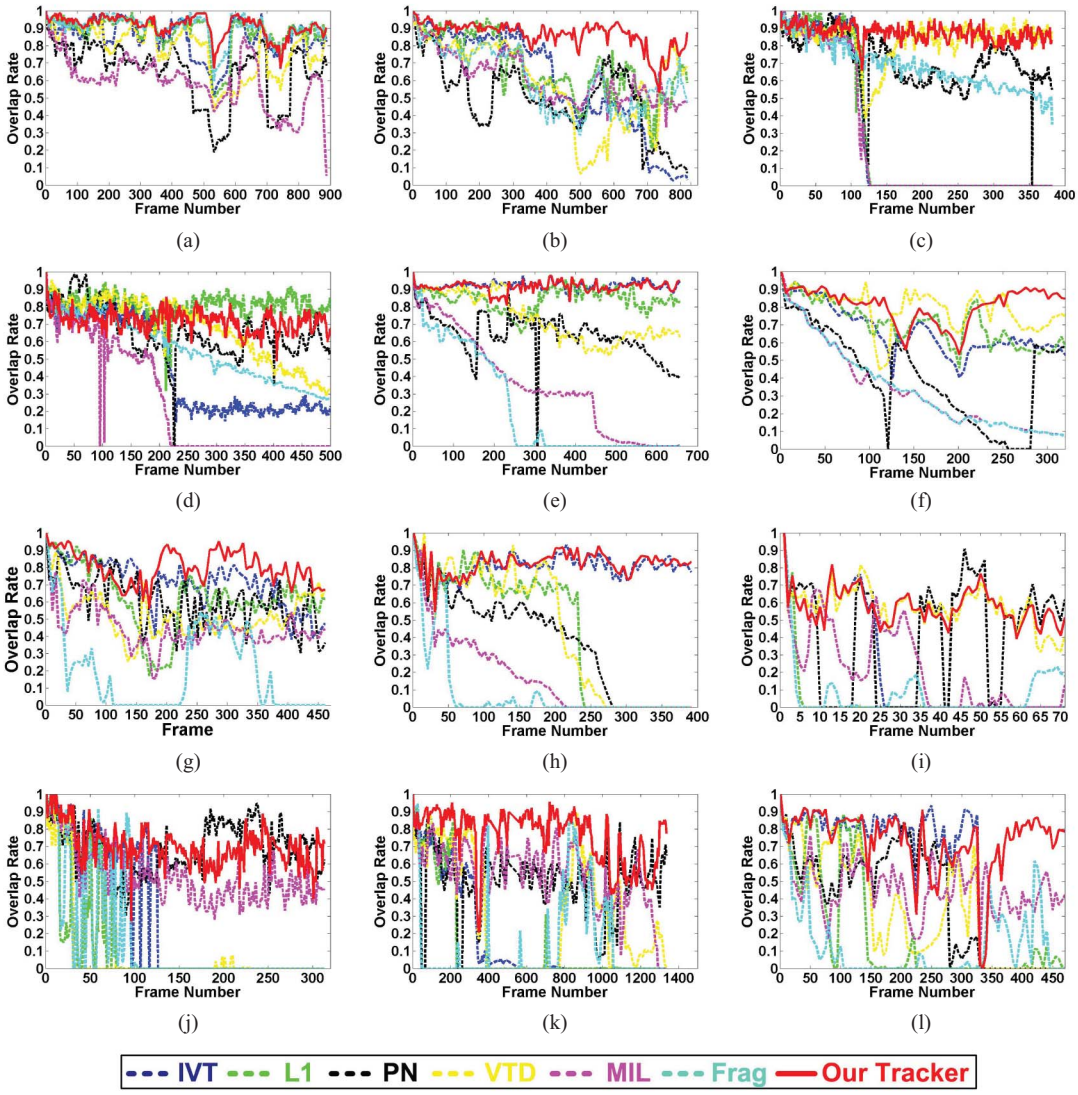

Fig. 9. Overlap rate evaluation. This figure shows overlap rates for twelve video clips we tested on. Our algorithm is compared with six state-of-the-art methods: IVT [5], L1 tacker [21], FragTrack [10], MILTrack [7], VTD [40], and PN methods [25]. (a) Occlusion 1. (b) Occlusion 2. (c) Caviar 1. (d) Caviar 2. (e) Car 4. (f) Singer 1. (g) David. (h) Car11. (i) Deer. (j) Jumping. (k) Lemming. (l) Cliffbar.

corresponding representations are denser. Thus, our tracker is able to distinguish the targets from their surrounding backgrounds. Furthermore, we note that the error term facilitates more accurate appearance update (with aligned samples).

B. Quantitative Evaluation

Performance evaluation is an important issue that requires sound criteria in order to fairly assess the strength of tracking algorithms. Quantitative evaluation of object tracking typically involves computing the difference between the predicated and the ground truth center locations, as well their average values. Table II summarizes the results in terms of average tracking errors. Our algorithm achieves lowest tracking errors in almost all the sequences. On the other hand, the tracking overlap rate indicates stability of each algorithm as it takes the size and pose of the target object into account. Given the tracking result of each frame R_T and the corresponding ground truth R_G, the overlap rate is defined by the PASCAL VOC [43] criterion, $score = \frac{area(R_T \cap R_G)}{area(R_T \cup R_G)}$. An object is regarded as being successfully tracked when the score is above 0.5. Figure 9 shows the overlap rates of each tracking algorithm for all the sequences and Table III presents the average overlap rates. Overall, our tracker performs favorably against the other algorithms.

While our work is different from sparse PCA (as mentioned in Section III), we also implement a SPCA based tracker within the same Bayesian framework. Given a Gram matrix G, the SPCA method [38] aims to compute sparse principal components, which only have a limited number of nonzero entries while capturing the maximum amount of variance

$$\begin{aligned} &\max \mathbf{u}^\top \mathbf{G}\mathbf{u} - \rho|\mathbf{u}|^2 \\ &\text{s.t. } \|\mathbf{u}\|_2 = 1 \end{aligned} \tag{12}$$

where $|\mathbf{u}|$ is the number of nonzero entries of $\mathbf{u}$ and ρ controls the sparsity of $\mathbf{u}$. In our experiments, ρ is empirically

TABLE II

AVERAGE CENTER ERROR (PIXELS). THE THIRD COLUMN FROM THE RIGHT SHOWS THE RESULTS FROM OUR IMPLEMENTED SPCA TRACKER, THE SECOND COLUMN FROM THE RIGHT SHOWS THE RESULTS USING (10), AND THE RIGHTMOST COLUMN SHOWS THE RESULTS USING (11) WITH OCCLUSION MASK

	IVT	ℓ_1	PN	VTD	MIL	FragTrack	SPCA	Ours Eq. 10	Ours Eq. 11
Occlusion 1	9.2	6.5	17.5	11.1	32.3	5.6	*4.4*	*5.3*	4.7
Occlusion 2	10.2	11.1	18.6	10.4	14.1	15.5	*10.4*	*17.9*	4.0
Caviar 1	45.1	119.9	5.6	3.9	48.5	5.7	*49.2*	*23.0*	1.7
Caviar 2	8.6	3.2	8.5	4.7	70.3	5.6	*30.5*	*65.1*	2.2
Car 4	2.9	4.1	18.8	12.3	60.1	179.8	*3.1*	*2.8*	3.0
Singer 1	8.5	4.6	32.7	4.1	15.2	22.0	*96.0*	*4.2*	4.7
David Indoor	3.6	7.6	9.7	13.6	16.2	76.7	*9.2*	*76.1*	3.7
Car 11	2.1	33.3	25.1	27.1	43.5	63.9	*1.6*	*2.3*	2.2
Deer	127.6	171.5	25.7	11.9	66.5	92.1	*152.8*	*121.7*	8.5
Jumping	36.8	92.4	3.6	63.0	9.9	58.5	*4.6*	*5.8*	5.0
Lemming	93.4	184.9	23.2	86.9	25.6	149.1	*25.0*	*58.1*	9.1
Cliffbar	24.8	24.8	11.3	34.6	13.4	48.7	*4.8*	*4.1*	3.5
Average	31.1	55.3	16.7	23.6	34.6	60.3	*32.6*	*32.2*	4.4

TABLE III

OVERLAP RATE OF TRACKING METHODS. THE THIRD COLUMN FROM THE RIGHT SHOWS THE RESULTS FROM OUR IMPLEMENTED SPCA TRACKER, THE SECOND COLUMN FROM THE RIGHT SHOWS THE RESULTS USING (10), AND THE RIGHTMOST COLUMN SHOWS THE RESULTS USING (11) WITH OCCLUSION MASK

	IVT	ℓ_1	PN	VTD	MIL	FragTrack	SPCA	Ours Eq. 10	Ours Eq. 11
Occlusion 1	0.85	0.88	0.65	0.77	0.59	0.90	*0.92*	*0.90*	0.91
Occlusion 2	0.59	0.67	0.49	0.59	0.61	0.60	*0.49*	*0.37*	0.84
Caviar 1	0.28	0.28	0.70	0.83	0.25	0.68	*0.28*	*0.28*	0.89
Caviar 2	0.45	0.81	0.66	0.67	0.25	0.56	*0.30*	*0.29*	0.71
Car 4	0.92	0.84	0.64	0.73	0.34	0.22	*0.92*	*0.92*	0.92
Singer 1	0.66	0.70	0.41	0.79	0.34	0.34	*0.26*	*0.84*	0.82
David Indoor	0.71	0.62	0.60	0.52	0.45	0.19	*0.47*	*0.76*	0.80
Car 11	0.81	0.44	0.38	0.43	0.17	0.09	*0.82*	*0.81*	0.81
Deer	0.22	0.04	0.41	0.58	0.21	0.08	*0.08*	*0.22*	0.61
Jumping	0.28	0.09	0.69	0.08	0.53	0.14	*0.70*	*0.67*	0.69
Lemming	0.18	0.13	0.49	0.35	0.53	0.13	*0.17*	*0.18*	0.75
Cliffbar	0.56	0.20	0.38	0.33	0.46	0.13	*0.74*	*0.76*	0.74
Average	0.54	0.48	0.54	0.56	0.39	0.33	*0.51*	*0.58*	0.79

set to 5, and the Gram matrix $\mathbf{G}$ is updated with new observation every 5 frames. We present the results of SPCA based tracker in Table II and Table III. The results show that our algorithm with Eq. 11 performs better than the SPCA based tracker, especially for some challenging sequences (e.g., *Occlusion 2*, *Singer 1*, *Deer*, *Lemming*) and surveillance videos (*Caviar 1* and *Caviar 2*). For presentation clarity, we put the tracking results of the proposed and SPCA based methods at http://faculty.ucmerced.edu/mhyang/video/prototype2.avi. We note that it takes about 5 seconds for the SPCA tracker requires to process each frame (using the algorithm proposed by d'Aspremont et al. [44]) since it is a time-consuming task to solve the optimization problem of Eq. 12.

To demonstrate how the occlusion map facilities object tracking and appearance update, we present the results using only Eq. 10 without occlusion map, and Eq. 11 with occlusion map in Table II and Table III. The results show that our algorithm is able to estimate occlusion maps effective, thereby further improving the tracking results both in terms of overlap rate and center location error. We note that the occlusion mask can be estimated reliably in the proposed algorithm whether the target object is occluded or not. Figure 10 shows some estimated occlusion maps and their use for model update.

If the target is well tracked and the occlusion rate is small, the tracking result is used to update the observation model directly (Figure 10(c) and (f)). When the tracked target suffers from partial occlusion (Figure 10(a) and (b)), the occlusion maps reflect this situation and the occlusion rates are consequently higher. Therefore, partial update prevents the tracker from inaccurate update. When the tracking results are not accurate (Figure 10(d), (e), and (i)), the occlusion rates are also consequently higher. In such cases, partial update is carried out to reduce the risk of inaccurate update (especially for Figure 10(d) and (i), some noise regions, shown in blue

TABLE IV

COMPUTATIONAL COMPLEXITY. THE CRITICAL STEPS OF IVT METHOD [5], ℓ_1 TRACKER [21] AND OUR ALGORITHM. IN THIS TABLE, d PRESENTS THE DIMENSION OF AN IMAGE OBSERVATION, k INDICATES THE NUMBER OF EIGENVECTORS OR TEMPLATES, $U \in R^{d\times k}$ STANDS FOR EIGENVECTORS CALCULATED BY PCA ($d \gg k$ IN THIS WORK), AND $A \in R^{d\times k}$ PRESENTS TEMPLATES OR SPARSE REPRESENTATION. THE LAST TWO COLUMNS PRESENT THE AVERAGE TIME FOR SOLVING ONE IMAGE PATCH (16×16 OR 32×32), WHERE $k = 16$

Algorithm	Aims	Computational complexity	time (16×16)	time (32×32)
IVT [5]	$\mathbf{z} = \mathbf{U}^\top \mathbf{y}$	$O(dk)$	0.11 ms	0.19 ms
ℓ_1 tracker [21]	$[\mathbf{z}, \mathbf{e}] = \arg\min_{\mathbf{z},\mathbf{e}} \frac{1}{2} \left\| \mathbf{y} - [\mathbf{A}, \mathbf{I}] \begin{bmatrix} \mathbf{z} \\ \mathbf{e} \end{bmatrix} \right\|_2^2 + \lambda \left\| \begin{bmatrix} \mathbf{z} \\ \mathbf{e} \end{bmatrix} \right\|_1$	$O\left(d^2 + dk\right)$	2.2 ms	248 ms
Our	$[\mathbf{z}, \mathbf{e}] = \arg\min_{\mathbf{z},\mathbf{e}} \frac{1}{2} \|\mathbf{y} - \mathbf{U}\mathbf{z} - \mathbf{e}\|_2^2 + \lambda \|\mathbf{e}\|_1$	$O(ndk)$	0.57 ms	1.5 ms

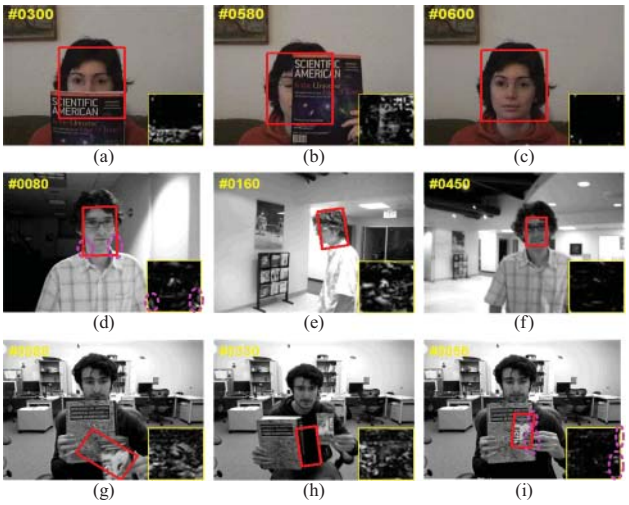

Fig. 10. Occlusion mask and appearance update. The estimated occlusion maps are shown in the lower right of each frame. The occlusion rate is used to determine whether a new observation is used for update or not. (a) 0.24, partial update. (b) 0.45, partial update. (c) 0.03, full update. (d) 0.33, partial update. (e) 0.52, partial update. (f) 0.05, full update. (g) 0.65, no update. (h) 0.67, no update. (i) 0.23, partial update.

circles, and backgrounds are factored out by occlusion maps before model update). If tracking returns are off the targets (Figure 10(g) and (h)), the occlusion rates are high and the observed image patches are discarded without update.

C. Computational Complexity

The most time consuming part of the evaluated tracking algorithms is to compute the coefficients using the basis vectors or templates. Table IV shows the computational complexity of the step for computing coefficients in the IVT method [5], ℓ_1 tracker [21] and the proposed algorithm. For the IVT method, the computation involves matrix-vector multiplication and the complexity is $O(dk)$. The computational complexity of the ℓ_1 tracker for computing the coefficients using the LASSO algorithm is $O\left(d^2 + dk\right)$. The computational load of our method is mainly in the step 3 of Table 1 (the cost of step 4 can be negligible) and the complexity is $O(ndk)$ where n is the number of iterations (e.g., 5 on average). While our tracker is much more efficient than the ℓ_1 tracker and slower than the IVT method, it achieves more favorable results in terms of center location error and overlap rate.

VI. CONCLUSION

This paper presents a robust tracking algorithm via the proposed sparse prototype representation. In this work, we explicitly take partial occlusion and motion blur into account for appearance update and object tracking by exploiting the strength of subspace model and sparse representation. Experiments on challenging image sequences demonstrate that our tracking algorithm performs favorably against several state-of-the-art algorithms. As the proposed algorithm involves solving ℓ_1 minimization problem for each drawn sample with the proposed model, we plan to explore more efficient algorithms for real-time applications. We will extend our representation scheme for other vision problems including object recognition, and develop other online orthogonal subspace methods (e.g., online NMF) with the proposed model. In addition, we plan to integrate multiple visual cues to better describe objects in different scenarios and to utilize prior knowledge with online learning for more effective object tracking.

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. Vision*, 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] H. Grabner and H. Bischof, "On-line boosting and vision," in *Proc. IEEE Conf. Comput. Vision Pattern Recogn.*, Jun. 2006, pp. 260–267.
[5] D. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vision*, vol. 77, nos. 1–3, pp. 125–141, 2008.
[6] S. Avidan, "Ensemble tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 2, pp. 261–271, Feb. 2007.
[7] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Proc. IEEE Conf. Comput. Vision Pattern Recogn.*, Jun. 2009, pp. 983–990.
[8] S. Wang, H. Lu, F. Yang, and M.-H. Yang, "Superpixel tracking," in *Proc. IEEE Int. Conf. Comput. Vision*, Nov. 2011, pp. 1323–1330.
[9] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Proc. IEEE Conf. Comput. Vision Pattern Recogn.*, Jun. 2012, pp. 1822–1829.
[10] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proc. IEEE Conf. Comput. Vision Pattern Recogn.*, Jun. 2006, pp. 798–805.
[11] D. Wang, H. Lu, and Y.-W. Chen, "Incremental MPCA for color object tracking," in *Proc. IEEE Int. Conf. Pattern Recogn.*, Aug. 2010, pp. 1751–1754.

[12] W. Hu, X. Li, X. Zhang, X. Shi, S. J. Maybank, and Z. Zhang, "Incremental tensor subspace learning and its applications to foreground segmentation and tracking" *Int. J. Comput. Vision*, vol. 91, no. 3, pp. 303–327, 2011.

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

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

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

[16] J. A. Lasserre, C. M. Bishop, and T. P. Minka, "Principled hybrids of generative and discriminative models," in *Proc. IEEE Conf. Comput. Vision Pattern Recogn.*, Jun. 2006, pp. 87–94.

[17] A. Y. Ng and M. I. Jordan, "On discriminative versus generative classifiers: A comparison of logistic regression and naive bayes," in *Proc. Adv. Neural Inform. Process. Syst.*, 2001, pp. 438–451.

[18] Q. Yu, T. B. Dinh, and G. G. Medioni, "Online tracking and reacquisition using co-trained generative and discriminative trackers," in *Proc. Eur. Conf. Comput. Vision*, 2008, pp. 678–691.

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

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

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

[22] T. Wang, I. Y. H. Gu, and P. Shi, "Object tracking using incremental 2-D-PCA learning and ml estimation," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Process.*, Apr. 2007, pp. 933–936.

[23] G. Li, D. Liang, Q. Huang, S. Jiang, and W. Gao, "Object tracking using incremental 2-D-LDA learning and bayes inference," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2008, pp. 1568–1571.

[24] P. A. Viola, J. C. Platt, and C. Zhang, "Multiple instance boosting for object detection," in *Adv. Neural Inform. Process. Syst.*, 2005, pp. 1681–1688.

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

[26] F. Yang, H. Lu, W. Zhang, and Y. Wei Chen, "Visual tracking via bag of features," *IET Image Process.*, vol. 6, no. 2, pp. 115–128, 2012.

[27] R. Fergus, F.-F. Li, P. Perona, and A. Zisserman, "Learning object categories from google's image search," in *Proc. IEEE Int. Conf. Comput. Vision*, Oct. 2005, pp. 1816–1823.

[28] X. Wang, T. X. Han, and S. Yan, "An HOG-LBP human detector with partial occlusion handling," in *Proc. IEEE Int. Conf. Comput. Vision*, Sep.–Oct. 2009, pp. 32–39.

[29] 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. Vision*, Jan. 2011, pp. 642–649.

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

[31] 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.

[32] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861–2873, Nov. 2010.

[33] J. Wright, Y. Ma, J. Maral, G. Sapiro, T. Huang, and S. Yan, "Sparse representation for computer vision and pattern recognition," *Proc. IEEE*, vol. 98, no. 6, pp. 1031–1044, Jun. 2010.

[34] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai, "Minimum error bounded efficient L1 tracker with occlusion detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recogn.*, 2011, pp. 1257–1264.

[35] E. J. Candès, Y. Eldar, D. Needell, and P. Randall, "Compressed sensing with coherent and redundant dictionaries," *Appl. Comput. Harmonic Anal.*, vol. 31, no. 1, pp. 59–73, 2010.

[36] E. T. Hale, W. Yin, and Y. Zhang, "Fixed-point continuation for ℓ_1-minimization: Methodology and convergence," *SIAM J. Opt.*, vol. 19, no. 3, pp. 1107–1130, 2008.

[37] I. Jolliffe, N. Trendafilov, and M. Uddin, "A modified principal component technique based on the LASSO," *J. Comput. Graphical Statist.*, vol. 12, no. 3, pp. 531–547, 2003.

[38] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *J. Comput. Graphical Statist.*, vol. 15, no. 2, pp. 265–286, 2006.

[39] R. Zass and A. Shashua, "Nonnegative sparse PCA," in *Proc. Adv. Neural Inform. Process. Syst.*, 2007, pp. 1561–1568.

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

[41] [Online]. Available: http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/

[42] P. N. Belhumeur and D. J. Kriegman, "What is the set of images of an object under all possible illumination conditions?" *Int. J. Comput. Vision*, vol. 28, no. 3, pp. 245–260, 1998.

[43] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. (2010). *The PASCAL Visual Object Classes Challenge Results* [Online]. Available: http://pascallin.ecs.soton.ac.uk/challenges/VOC/

[44] A. Aspremont, L. El Ghaoui, M. Jordan, and G. Lanckriet, "A direct formulation for sparse pca using semidefinite programming," *SIAM Rev.*, vol. 49, no. 3, pp. 434–448, 2007.

Dong Wang received the B.E. degree from the Dalian University of Technology (DUT), Dalian, China, in 2008, where he is currently pursuing the Ph.D. degree with the School of Information and Communication Engineering.

His current research interests include face recognition, interactive image segmentation, and object tracking.

Huchuan Lu (M'06) received the M.Sc. degree in signal and information processing and the Ph.D. degree in system engineering from the Dalian University of Technology (DUT), Dalian, China, in 1998 and 2008, respectively.

He has been a Faculty Member, since 1998, and a Professor, since 2012, with the School of Information and Communication Engineering, DUT. He focuses on visual tracking and segmentation. His current research interests include computer vision and pattern recognition.

Dr. Lu was a member of ACM in 2006 and 2010.

Ming-Hsuan Yang (SM'06) received the Ph.D. degree in computer science from the University of Illinois at Urbana-Champaign, Urbana, in 2000.

He was a Senior Research Scientist with Honda Research Institute, working on vision problems related to humanoid robots. He is an Assistant Professor with the Department of Electrical Engineering and Computer Science, the University of California (UC), Merced. He has co-authored the book *Face Detection and Gesture Recognition for Human-Computer Interaction* (Kluwer 2001) and edited the special issue on face recognition for *Computer Vision and Image Understanding*, in 2003.

Dr. Yang was a recipient of the Ray Ozzie fellowship for his research work in 1999. He received the Natural Science Foundation CAREER Award in 2012, the Campus Wide Senate Award for Distinguished Early Career Research at UC in 2011, and the Google Faculty Award in 2009. He edited a special issue on real world face recognition for the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. He serves as an Area Chair for the IEEE International Conference on Computer Vision in 2011, the IEEE Conference on Computer Vision and Pattern Recognition in 2008 and 2009, Asian Conference on Computer in 2009, 2010, and 2012. He served as an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE from 2007 to 2011, and the *Image and Vision Computing*. He is a Senior Member of the ACM.