



# Object tracking via appearance modeling and sparse representation <sup>☆</sup>

Feng Chen <sup>a,\*</sup>, Qing Wang <sup>a,\*</sup>, Song Wang <sup>b</sup>, Weidong Zhang <sup>c</sup>, Wenli Xu <sup>a</sup>

<sup>a</sup> Tsinghua National Laboratory for Information Science and Technology, Department of Automation, Tsinghua University, Beijing 100084, China

<sup>b</sup> Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29208, USA

<sup>c</sup> National Institutes of Health, Clinical Center, Bethesda, MD 20892, USA

## ARTICLE INFO

### Article history:

Received 31 January 2011

Received in revised form 26 June 2011

Accepted 29 August 2011

### Keywords:

Target variation

Online appearance modeling

Sparse representation

Bayesian inference

## ABSTRACT

This paper proposes a robust tracking method by the combination of appearance modeling and sparse representation. In this method, the appearance of an object is modeled by multiple linear subspaces. Then within the sparse representation framework, we construct a similarity measure to evaluate the distance between a target candidate and the learned appearance model. Finally, tracking is achieved by Bayesian inference, in which a particle filter is used to estimate the target state sequentially over time. With the tracking result, the learned appearance model will be updated adaptively. The combination of appearance modeling and sparse representation makes our tracking algorithm robust to most of possible target variations due to illumination changes, pose changes, deformations and occlusions. Theoretic analysis and experiments compared with state-of-the-art methods demonstrate the effectivity of the proposed algorithm.

© 2011 Elsevier B.V. All rights reserved.

## 1. Introduction

Object tracking is very important in the field of computer vision. It plays a key role in many applications such as intelligent surveillance, human–computer interface, traffic control and vehicle navigation. The main challenge in designing a robust tracking algorithm is the inevitable appearance variations of the target over time, which can be caused by many intrinsic and extrinsic reasons, such as pose changes, scale changes, significant illumination variations, partial occlusions and so on. Therefore, to achieve reliable and accurate tracking, it is critical to design a flexible representation which can adaptively model the object appearance variations.

In this paper, we propose an effective online appearance modeling method to account for the target appearance variations during tracking. Although the appearance manifold of an object may be quite non-linear and complex, we make a reasonable assumption that the manifold can be approximated by multiple linear models. With this assumption, we learn multiple linear subspaces to model the target appearance variations during tracking. In order to make our tracking algorithm more robust to abrupt target appearance changes due to occlusion or image corruption, we combine the learned appearance model and the sparse representation [1] to further learn a similarity measure for distinguishing the target object from background. More specifically, when abrupt appearance variation occurs, the target appearance can be reconstructed by the learned appearance model

and a limited number of noise bases. Tracking is then achieved by Bayesian inference, in which a particle filter is adopted to estimate the target state sequentially. With the tracking results in new frames, we update the appearance model adaptively.

Compared to other tracking methods, the proposed method shows two improvements in dealing with appearance variations of the target. First, in our method, different kinds of target observations can be modeled by different subspaces of the constructed appearance model. When the target shows a large appearance change, new subspaces will be added to cover it. Second, we combine the learned multiple subspaces and sparse representation to deal with abrupt appearance changes in the proposed method. A target object can be always represented by the learned appearance model and some additional noise bases even when the appearance of the target changes abruptly.

The rest of the paper is organized as follows. Section 2 reviews the related works. Section 3 gives an overview of the proposed tracking algorithm. Section 4 introduces the proposed appearance modeling and sparse representation algorithm. The proposed tracking algorithm is presented in Section 5. Section 6 reports the experimental results. The paper is concluded in Section 7.

## 2. Related work

There have been numerous tracking algorithms in the literature. Tracking can be casted as an optimization problem. Given an appropriate cost function, the tracking task can be implemented by minimizing the cost function. Sum of Squared Difference (SSD) is widely used in many tracking algorithms, e.g., optical flow approaches [2]. An alternative cost function derived from the color histogram is utilized in the popular mean shift tracking algorithm [3]. In these

<sup>☆</sup> This paper has been recommended for acceptance by Stefanos Zafeiriou.

\* Corresponding authors.

E-mail addresses: [chenfeng@mails.tsinghua.edu.cn](mailto:chenfeng@mails.tsinghua.edu.cn) (F. Chen), [qing-wang07@mails.tsinghua.edu.cn](mailto:qing-wang07@mails.tsinghua.edu.cn) (Q. Wang).

methods, the gradient descent algorithm is usually used to find the optimal solutions. Tracking can also be casted as a state estimation problem. Early works such as [4] used Kalman filter for object tracking. However the dynamic and observation models in these methods are restricted to linear Gaussian ones. Recently, particle filters have attracted much attention in the tracking field because of their capabilities in modeling nonlinear/non-Gaussian cases [5]. With a particle filter, the target state can be estimated by computing the posterior probability of sample distributions.

As an essential component in all tracking methods, object representation recently gets more and more attention because it plays a key role in determining the tracking performance. A good representation should have strong description or discrimination power to distinguish the target from the background. To account for the appearance variations of the target during tracking, many sophisticated object representation methods have been developed, including both generative and discriminative methods.

For generative appearance modeling methods, Black et al. [6] propose a tracking algorithm based on the subspace constancy assumption. They construct a subspace model offline with some collected target observations, and keep the model fixed during tracking. However, for most tracking applications, it may be difficult to obtain such target observations in advance. As a result, this method has limited application domains and may fail when the target undergoes a different view from the ones used for constructing the model. Recently, many adaptive appearance models have been proposed for object tracking. Jepson et al. [7] propose a wavelet-based mixture model via an online *Expectation–Maximization* (EM) algorithm to account for appearance variations of the target during tracking. Incremental subspace methods based on *Principal Component Analysis* (PCA) or its variants have also been used for online object representation [8,9]. These two methods use target observations obtained online to learn a linear subspace for object representation. Since the appearance manifold of a target in a long-time interval may be quite nonlinear and complex, these models may not describe the appearance variations of the target well. Moreover, grossly corrupted observations often hurt the validity of the learned subspaces [10]. In addition, when an abrupt appearance variation of the target (due to changes such as lighting variation, pose variation or occlusion) occurs, the first few observations of the target will be identified as outliers and will not be fit into such linear appearance models. Therefore, these linear-subspace methods will run into the danger of losing the target as time progresses. Different from these subspace methods based on PCA, Ho et al. [11] propose a subspace learning method based on uniform L2 optimization. During tracking, they divide the most recent observations of the target into several batches and use the means of these batches for appearance modeling. However, this method only preserves the most recent appearance information of the target, and is sensitive to the chosen size of batches. When the batch size is not appropriate, the learned subspace model will be inaccurate and the resulting tracking may be in danger of drifting.

For discriminative appearance modeling methods, Collins et al. [12] propose an online feature selection method to select the most discriminative color spaces for tracking. Avidan et al. [13] use online

boosting method for tracking. They propose an ensemble tracking framework, in which a set of weak classifiers are trained to construct a strong classifier to distinguish the target from the background. Parag et al. [14] also utilize the online boosting algorithm for tracking, but propose a different method for classifier updating. Babenko et al. [15] use *Multiple Instance Learning* (MIL) instead of traditional supervised learning to avoid the inaccuracy accumulation problem caused by self-learning. In these methods, tracking is usually treated as a binary classification problem. Different from the generative models which only model the target, the discriminative models can model both the target and background. Although these discriminative tracking methods have the capability to select good features for tracking, they need correctly labeled samples to train and update classifiers, which may not be available in many real tracking applications.

Recently, a sparse representation framework [1] is proposed to recover the appearance subspace when the corruption of the observation is gross but sparse. In this framework, most or all information of a signal can be represented by a linear combination of a small number of elementary signals in an over-complete signal dictionary. This theory offers a new insight for solving object occlusion and image corruption problems, and has been successfully used in face recognition [16]. Sparse representation has also been used in visual tracking. In [17], a target template set is chosen in the first frame and is updated using new tracking results. However, the template set cannot model the appearance manifold of the target well without considering most of the obtained target observations. Furthermore, this algorithm is sensitive to tracking inaccuracies: When an undesirable template is added to the template set, the entire tracking algorithm will be in danger.

Different from the above methods, we propose an object representation method combining both online appearance modeling and sparse representation. The learned subspace models can account for most of the appearance variations of the target during tracking. The sparse representation makes our method more robust to abrupt appearance changes such as partial occlusions. When a tracking result is obtained, the basis set for sparse representation is updated by our online appearance modeling method.

### 3. Overview of the proposed tracking algorithm

We propose a tracking algorithm based on the combination of appearance modeling and sparse representation. As shown in Fig. 1, the proposed algorithm consists of four steps:

1. Initialization. Choosing the tracking target in the first frame manually or by an automatic detector.
2. Online appearance modeling. Target in each frame is described by the raw intensity values inside the target region. The appearance model is initialized in the first frame, and will be updated online when new tracking results are obtained.
3. Sparse representation. With the learned appearance model, we construct an over-complete basis set for target representation. Using the reconstruction error based on this over-complete basis set, we define a similarity measure to evaluate the distance between a target candidate and the learned appearance model.

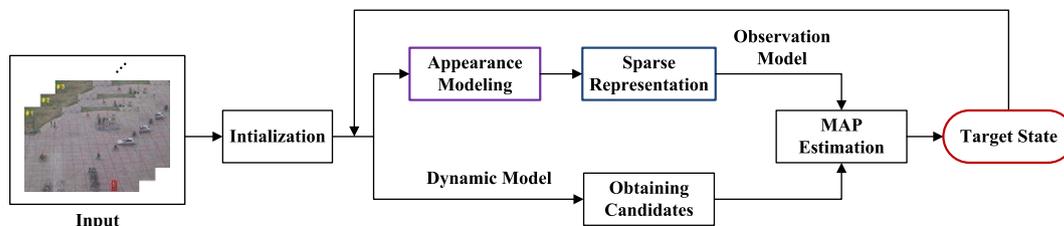


Fig. 1. Flow chart of the proposed tracking algorithm.

4. Bayesian inference for tracking. Tracking is formulated as Bayesian inference, in which a particle filtering method is applied to estimate the target state sequentially over time. With the learned similarity measure, we use the *Maximum a Posterior* (MAP) estimation to define the observation model and estimate the target state in each frame. After this step, we go back to step 2 to update the appearance model using the current target observation.

#### 4. Online appearance modeling and sparse representation

In this section, we firstly propose an appearance modeling approach, which considers the nonlinearity of the target appearance manifold and can preserve most of the target appearance variations during tracking. Then using sparse representation, we define a similarity measure for tracking with the reconstruction error on the learned appearance model.

##### 4.1. Appearance modeling

Although the appearance manifold of an object may be quite nonlinear and complex, we can still make a reasonable assumption that the appearance manifold can be approximated by multiple linear models. It is because the observations of a target from consecutive frames are temporally correlated and may repeat over time. As illustrated in Fig. 2, to represent the target appearance manifold, we learn multiple linear subspace models, which is different from appearance modeling methods based on a single linear subspace such as [8].

In our method, the observation of the object in each frame is described by the raw pixel intensity values inside the target region. Let  $\mathbf{x}_t \in R^d$  denote the observation (feature vector) of the object at time  $t$ ,  $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \in R^{d \times n}$  denote an observation set of the target (where  $d$  is number of pixels inside the target region). Suppose the entire appearance manifold constructed by  $X$  can be represented by  $K$  linear subspaces. We first cluster  $X$  into  $K$  sub-sets  $X = \{X_1, \dots, X_K\}$ , where  $X_i \in R^{d \times n_i}$ , and  $n_i$  is the number of observations in the  $i$ th sub-set with  $\sum_{i=1}^K n_i = n$ . Many existing clustering methods can be used to obtain the  $K$  sub-sets. For simplicity, here we describe our appearance modeling method by assuming that the set of  $n$  target observations is available. Later in Section 4.3, we will elaborate the adaptive learning of the appearance model, starting from a single target observation in the first frame.

For the  $i$ th sub-set  $X_i$ , we learn a linear subspace to represent it. To eliminate the influence caused by the mean of training data, we first center  $X_i$  to:

$$X_i^c = X_i - \mathbf{m}_i \mathbf{1}_{1 \times n_i} \tag{1}$$

where  $\mathbf{m}_i \in R^{d \times 1}$  is the mean of the  $n_i$  observations in  $X_i$ , and  $\mathbf{1}_{1 \times n_i} = (\underbrace{1, \dots, 1}_{n_i})$ . With  $X_i^c$ , our goal is to find a subspace (spanned by a projection matrix  $U_i = \{\mathbf{u}_i^1, \mathbf{u}_i^2, \dots, \mathbf{u}_i^{p_i}\} \in R^{d \times p_i}$ ) which can preserve most variations in  $X_i^c$ . Therefore, the objective function can be defined as follows:

$$\begin{aligned} \max_{\mathbf{u}_i^1, \dots, \mathbf{u}_i^{p_i}} \quad & \mathbf{u}_i^T S_i \mathbf{u}_i \\ \text{s.t.} \quad & \\ \mathbf{u}_i^{jT} \mathbf{u}_i^k = 1, \quad & \mathbf{u}_i^{jT} \mathbf{u}_i^k = 0 (j \neq k, j, k = 1, 2, \dots, p_i), \end{aligned} \tag{2}$$

where  $S_i$  is the scatter matrix of  $X_i^c$ , which is defined as:

$$S_i = X_i^c X_i^{cT} \tag{3}$$

It is well known that the solutions  $\{\mathbf{u}_i^1, \dots, \mathbf{u}_i^{p_i}\}$  of Eq. (2) are the orthogonal eigenvectors of  $S_i$  corresponding to the  $p_i$  largest eigenvalues of  $S_i$ . Actually, we can directly decompose the matrix  $X_i^c$  using the *singular value decomposition* (SVD) method to get  $\{\mathbf{u}_i^1, \dots, \mathbf{u}_i^{p_i}\}$ , which correspond to the  $p_i$  largest singular values  $\{\xi_i^1, \xi_i^2, \dots, \xi_i^{p_i}\}$  of  $X_i^c$ . We denote the singular value matrix as  $\Sigma_i = \text{diag}(\xi_i^1, \xi_i^2, \dots, \xi_i^{p_i})$  in this paper.  $p_i$  is the number of principal components and can be chosen by

$$p_i = \arg \min_k \left\{ \frac{\sum_{j=1}^k \xi_i^j}{\sum_{j=1}^{\min\{d, n_i\}} \xi_i^j} \geq r, \quad k = 1, \dots, \min\{d, n_i\}, \quad 0 \leq r \leq 1 \right\}, \tag{4}$$

where  $r$  is a predefined threshold. A larger  $r$  will lead to a larger  $p_i$ , and a subspace that preserves more variation of the original target observations. The upper bound of  $p_i$  is  $\min\{d, n_i\}$ . The learned subspace from  $X_i$  can be represented by the mean  $\mathbf{m}_i$  and the projection matrix  $U_i$ . Given a test sample  $\mathbf{x}$  and the centering result  $\mathbf{x}^c = \mathbf{x} - \mathbf{m}_i$ , the projection of  $\mathbf{x}^c$  onto the subspace can be computed as:

$$\mathbf{y} = U_i^T \mathbf{x}^c \tag{5}$$

When  $p_i < \min\{d, n_i\}$ , there is a dimensionality reduction and only the principal variation of  $\mathbf{x}$  is preserved after this projection. The reconstruction of  $\mathbf{x}$  using this subspace is

$$\hat{\mathbf{x}} = U_i \mathbf{y} + \mathbf{m}_i \tag{6}$$

For  $K$  sub-sets  $\{X_1, X_2, \dots, X_K\}$ , we compute  $K$  linear subspaces. Then the appearance model learned from  $X$  can be denoted by  $\mathcal{A} = \{A_1, \dots, A_K\}$ , where  $A_i = \{\mathbf{m}_i, U_i\}$ ,  $i = 1, \dots, K$ . This learned appearance model covers different observations of the object. During

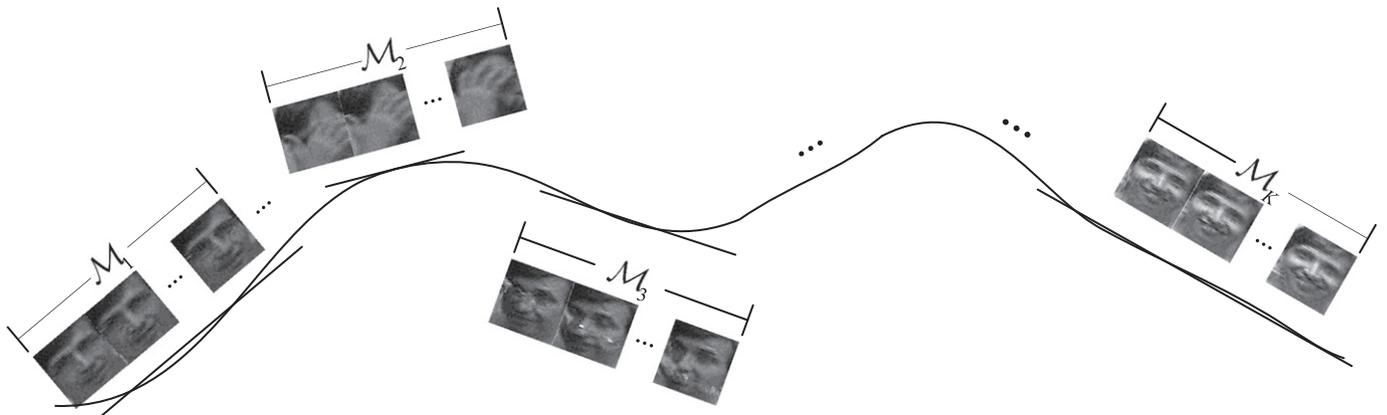


Fig. 2. The appearance manifold of the target object is represented by multiple linear models.

tracking, if the appearance of the target changes smoothly, a new observation of the target can be well reconstructed by one subspace (e.g.,  $A_i$ ) in the learned appearance model  $\mathcal{A}$ . We can then use the reconstruction error as the similarity measure between a test sample and the learned appearance model to get the tracking result like in [8,9]. However, when abrupt appearance changes occur (e.g., partial occlusion), there will be no proper subspace in  $\mathcal{A}$  which can reconstruct the corrupted target observation well. In such a case, it is difficult for the learned appearance model to get good tracking result. To deal with this problem, we further introduce a sparse representation technique to learn a similarity measure in the next section.

#### 4.2. Learning a similarity measure by sparse representation

In this section, we make use of the sparse representation framework [16,1] to generate a robust similarity measure for tracking. As mentioned in Section 2, with the sparse representation framework, an occluded or corrupted target observation can be well reconstructed by the learned appearance model and some noise bases. Sparse representation has been used for object tracking in [17], where some instances of the target observations are used to construct an over-complete basis set. Different from [17], we construct the over-complete basis set with the learned appearance model:

$$B = [A, E], \quad (7)$$

where  $E = \{I, -I\} \in \mathbb{R}^{d \times 2d}$  represents a trivial basis set for the noise resulting from target occlusion or image corruption.  $I \in \mathbb{R}^{d \times d}$  is an identity matrix.  $A$  comes from the learned appearance model  $\mathcal{A}$ , in the form of

$$A = [m'_1, U_1, m'_2, U_2, \dots, m'_K, U_K], \quad (8)$$

where  $m'_i = \frac{\mathbf{m}_i}{\|\mathbf{m}_i\|}$  ( $i=1, \dots, K$ ) is the normalized mean of the  $i$ th subspace in the learned appearance model  $\mathcal{A}$ . Compared to the basis set used in [17], ours contains more information of the target appearance variations and can represent the target object better. With the over-complete basis set  $B$ , a test sample  $\mathbf{x}$  can be represented by:

$$\mathbf{x} = B\mathbf{c}, \quad (9)$$

where  $\mathbf{c}$  is the reconstruction coefficient vector on the basis set  $B$ . Eq. (9) can be also written as:

$$\mathbf{x} = B\mathbf{c} = [A, E] \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} = A\mathbf{a} + E\mathbf{b}, \quad (10)$$

where  $\mathbf{a}$  is the reconstruction coefficient vector on  $A$  and  $\mathbf{b}$  is coefficient vector on  $E$ . Fig. 3 shows an example of sparse representation, where a test sample is exactly represented by one subspace in  $\mathcal{A}$  and a noise term. Corresponding to Eq. (10),  $\mathbf{a} = [1, a^1, \dots, a^{p_i}]$  and  $E\mathbf{b} = \zeta$ . In this ideal case, all the representation coefficients of other subspaces in  $\mathcal{A}$  are zeros.

By assuming that a good target candidate can be well represented by a certain subspace  $A_i$  in  $\mathcal{A}$  and some noise bases, we

solve the following  $\ell_0$  optimization objective function for optimal coefficients  $\mathbf{c}$ :

$$\begin{aligned} \mathbf{c}^* &= \arg \min_{\mathbf{c}} \|\mathbf{c}\|_0 \\ \text{s.t.} \\ \mathbf{x} &= B\mathbf{c}, \end{aligned} \quad (11)$$

where  $\|\cdot\|_0$  counts the number of nonzero entries in coefficient vector  $\mathbf{c}$ . However, the  $\ell_0$  minimization is an NP-hard problem. Recent development in the emerging theory of sparse representation and compressive sensing [18] reveals that if  $\mathbf{c}$  is sparse enough, the solution of the  $\ell_0$  minimization problem is equal to the solution of the following  $\ell_1$  minimization problem:

$$\begin{aligned} \mathbf{c}^* &= \arg \min_{\mathbf{c}} \|\mathbf{c}\|_1 \\ \text{s.t.} \\ \mathbf{x} &= B\mathbf{c}. \end{aligned} \quad (12)$$

The objective function (12) can be solved in polynomial time by standard linear programming algorithms. By allowing certain reconstruction errors, we can instead solve the Lagrange optimization problem:

$$\mathbf{c}^* = \arg \min_{\mathbf{c}} \|\mathbf{x} - B\mathbf{c}\|_2^2 + \lambda \|\mathbf{c}\|_1, \quad (13)$$

where  $\lambda$  is a parameter that balances reconstruction error and sparsity. With the solution  $\mathbf{c} = \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}$ , we can construct a robust similarity measure to evaluate the distance between a target candidate and the learned appearance model.

More specifically, the candidate which belongs to the target class should be well approximated by  $A$  and some noise bases even when some occlusion or image corruption occurs. As a result, the coefficient vector  $\mathbf{c}$  should be sparse for good target candidates. In contrast, a sample from the background can't be reconstructed well using the basis set  $B$  using this objective function. The reconstruction error of a test sample  $\mathbf{x}$  based on the over-complete basis set can be computed as

$$RE = \|\mathbf{x} - B\mathbf{c}^*\|_2 \quad (14)$$

We use  $RE$  as the similarity measure between a target candidate and the learned appearance model. In the following sections, we will discuss how to use this similarity measure to obtain tracking result in each frame.

#### 4.3. Updating the appearance model

The target and background appearances may change due to many factors such as illumination changes, pose changes and occlusions. A fixed appearance model is not sufficient to handle appearance changes during tracking. Intuitively, target appearance remains the same only for a certain period of time and an initial appearance model will become more and more inaccurate over time. If we do not update the initial appearance model, the tracking system will not be able to capture the variations of target and background. To

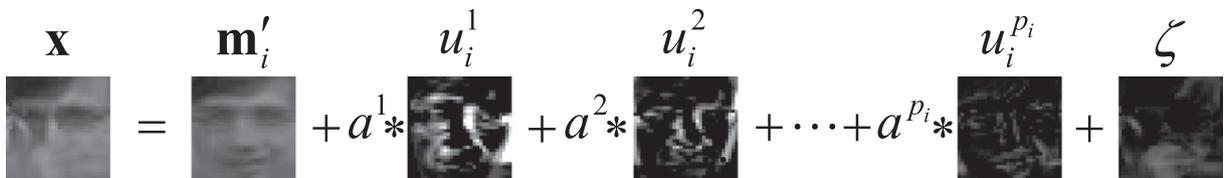


Fig. 3. An illustration of sparse representation.  $\mathbf{m}'_i$  is the normalized mean of the  $i$ th subspace.  $U_i = \{u_i^1, \dots, u_i^{p_i}\}$  is the projection matrix of the  $i$ th subspace.  $\zeta$  is the reconstruction error.

deal with this problem, we update our appearance model adaptively when new tracking results are obtained.

Suppose the current appearance model is  $\mathcal{A} = \{A_1, \dots, A_K\}$ . When the tracking result at time  $t$  is obtained, we use its corresponding observation  $\mathbf{x}_t$  to update  $\mathcal{A}$ . If  $\mathbf{x}_t$  belongs to certain subspace (denoted by  $A_i$ ) in  $\mathcal{A}$ , we will use it to update  $A_i$ . If not, we will add a new subspace to  $\mathcal{A}$ , and delete one existing subspace from  $\mathcal{A}$  simultaneously. In the step of sparse representation, we have calculated the coefficient vector  $\mathbf{a}$ . The subspace  $A_i$  which has the largest reconstruction coefficients (as discussed above, the reconstruction coefficients of other subspaces are close to zeros) and the subspace  $A_j$  which has the smallest reconstruction coefficients will be selected. First, we calculate the reconstruction error by  $A_i$ :

$$RE_i = \|\mathbf{x}_t - A_i' \mathbf{a}_i\|^2 \quad (15)$$

where  $A_i' = [\mathbf{m}_i', U_i]$ , and  $\mathbf{a}_i$  is the reconstruction coefficient vector corresponding to  $A_i$ , which is a subset of  $\mathbf{a}$ . If  $RE_i$  is larger than certain threshold, we will construct a new subspace and use it to replace the subspace  $A_j$ . The mean of this new subspace is  $\mathbf{m}_j^* = \mathbf{x}_t$ , the projection matrix is initialized as  $U_j^* = \begin{Bmatrix} \mathbf{x}_t \\ \mathbf{x}_t \end{Bmatrix}$ , and  $A_j^* = \{\mathbf{m}_j^*, U_j^*\}$ .

When  $RE_i$  is smaller than certain threshold (which means  $\mathbf{x}_t$  can be well represented in the subspace  $A_i$ ), we need to adjust  $A_i$  with  $\mathbf{x}_t$ . Supposing the observation sub-set from which the current  $A_i$  is learned is  $X_i = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{n_i}\}$ , we update this target observation sub-set  $X_i$  to  $X_i^* = \{X_i, \mathbf{x}_t\} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{n_i}, \mathbf{x}_t\}$ . With this new  $X_i^*$ , we can learn a new subspace model  $A_i^*$ . Since we have known the mean  $\mathbf{m}_i$  and projection matrix  $U_i$  learned from  $X_i$ , the updating problem can be regarded as retraining the mean  $\mathbf{m}_i^*$  and projection matrix  $U_i^*$  when  $\mathbf{x}_t$  is obtained. To make full use of the orthogonal information within the previous projection matrices  $U_i$ , we use the recursive SVD [8,19,20] method for this retraining.

The entire procedure of the proposed online appearance modeling algorithm is summarized in Algorithm 1. Note that in this algorithm, we only need to store the appearance model  $\mathcal{A}$  (including data means, projection matrices and the corresponding singular value matrices) in memory and do not need to store the target observation set. Moreover, we initialize an appearance model with a single target observation in the first frame and update it continuously when tracking from frame to frame.

### 5. Tracking by Bayesian inference

With the learned appearance model, we formulate the state estimation problem of object tracking within the Bayesian inference framework. A Markov model with hidden state variable is used to estimate the tracking result. Given the observation set of the target up to time  $t$  to be  $\mathbf{x}_{1:t} = \{\mathbf{x}_1, \dots, \mathbf{x}_t\}$ , the target state (tracking result)  $\mathbf{s}_t$  can be determined by the *Maximum A Posteriori* (MAP) estimation:

$$\hat{\mathbf{s}}_t = \arg \max_{\mathbf{s}_t} p(\mathbf{s}_t | \mathbf{x}_{1:t}), \quad (16)$$

where  $p(\mathbf{s}_t | \mathbf{x}_{1:t})$  can be inferred by the Bayesian theorem in a recursive manner:

$$p(\mathbf{s}_t | \mathbf{x}_{1:t}) \propto p(\mathbf{x}_t | \mathbf{s}_t) p(\mathbf{s}_t | \mathbf{x}_{1:t-1}), \quad (17)$$

where  $p(\mathbf{s}_t | \mathbf{x}_{1:t-1}) = \int p(\mathbf{s}_t | \mathbf{s}_{t-1}) p(\mathbf{s}_{t-1} | \mathbf{x}_{1:t-1}) d\mathbf{s}_{t-1}$ . We can see that the tracking process is governed by the *dynamic model*  $p(\mathbf{s}_t | \mathbf{s}_{t-1})$  which delineates the temporal correlation of the target states in consecutive frames, and the *observation model*  $p(\mathbf{x}_t | \mathbf{s}_t)$  which denotes the likelihood of  $\mathbf{s}_t$  generating observation  $\mathbf{x}_t$ .

A particle filter [5] is adopted in this paper to estimate the target state. Using particle filter, the posterior  $p(\mathbf{s}_t | \mathbf{x}_{1:t})$  is approximated by

a finite set of samples  $\{\mathbf{s}_t^i, i = 1, \dots, N_s\}$  with importance weights  $\{\omega_t^i, i = 1, \dots, N_s\}$ . The candidate sample  $\mathbf{s}_t^i$  is drawn from an importance distribution  $q(\mathbf{s}_t | \mathbf{s}_{1:t-1}, \mathbf{x}_{1:t})$  and the weight of the  $i$ th sample is:

$$\omega_t^i = \omega_{t-1}^i \frac{p(\mathbf{x}_t | \mathbf{s}_t^i) p(\mathbf{s}_t^i | \mathbf{s}_{t-1}^i)}{q(\mathbf{s}_t^i | \mathbf{s}_{1:t-1}^i, \mathbf{x}_{1:t})}. \quad (18)$$

**Algorithm 1.** The online appearance modeling algorithm.

**Input:** (1) The appearance model  $\mathcal{A} = \{A_1, \dots, A_K\}$ , where  $A_i = \{\mathbf{m}_i, U_i\}$ .  
 (2) The corresponding singular value matrices  $\{\Sigma_1, \dots, \Sigma_K\}$  of the appearance model  $\mathcal{A}$ . (3) The new target observation  $\mathbf{x}_t$ .

**Output:** The new appearance model  $\mathcal{A}^*$ .

**for**  $t = 1 : \text{FrameNumber}$  **do**

1. Find the subspace  $A_i$  in  $\mathcal{A}$  which has the largest coefficients in  $\mathbf{a}$ . Find the subspace  $A_j$  in  $\mathcal{A}$  which has the smallest coefficients in  $\mathbf{a}$ ;
2. Compute the distance between  $\mathbf{x}_t$  and  $A_i$ , which is denoted by  $RE_i$ ;

**if**  $RE_i < Th$  (some predefined threshold, e.g.,  $1/2 \|\mathbf{x}_t\|^2$ )

1. Update the mean of the  $i$ th subspace  $\mathbf{m}_i^* = \frac{f n}{f n + 1} \mathbf{m}_i + \frac{1}{f n + 1} \mathbf{x}_t$ , where  $f$  is a forgetting factor;
2. Center the new observation as  $\mathbf{x}_t^c = \mathbf{x}_t - \mathbf{m}_i^*$ ;
3. Compute the added information  $\bar{E} = \left\{ \mathbf{x}_t^c, \sqrt{\frac{n}{n+1} (\mathbf{m}_i - \mathbf{x}_t)} \right\}$ ;
4. Compute the projection matrices  $U_i^*$  and related singular value matrices  $\Sigma_i^*$  using the recursive SVD algorithm with the forgetting factor  $f$ ;

**else**

Use  $\mathbf{x}_t$  to construct a new subspace and use this subspace to replace  $A_j$  in  $\mathcal{A}$ . This new subspace model is initialized as:  $A_j^* = \{\mathbf{m}_j^*, U_j^*\}$ , where  $\mathbf{m}_j^* = \mathbf{x}_t$ ,  $U_j^* = \frac{\mathbf{x}_t}{\|\mathbf{x}_t\|}$ .

The samples are resampled to generate an unweighted particle set according to their importance weights to avoid degeneracy. In this paper, we choose  $q(\mathbf{s}_t | \mathbf{s}_{1:t-1}, \mathbf{x}_{1:t}) = p(\mathbf{s}_t | \mathbf{s}_{t-1})$  and the weight  $\omega_t^i$  becomes the observation likelihood  $p(\mathbf{x}_t | \mathbf{s}_t^i)$ .

#### 5.1. Dynamic model

Within a particle filter, the dynamic model can help generate the hypotheses set and save computations but may not have fundamental impact on the tracking performance. In our algorithm,  $p(\mathbf{s}_t | \mathbf{s}_{t-1})$  is assumed to be a simple smoothness model. We approximate the motion of a target between two consecutive frames with an affine image warping. Let  $\mathbf{s}_t = (x_t, y_t, \theta_t, s_t, \alpha_t, \phi_t)$ , where  $x_t, y_t, \theta_t, s_t, \alpha_t, \phi_t$  denote  $x, y$  translation, rotation angle, scale, aspect ratio, and skew direction at time  $t$  respectively. Each parameter in  $\mathbf{s}_t$  is modeled independently by a Gaussian distribution around its counterpart in  $\mathbf{s}_{t-1}$ . So  $p(\mathbf{s}_t | \mathbf{s}_{t-1})$  takes the form of:

$$p(\mathbf{s}_t | \mathbf{s}_{t-1}) = \mathcal{N}(\mathbf{s}_t; \mathbf{s}_{t-1}, \Sigma), \quad (19)$$

where  $\Sigma$  is a diagonal covariance matrix whose elements are the corresponding variances of the affine-transformation parameters, i.e.,  $\sigma_x^2, \sigma_y^2, \sigma_\theta^2, \sigma_s^2, \sigma_\alpha^2, \sigma_\phi^2$ .

#### 5.2. Observation model

In this inference framework, the observation model is very important because it reflects the unpredictable variations such as the target appearance or background changes. Using the similarity measure defined in Eq. (14), the observation model  $p(\mathbf{x} | \mathbf{s})$  (we omit  $t$  without



Fig. 4. Tracking results of the *David* sequence.

causing confusion) which denotes the likelihood of the learned appearance model generating  $\mathbf{x}$  can be defined as:

$$p(\mathbf{x}|\mathbf{s}) \propto \exp(-RE). \quad (20)$$

We can see that this observation model is dependent on the learned appearance model and the sparse representation model developed above. When our appearance model is updated, the observation model  $p(\mathbf{x}|\mathbf{s})$  is updated simultaneously. Algorithm 2 gives a summary of the complete tracking algorithm.

## 6. Experiments

For performance evaluation, we run the proposed algorithm on several challenging video sequences, most of which are publicly available. The challenging factors include large pose variation, significant lighting condition variation, occlusion, large scale change, scene blur, unknown camera movement, and so on.

### 6.1. Implementation

For  $\ell_1$  minimization, we choose the efficient sparse coding algorithm proposed in [21]. Without loss of generality, we first transform all sequences to gray scale (the tracking result is illustrated in the original RGB format). The target is manually selected in the first frame. The target region used in our experiments is normalized to  $32 \times 32$  patch. The number of subspace models are empirically set to be  $K=5$ . The subspace models are updated in each frame, and the threshold  $r$  is set to be 0.9. The forgetting factor is empirically set to be 0.95. With particle filtering, we have to make a trade-off between effectiveness and efficiency. Since this paper focuses on designing robust object representation and corresponding observation model which play the key role in effectiveness, we fix the number of particles to 300 and tune the parameter  $\Sigma$  on each test sequence for efficiency consideration. Implemented using MATLAB, our algorithm runs at about 6.5 frames per minute on a standard Core Duo 2.0 GHz computer.

**Algorithm 2.** Summary of the complete tracking algorithm.

**Input:** Video frames  $F_1, F_2, \dots, F_t$ .

**Output:** (1) Target states  $\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_t$  in frames  $F_1, F_2, \dots, F_t$  respectively;  
(2) Target regions associated with target states.

**for**  $t = 1 : \text{FrameNumber}$  **do**

**if**  $t = 1$  **then**

Initialization. Select the target in the first frame manually or using an automatic detector. Initialize the appearance model using the target observation in the first frame.

**else**

1. In frame  $F_t$ , draw particles  $\{\mathbf{s}_t^i, i = 1, \dots, N_s\}$  according to the dynamic model  $p(\mathbf{s}_t|\mathbf{s}_{t-1})$ ;
2. For each particle  $\mathbf{s}_t^i$ , calculate the likelihood  $p(\mathbf{x}_t|\mathbf{s}_t^i)$ ;
3. Estimate the target state  $\mathbf{s}_t$  using the MAP estimation method and store the target observation  $\mathbf{x}_t$  simultaneously. Draw the target region in the current frame;
4. Update the appearance model with the new target observation  $\mathbf{x}_t$ .

### 6.2. Experimental results

We use the *David* sequence [8] and the *Dudek* sequence [7] to test the performance of our tracking method in handling illumination change, pose change, occlusion and camera movement. In the *David* sequence, there are significant illumination change, pose change, camera movement and partial occlusion when the target walks around. Tracking results on sample frames are shown in Fig. 4. From the tracking results, we can see that our method succeeds in tracking the target. In the *Dudek* sequence, there are pose change, illumination change, expression variation, partial occlusion and camera movement. From the tracking result shown in Fig. 5, we can find that our tracking algorithm performs well throughout this sequence. Our method learns multiple subspace models to account for the appearance variation of the target object, and makes use of the sparse representation framework to deal with observation noise caused by partial occlusion or image corruption, thereby generating good tracking performance.

### 6.3. Comparison with other tracking methods

For comparison, we test two state-of-the-art tracking algorithms including the *incremental subspace learning* tracker (referred to as IVT here) [8], and the *sparse representation* tracker (referred to as L1T here) [17]. For fair comparison, we use the same  $\ell_1$  minimization method in L1T as ours. All these test trackers use the same feature vector (intensity values inside the target region with the same resolution), the same dynamic model (the same  $\Sigma$  for each sequence) and the same number of particles to estimate the tracking result. IVT uses an online linear subspace learning method for appearance modeling, and L1T uses sparse approximation for object representation. In these three methods, the only difference is the observation model and all the other components are exactly the same. This allows us to isolate the object modeling or representation component to make sure that it is the cause of the performance difference. We present some representative frames to show the performances of the proposed tracker and the other two trackers.

#### 6.3.1. Qualitative comparison

The *Sylvester* sequence is used to test the performance of these three tracking methods when the target object undergoes drastic pose change and scale change. The bad and changing lighting condition also make the target hard to be distinguished from the background. Tracking results on samples frames are shown in Fig. 6. IVT gets lost in tracking the target and goes out of range gradually. L1T performs better than IVT, but has scale errors on many frames when the target shows frequent pose changes. It is clear that our proposed tracker performs better than both L1T and IVT in this sequence. We also use this sequence to test the necessity of learning multiple subspace models for object representation. When we set the number of subspace models to be  $K=1$ , the tracking result is illustrated in the third row of Fig. 6, where the tracking result drifts away from the ground truth quickly.

We use the *singer* sequence [22] to evaluate whether these three tracking methods are able to handle drastic illumination changes. Some representative tracking results are shown in Fig. 7. In this



Fig. 5. Tracking results of the *Dudek* sequence.

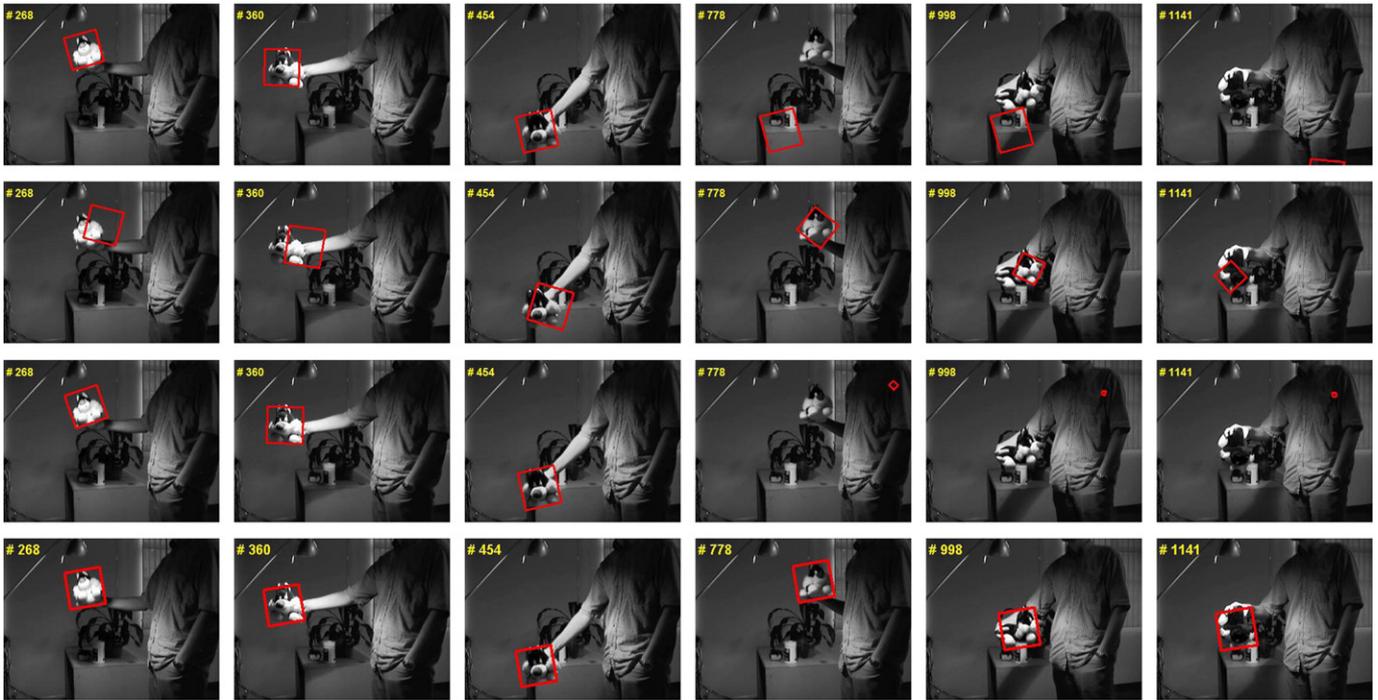


Fig. 6. Tracking results of the *Sylwester* sequence. Rows 1, 2, 3 and 4 are the results of IVT, L1T, our tracker when  $K=1$  and our tracker when  $K=5$ , respectively.

sequence, there are large scale changes of the target object and unknown camera movement besides the illumination changes. From the representative tracking result, we can find our tracker achieves good result. IVT fails after the drastic illumination change. L1T performs better than IVT, but also fails when the target undergoes significant scale change and camera movement.

In the *faceocc* sequence [15], the target undergoes partial occlusion and pose change. Tracking results on sample frames are shown in Fig. 8. We can find IVT and our tracking method perform well

throughout this sequence, while L1T fails when the target object is partially occluded by a hat.

The last test video sequence *mei* is obtained from [11]. In this sequence, a woman walks in a cluttered office environment. The challenges of this sequence include large pose variation, significant lighting variation, scale change and severe occlusion. The unknown camera movement also increases the tracking difficulty. Tracking results on sample frames are shown in Fig. 9. IVT drifts easily and loses the target after the lighting change. L1T performs better than

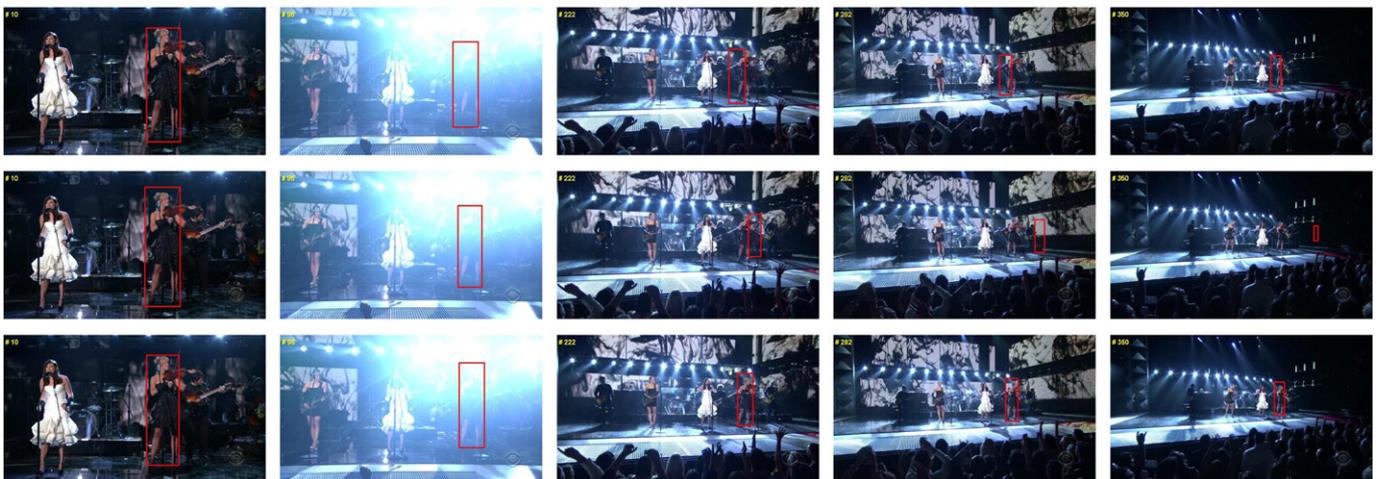


Fig. 7. Tracking results of the *singer* sequence. Rows 1, 2 and 3 are the results of IVT, L1T, and our tracker, respectively.

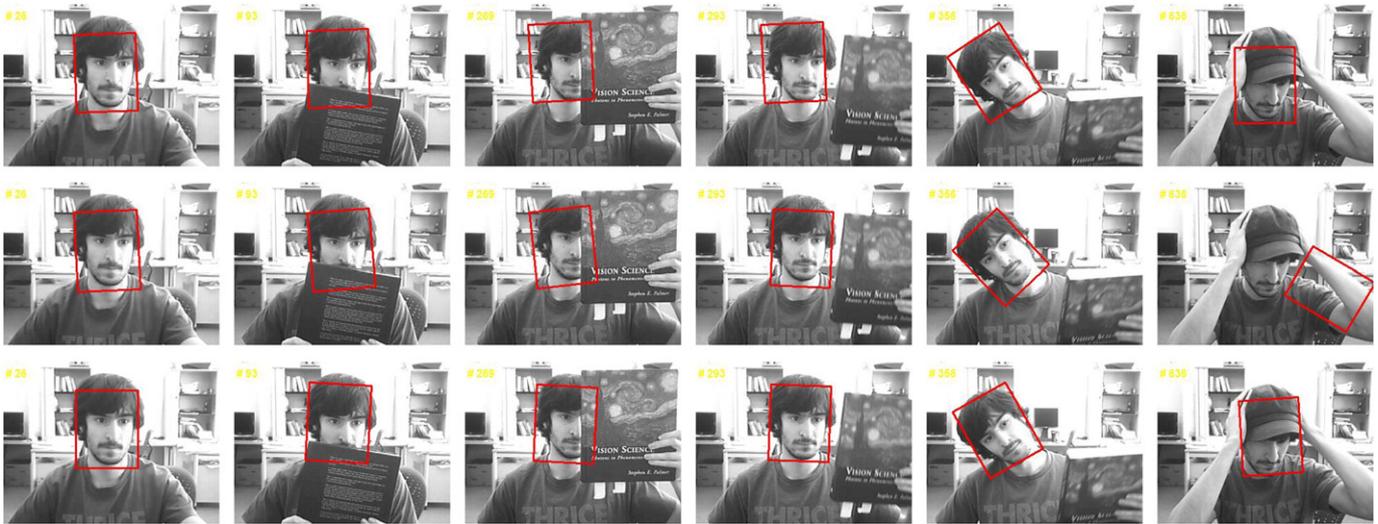


Fig. 8. Tracking results of *faceocc* sequence. Rows 1, 2 and 3 are the results of IVT, L1T, and our tracker, respectively.

IVT, but also gets lost in tracking the target due to occlusion and camera movement. From the tracking results, it is clear that the proposed tracking algorithm performs the best.

### 6.3.2. Quantitative comparison

We use two criteria to quantitatively evaluate the performance of the proposed tracker. The first one is Success Rate. To calculate the Success Rate, we first choose a score from the PASCAL challenge [23] to evaluate if tracking is successful in each frame: Given the tracked bounding box  $ROI_{TK}$  and the ground truth bounding box  $ROI_{GT}$ , the score is defined as

$$\text{score} = \frac{\text{area}(ROI_{TK} \cap ROI_{GT})}{\text{area}(ROI_{TK} \cup ROI_{GT})} \quad (21)$$

Tracking in each frame is considered to be successful when this score exceeds 0.5. Then the Success Rate is defined as the percentage of successfully tracked frames for each sequence. The success rates of our tracker and the other two trackers on the test video sequences are listed in Table 1. From it, we can see the proposed tracker performs well against the other trackers.

The second criterion we used for evaluating the tracking performance is the Center Location Error. The Center Location Error is approximated by the distance between the central position of the tracking result and that of the manually labeled ground truth. Fig. 10 shows the errors of all three trackers, and Table 2 shows the average center location errors. We can see that our tracking method has smaller tracking errors than the other two on these test video sequences.

From these comparison results, we can see that our proposed tracking algorithm outperforms IVT and L1T in the scenarios with occlusions, large pose variations, drastic illumination changes and scale changes. IVT use a single linear subspace model to represent the target appearance during tracking, which may not generate a good performance when the target appearance manifold is not linear at all. L1T uses sparse representation to address the occlusion problem, but the selection of bases (templates) for target representation is difficult. Only partial observations of the target are preserved in L1T's template set, and an improper update will influence the whole tracking performance. By contrast, we use subspace learning method to model the nonlinear appearance manifold of the target. Then we use the learned appearance model to represent the target in a sparse representation framework, which makes our tracking algorithm insensitive to occlusion. Finally, we use the reconstruction error to construct a likelihood

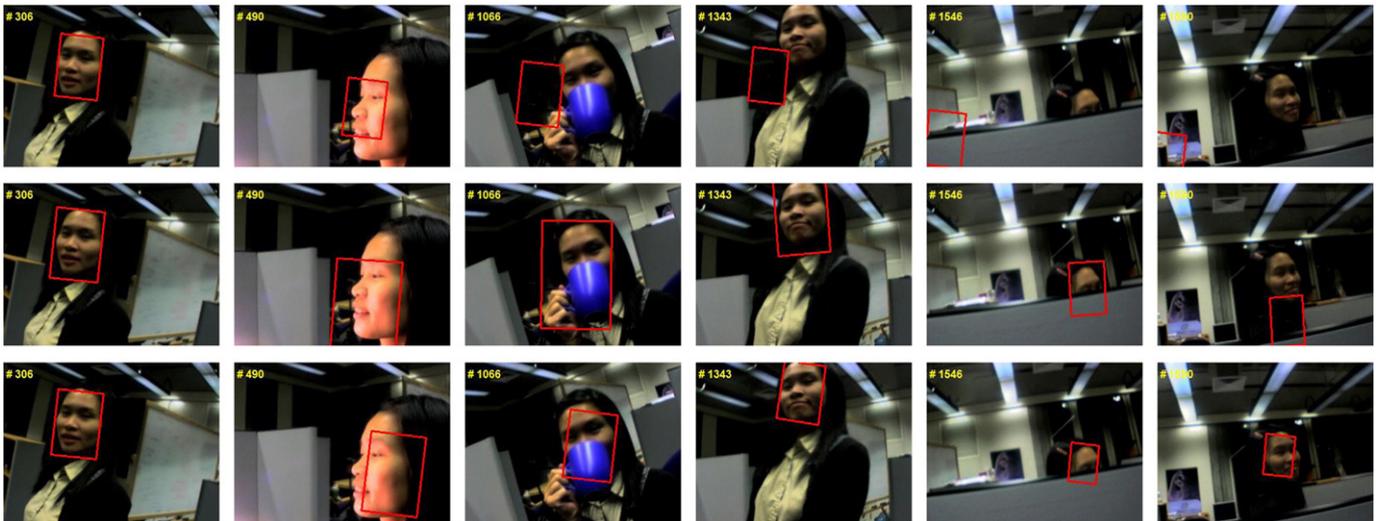


Fig. 9. Tracking results of *mei* sequence. Rows 1, 2 and 3 are the results of IVT, L1T, and our tracker, respectively.

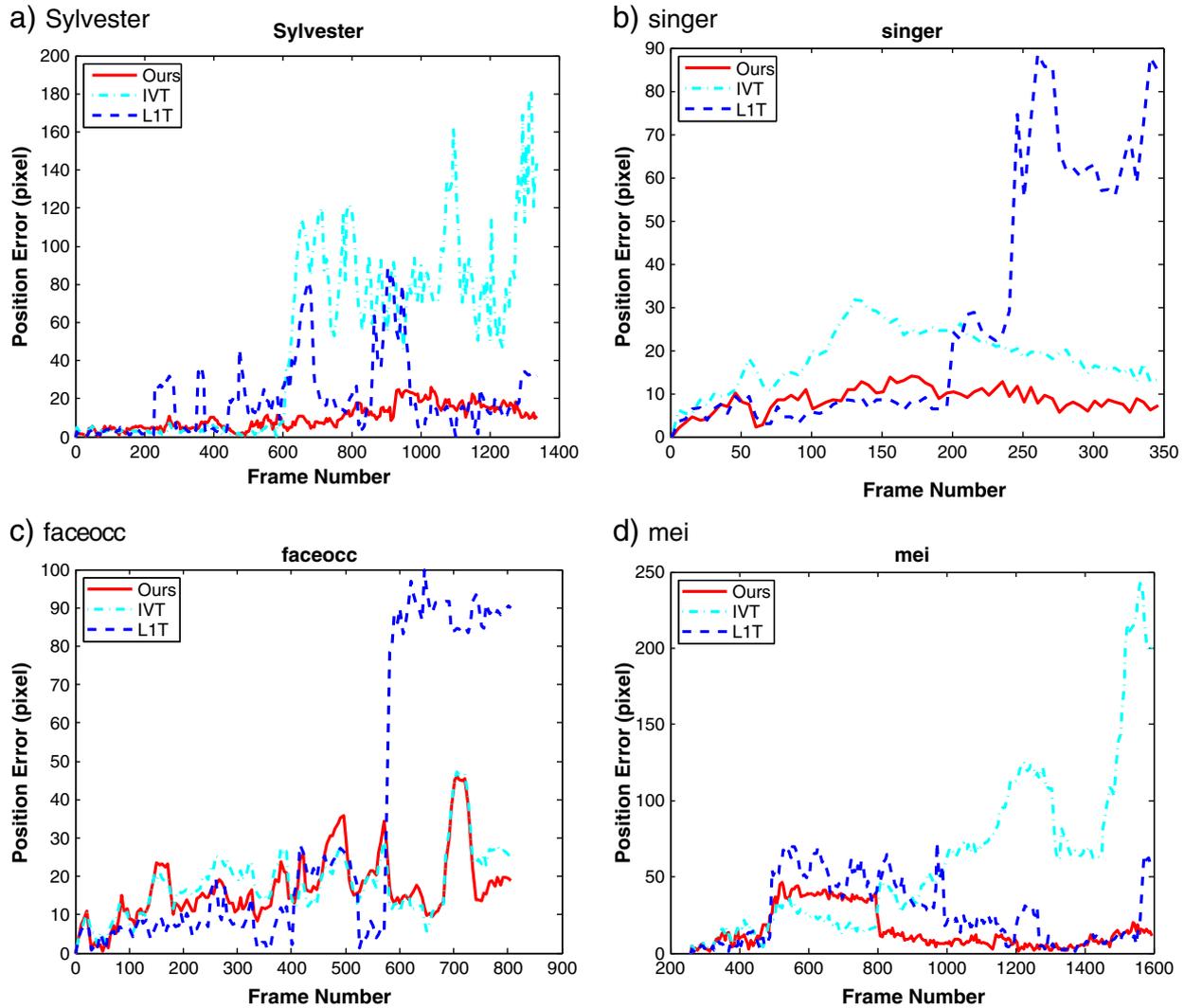


Fig. 10. Error plots of the tested sequences.

function for tracking. This makes our tracking algorithm performs better in these complex scenarios.

7. Conclusion

We have presented a tracking algorithm based on appearance modeling and sparse representation. Based on the image intensities,

Table 1 Tracking Success Rate.

	IVT	L1T	Our tracker
<i>Sylvester</i>	56.3%	74.7%	<b>96.8%</b>
<i>singer</i>	34.6%	60.0%	<b>100%</b>
<i>faceocc</i>	<b>100%</b>	72.0%	<b>100%</b>
<i>mei</i>	33.0%	55.1%	<b>78.2%</b>

The best result is printed in bold faced letters.

Table 2 The average tracking errors (in pixels).

	IVT	L1T	Our tracker
<i>Sylvester</i>	49.3	19.6	<b>10.0</b>
<i>singer</i>	18.4	27.5	<b>8.6</b>
<i>faceocc</i>	18.1	32.6	<b>16.9</b>
<i>mei</i>	60.3	26.6	<b>15.1</b>

The best result is printed in bold faced letters.

we proposed an effective online learning algorithm to model the target appearance variations during tracking. Different kinds of target appearances are modeled by different subspaces in our appearance model. Then we used the sparse representation framework to deal with possible abrupt appearance change in tracking. The reconstruction error with the bases of our learned appearance model is used to measure the likelihood of a test candidate. Compared to the state-of-the-art tracking methods, our tracking algorithm is more robust in complex environment when the target undergoes large pose variation, illumination variation, occlusion, scale change and unknown camera movement. Experiments have confirmed the effectiveness of the proposed tracking algorithm.

Acknowledgments

This work was supported by National Natural Science Foundation of China (Project no. 60772050 and Project no. 61071131), No.2 Important National Science and Technology Specific Projects (Project no. 2009ZX02001) and United Technologies Research Center (UTRC).

Appendix A. Supplementary data

Supplementary data to this article can be found online at doi:10.1016/j.imavis.2011.08.006.

## References

- [1] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. Huang, S. Yan, Sparse representation for computer vision and pattern recognition, *Proceedings of the IEEE* 98 (6) (2010) 1031–1044.
- [2] G.D. Hager, P.N. Belhumeur, Efficient region tracking with parametric models of geometry and illumination, *PAMI* 20 (10) (1998) 1025–1039.
- [3] D. Comaniciu, V. Ramesh, P. Meer, Kernel-based object tracking, *PAMI* 25 (5) (2003) 564–575.
- [4] A. Azarbayejani, A. Pentland, Recursive estimation of motion, structure, and focal length, *IEEE Trans. Pattern Anal. Mach. Intell.* 17 (6) (1995) 562–575.
- [5] M. Isard, A. Blake, Condensation— conditional density propagation for visual tracking, *IJCV* 29 (1) (1998) 5–28.
- [6] M. Black, A. Jepson, Eigenttracking: robust matching and tracking of articulated objects using a view-based representation, *ECCV*, 1996, pp. 329–342.
- [7] A.D. Jepson, D.J. Fleet, T.F. El-Maraghi, Robust online appearance models for visual tracking, *PAMI* 25 (10) (2003) 1296–1311.
- [8] D. Ross, J. Lim, R.-S. Lin, M.-H. Yang, Incremental learning for robust visual tracking, *IJCV* 77 (1–3) (2008) 125–141.
- [9] X. Li, W. Hu, Z. Zhang, X. Zhang, M. Zhu, J. Cheng, Visual tracking via incremental log-euclidean riemannian subspace learning, *Proc. IEEE Conference on Computer Vision and Pattern Recognition CVPR 2008*, 2008, pp. 1–8, doi:10.1109/CVPR.2008.4587516.
- [10] E. Candes, X. Li, Y. Ma, J. Wright, Robust principal component analysis, preprint.
- [11] J. Ho, K.-C. Lee, M.-H. Yang, D. Kriegman, Visual tracking using learned linear subspaces, *CVPR*, 2004, pp. 782–789.
- [12] R.T. Collins, Y. Liu, M. Leordeanu, Online selection of discriminative tracking features, *PAMI* 27 (10) (2005) 1631–1643.
- [13] S. Avidan, Ensemble tracking, *PAMI* 29 (2) (2007) 261–271.
- [14] T. Parag, F. Porikli, A. Elgammal, Boosting adaptive linear weak classifiers for online learning and tracking, *CVPR*, 2008, pp. 1–8, doi:10.1109/CVPR.2008.4587556.
- [15] B. Babenko, M.-H. Yang, S. Belongie, Visual tracking with online multiple instance learning, *CVPR*, 2009, pp. 983–990.
- [16] J. Wright, A. Yang, A. Ganesh, S. Sastry, Y. Ma, Robust face recognition via sparse representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2008) 210–227.
- [17] X. Mei, H. Ling, Robust visual tracking using  $l_1$  minimization, *ICCV*, 2009, pp. 1436–1443.
- [18] E. Candes, T. Tao, Near-optimal signal recovery from random projections: universal encoding strategies? *IEEE transactions on information theory* 52 (12) (2006) 5406–5425.
- [19] M. Brand, Incremental singular value decomposition of uncertain data with missing values, *proc. ECCV'02*, 2002, pp. 707–720.
- [20] A. Levy, M. Lindenbaum, Sequential Karhunen–Loeve basis extraction and its application to images, *IEEE Trans. on Image Processing* 9 (8) (2000) 1371–1374.
- [21] H. Lee, A. Battle, R. Raina, A. Ng, Efficient sparse coding algorithms, *NIPS*, 2007.
- [22] J. Kwon, K. Lee, Visual tracking decomposition, *CVPR*, 2010, pp. 1269–1276.
- [23] M. Everingham, L. Van Gool, C. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, *International journal of computer vision* 88 (2) (2010) 303–338.