Tracking Nonstationary Visual Appearances by Data-Driven Adaptation

Ming Yang, *Member, IEEE*, Zhimin Fan, Jialue Fan, *Student Member, IEEE*, and Ying Wu, *Senior Member, IEEE*

***Abstract*—Without any prior about the target, the appearance is usually the only cue available in visual tracking. However, in general, the appearances are often nonstationary which may ruin the predefined visual measurements and often lead to tracking failure in practice. Thus, a natural solution is to adapt the observation model to the nonstationary appearances. However, this idea is threatened by the risk of adaptation drift that originates in its ill-posed nature, unless good data-driven constraints are imposed. Different from most existing adaptation schemes, we enforce three novel constraints for the optimal adaptation: 1) negative data, 2) bottom-up pair-wise data constraints, and 3) adaptation dynamics. Substantializing the general adaptation problem as a subspace adaptation problem, this paper presents a closed-form solution as well as a practical iterative algorithm for subspace tracking. Extensive experiments have demonstrated that the proposed approach can largely alleviate adaptation drift and achieve better tracking results for a large variety of nonstationary scenes.**

***Index Terms*—Appearance model adaptation, subspace tracking, visual tracking.**

I. Introduction

VISUAL tracking establishes the correspondences of the target of interest between successive frames, which is a fundamental research problem in video analysis and is important for a large variety of applications including video surveillance and human-computer interaction. In recent years, video-based tracking experienced a steady advance in both theory and practice, e.g., the sampling-based methods [1]–[3] and the kernel-based methods [4]–[7]. Although many tracking tasks can be successfully handled by using these techniques, real situations in practice, such as long duration tracking in unconstrained environments, still pose enormous challenges to these techniques. One common challenge arisen from these real situations is the nonstationary changes of the visual appearances of the target due to the view changes, illumination variations [8] and shape deformation [9]. Such appearance changes can ruin the prespecified visual measurement (or observation) model and lead to tracking failure.

Generally speaking, two approaches can be taken to deal with this challenge. One is to exploit the visual invariants [10], [11] of the targets. However, in general, finding invariants itself is very difficult, if not impossible, although learning methods can be employed [12]–[16]. Another approach is to adapt the tracker to the changes, for example, by updating the appearance models [17]–[19], [2], [20], or selecting the best visual features [21]–[25]. Unlike the invariants-based methods which require off-line learning of the visual measurement models (the appearance models) and sometimes are tantamount to detection and recognition problems, the adaptation-based methods tend to be more flexible, since the measurement models are adaptive or the features used for tracking can be adaptively selected [26]–[28].

In most existing adaptation-based methods, it is not uncommon to observe adaptation drift, i.e., the appearance models adapt to other image regions rather than the target of interest and lead to tracking failure. Many *ad hoc* remedies have been proposed to alleviate the drift, e.g., by enforcing the similarity to the initial model [18], [21] which confines the range of possible adaptations. However, in general, such a phenomenon of adaptation drift is not accidental but widely exists in a large variety of adaptation schemes, and poses a threat to adaptation-based visual trackers. Thus, it justifies an in-depth investigation to facilitate careful designs of the adaptation schemes.

In most existing adaptive tracking methods, the model at the current time instant is updated by the new data that are closest to the model at previous time step, with a hidden assumption that the best model (or feature) up to time $t-1$ is also the best for time t. Unfortunately, this assumption may not universally hold. As a result, when this assumption becomes invalid, the data closest to the model at time $t-1$ may actually be far from the right model at time t, and thus deviating the adaptation and failing the tracker.

The nature of the adaptive tracking problem lies in a chicken-and-egg dilemma [29]: the right data at time t are found by the right model at time t, while the right model can only be adapted by using the right data at time t. If no constraints are enforced, any new data can lead to a valid and stable adaptation, since the adapted model tends to best fit the new data. Therefore, in order to make this problem well-posed, we need to introduce good data-driven constraints from the image observations at the current time instant, and they should be reasonable and allow a wide range of adaptation.

In this paper, we substantialize the general adaptation problem as a subspace adaptation problem in nonstationary appearance tracking, where the target visual appearances for

Manuscript received June 08, 2008; revised March 03, 2009. First published May 26, 2009; current version published June 12, 2009. This work was supported in part by the National Science Foundation Grant IIS-0347877 and in part by the U.S. Army Research Laboratory and the U.S. Army Research Office under Grant ARO W911NF-08-1-0504. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Xuelong Li.

M. Yang is with NEC laboratories America, Inc., Cupertino, CA 95014 USA (e-mail: m-yang4@u.northwestern.edu).

Z. Fan, J. Fan, and Y. Wu are with the Electrical Engineering and Computer Science Department, Northwestern University, Evanston, IL 60208-3118 USA (e-mail: zfa825@ece.northwestern.edu; jfa699@ece.northwestern.edu; yingwu@ece.northwestern.edu).

Digital Object Identifier 10.1109/TIP.2009.2019807

a short time interval are represented as a linear subspace. We analyze the ill-posed adaptive tracking problem in this setting. In our approach, we enforce three novel constraints for the optimal adaptation: 1) negative data that are easily available, 2) pair-wise data constraints that are used to identify positive data from bottom-up, and 3) adaptation dynamics that smooth the updating process. Then, we give a closed-form solution to this subspace tracking problem and also provide a practical iterative algorithm. Our method not only estimates the motion parameters of the target but also keeps track of the appearance subspaces. Note our model adaptation strategy is fundamentally different from the methods using on-line learning [23], [25], where the appearance models at previous time frames are directly utilized to supervise labelling the current image observations, and to determine the new information for updating the model. In contrast, we mainly resort to the data-driven constraints to select positive and negative data from current image observations for the appearance model adaptation.

In the next section, we briefly review related tracking algorithms in terms of different observation models, afterwards the dilemma in the traditional adaptation schemes is investigated. Our solution to the dilemma of the adaptation is elaborated in Section III. In Section IV, we present the experiments results and discussions. The concluding remarks are given in Section V.

II. Related Work and Motivation

In tracking, the target is tracked or detected based on the matching between the observed visual evidence (or measurements) and the visual model. We generally call the model that stipulates the matching as the *observation model* or the *measurement model*. Without any prior of the targets, appearance-based tracking approaches are more general than feature-based methods. The visual appearances of an object may bear a manifold in the high-dimensional image space. Depending on the features used to describe the target and on the variances of the appearances, such a manifold can be quite nonlinear and complex. Therefore, the complexity in the appearances largely determines the degrees of difficulty of the tracking task.

A. Visual Observation Models in Tracking

We can roughly categorize the observation models in various tracking algorithms into three classes: 1) with fixed appearance templates, 2) with known appearance manifolds, 3) with adaptive appearance manifolds on-the-fly.

In the observation models with fixed appearance templates, the motion parameters to be estimated (denoted by $\mathbf{x}$) are the only variables that affect the appearance changes. We denote the image observations as $\mathbf{z}$ and the hypothesized one as $\hat{\mathbf{z}}(\mathbf{x})$. Then the observation model needs to measure the similarity of $\mathbf{z}$ and $\hat{\mathbf{z}}(\mathbf{x})$, or the likelihood $p(\mathbf{z}|\mathbf{x}) = p(\mathbf{z}|\hat{\mathbf{z}}(\mathbf{x}))$, assuming the motion parameter $\mathbf{x}$ deterministically specifies the corresponding hypothesis $\hat{\mathbf{z}}(\mathbf{x})$. If $\mathbf{z}$ is a vector, i.e., $\mathbf{z} \in \mathbb{R}^m$, this class of observation models is concerned with the distance between two vectors. The image observations $\mathbf{z}$ can be edges [1], color histograms [4], feature points [30], etc. Most tracking algorithms employ this type of observation model.

The motion parameters of interest may not be the only contribution to appearance changes, but there can be many other factors. We denote the hypothesized observation by $\hat{\mathbf{z}}(\mathbf{x}, \theta)$ to indicate the influences of other factors θ besides the target motion. For example, the illumination also affects the appearance [14] (e.g., in tracking a face), or the nonrigidity of the target changes the appearances (e.g., in tracking a pedestrian), but we may not be interested in recovering too many delicate nonrigid motion parameters. Thus, there are uncertainties in the appearances model itself, and the observation model needs to integrate all uncertainties, i.e.,

$$\begin{aligned} p(\mathbf{z}|\mathbf{x}) &= \int_\theta p(\mathbf{z}|\mathbf{x}, \theta) p(\theta|\mathbf{x}) d\theta \\ &= \int_\theta p(\mathbf{z}|\hat{\mathbf{z}}(\mathbf{x}, \theta)) p(\theta|\mathbf{x}) d\theta. \end{aligned} \tag{1}$$

In other words, given a motion hypothesis $\mathbf{x}$, its hypothesized observation $\hat{\mathbf{z}}(\mathbf{x})$ is no longer a vector, but a manifold in $\mathbb{R}^m$, and the observation model needs to calculate the distance of the evidence $\mathbf{z}$ to this manifold. Depending on the free parameters θ, such a manifold can be as simple as a linear subspace [13], [14], or as complex as a highly nonlinear one [12], [15]. The second class of observation models assumes a known manifold, which can be learned from training data off-line in advance.

Although the appearance manifolds exist, in most cases, they are quite complex, and the learning task itself is challenging enough. In addition, in real applications, we may not have the luxury of being able to learn the manifolds of arbitrary objects for two reasons: we may not be able to collect enough training data, and the applications may not allow the off-line processing. Thus, we need to recover and update the appearance manifolds online [2], [17], [19], [20], or the densities of certain appearance features [31] during the tracking. In general, we make a reasonable assumption that the manifold during a short time interval is linear [18], [32]. The nonlinear manifold is approximated by piece-wise linear subspace [33] or mapped to low-dimensional manifold using nonlinear mapping [34], or the learned general subspace could be updated to a specific one during the tracking [35]. The method of online feature selection, e.g., in [21], can also be categorized in this class, since the selected features span a subspace. In these methods, model drift is one of the common and fundamental challenges.

This paper studies the problem of adaptive appearance models. The differences from the existing work include an in-depth analysis of the adaptation drift, and a novel solution that alleviates the drift by enforcing both bottom-up data-driven and top-down constraints.

B. Chicken-and-Egg Dilemma

Although the appearance manifold of a target can be quite complex and nonlinear, it is reasonable to assume the linearity over a short time interval. In this paper, we assume the appearances (or visual features) $\mathbf{z} \in \mathbb{R}^m$ lie in a linear subspace $\mathcal{L}$ spanned by r linearly independent columns of a linear transform $\mathbf{A} \in \mathbb{R}^{m \times r}$, i.e., $\mathbf{z}$ is a linear combination of the columns of $\mathbf{A}$. We write $\mathbf{z} = \mathbf{A}\mathbf{y}$.

The projection of $\mathbf{z}$ to the subspace $\mathbb{R}^r$ is given by the least square solution of $\mathbf{z} = \mathbf{A}\mathbf{y}$, i.e.,

$$\mathbf{y} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{z} = \mathbf{A}^\dagger\mathbf{z} \tag{2}$$

where $\mathbf{A}^{\dagger} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$ is the pseudo-inverse of $\mathbf{A}$. The reconstruction of the projection in $\mathbb{R}^m$ is given by

$$\bar{\mathbf{z}} = \mathbf{A}\mathbf{A}^{\dagger}\mathbf{z} = \mathbf{P}\mathbf{z} \tag{3}$$

where $\mathbf{P} = \mathbf{A}\mathbf{A}^{\dagger} \in \mathbb{R}^{m\times m}$ is called the *projection matrix*. Unlike the orthonormal basis, the projection matrix is unique for a subspace. We can decompose the Hilbert space $\mathbb{R}^m$ into two orthogonal subspaces: a r-dimensional subspace characterized by $\mathbf{P}$ and its $(m-r)$-dimensional orthogonal complement characterized by $\mathbf{P}^{\perp} = \mathbf{I} - \mathbf{P}$.

Therefore, the subspace $\mathcal{L}$ delineated by a random vector process $\{\mathbf{z}\}$ is given by the following optimization problem:

$$\begin{aligned}\mathbf{P}^* &= \arg\min_P \mathrm{E}(\|\mathbf{z} - \mathbf{P}\mathbf{z}\|^2)\\ &= \arg\min_P \mathrm{E}(\|\mathbf{P}^{\perp}\mathbf{z}\|^2)\end{aligned} \tag{4}$$

where $\mathrm{E}(\cdot)$ is the expectation over the random vector process $\{\mathbf{z}\}$. It is easy to prove that the optimal subspace is spanned by the r principal components of the data covariance matrix. This problem is well-posed since the samples from $\{\mathbf{z}\}$ are given, thus the covariance matrix is known.

However, in the tracking scenario, the problem becomes

$$\{\mathbf{P}_t^*, \mathbf{x}_t^*\} = \arg\min_{P_t, x_t} \mathrm{E}\left(\left\|\mathbf{P}_t^{\perp}\mathbf{z}(\mathbf{x}_t)\right\|^2\right) \tag{5}$$

where $\mathbf{x}_t$ is the motion parameters to be tracked. In this setting, we are facing a dilemma: if $\{\mathbf{x}\}$ cannot be determined, then neither can $\mathbf{P}$, and vice versa. Namely, given any tracking result, good or bad, we can always find an optimal subspace that can best explain this particular result.

Unlike some other chicken-and-egg problems, this problem is even worse since no constraints on either $\mathbf{P}$ or $\{\mathbf{x}\}$ are imposed. For arbitrary tracking result $\{\mathbf{x}\}$, there exists a projection matrix $\mathbf{P}$ to minimize (5), and vice versa. Therefore, this problem is ill-posed and the formulation allows arbitrary subspace adaptations.

From the analysis above, it is clear that constraints need to be added to make this problem well-posed. A commonly used constraint is the "smoothness" of the adaptation, i.e., the updated model should not deviate much from the previous one, and most existing methods [17], [18], [21], [32] solve this dilemma in the following manner:

$$\begin{cases}\mathbf{x}_t^* = \arg\min_{x_t} \mathrm{E}\left(\left\|\mathbf{P}_{t-1}^{\perp}\mathbf{z}(\mathbf{x}_t)\right\|^2\right)\\ \mathbf{P}_t^* = \arg\min_{P_t} \mathrm{E}\left(\left\|\mathbf{P}_t^{\perp}\mathbf{z}(\mathbf{x}_t^*)\right\|^2\right).\end{cases} \tag{6}$$

In this adaptation scheme, at time t, the data that are the closest to the subspace at the previous time instant are found first, then they are used to update the subspace. This approach is valid only if the following assumption holds: the optimal subspace at $t-1$ is also optimal for time t. In reality, this assumption may not necessarily be true, since a data point that is the closest to the subspace $\mathcal{L}_t$ may not be the closest to $\mathcal{L}_{t-1}$. Thus, we often observe that the model adaptation cannot keep up with the real changes and the model gradually drifts away. When the data found based on $\mathbf{P}_{t-1}$ in fact deviate from $\mathbf{P}_t$ significantly, the adaptation is catastrophic. Although this approach makes the original ill-posed problem in (5) well-posed, it is prone to drift and thus not robust.

III. Our Solution

From the analysis in Section II-B, it is clear that we need more constraints than the adaptation dynamics constraint alone. In the tracking problem, at time t before the target is detected, all the observation data are unlabelled data, i.e., we cannot tell whether or not a certain observation should be associated (or classified) to the target appearance subspace. The adaptation dynamics constraint is a top-down constraint, which does not provide much supervised information to the data at time t. Therefore, to make the adaptation more robust, we need to also identify and employ bottom-up data-driven constraints, besides the smoothness constraint.

In this paper, we propose to integrate the following three constraints.

- *Negative data constraints*. At the current time t, although it is difficult to obtain the positive data (i.e., the visual observations that are truly produced by the target), negative data are everywhere. In fact, positive data are very rare in all the set of possible observation data. The negative data may help to constrain the target appearance subspaces. We denote the positive data at time t by $\mathbf{z}_t^+$, and the negative data by $\mathbf{z}_t^-$.
- *Pair-wise data constraints*. Given a pair of data points, it is relatively easier to determine whether or not they belong to the same class. Such pair-wise data constraints are also widely available. A large number of pair-wise constraints may lead to a rough clustering of the data. Based on the smoothness constraints, we can determine a set of *possible positive* data to constrain the subspace updating. The detail is in Section III-E.
- *Adaptation smoothness constraints*. The smoothness constraints are essential for the tracking process, since the process of the data at time t should take advantage of the subspace at time $t-1$. There are many ways to represent and use this type of constraints. The most common scheme as indicated in Section II-B enforces a very strong smoothness constraint. In our approach, we treat the constraint as a penalty which can be balanced with other types of constraints. The penalty is proportional to the distance of two subspaces, i.e., the Frobenius norm of the difference of the two projection matrices $\|\mathbf{P}_t - \mathbf{P}_{t-1}\|_F^2$.

A. Formulation

When processing the current frame t, the following are assumed to be known: 1) the projection matrix of the previous appearance subspace $\mathbf{P}_{t-1}$, 2) a set of negative data collected from the current image frame, $\{\mathbf{z}_t^-\}$, 3) a set of possible positive data identified based on the pair-wise constraints, $\{\mathbf{z}_t^+\}$, 4) previous negative covariance matrix $\mathbf{C}_{t-1}^-$ and positive covariance matrix $\mathbf{C}_{t-1}^+$.

An optimal subspace should have the following properties: The positive data should be close to their projections; the negative data should be far from their projections onto this subspace; and this subspace should be close to the previous one. Therefore, we form an optimization problem to solve for the optimal subspace at current time t

$$\begin{aligned}\min_{A_t} J_0(\mathbf{A}_t) = \min_{A_t} \big\{ & \mathtt{E}(\|\mathbf{z}_t^+ - \mathbf{P}_t\mathbf{z}_t^+\|^2) \\ & +\mathtt{E}(\|\mathbf{P}_t\mathbf{z}_t^-\|^2) + \alpha\,\|\mathbf{P}_t - \mathbf{P}_{t-1}\|_F^2 \big\} \qquad (7)\end{aligned}$$

where $\mathbf{P}_t = \mathbf{A}_t\mathbf{A}_t^\dagger$ is the projection matrix and $\alpha > 0$ is a weighting factor. The three terms on the right-hand side of (7) correspond to the aforementioned properties. The optimal subspace strives to ensure $\mathbf{z}_t^+$ close to their projections $\mathbf{P}_t\mathbf{z}_t^+$, the norms of negative data's projections $\mathbf{P}_t\mathbf{z}_t^-$ are small, and the projection matrices $\mathbf{P}_t$ and $\mathbf{P}_{t-1}$ in consecutive frames are close. We denote by $\mathbf{C}_t^+ = \mathtt{E}(\mathbf{z}_t^+\mathbf{z}_t^{+T})$, and $\mathbf{C}_t^- = \mathtt{E}(\mathbf{z}_t^-\mathbf{z}_t^{-T})$. It is easy to show (7) is equivalent to the following:

$$\begin{aligned}\min_{A_t} J_1(\mathbf{A}_t) = \min_{A_t} \big\{ & \mathtt{tr}\left(\mathbf{P}_t\mathbf{C}_t^-\right) \\ & -\mathtt{tr}\left(\mathbf{P}_t\mathbf{C}_t^+\right) + \alpha\|\mathbf{P}_t - \mathbf{P}_{t-1}\|_F^2 \big\} \qquad (8)\end{aligned}$$

where $\mathtt{tr}(\cdot)$ denotes the trace of a matrix. In the regularization term, α is set to 0.2 in all our experiments.

B. Closed-Form Solution

Theorem 1: The solution to the problem in (8) is given by $\mathbf{P}_t = \mathbf{U}\mathbf{U}^T$, where $\mathbf{U}$ is constituted by the r eigenvectors that corresponds to the r smallest eigenvalues of a symmetric matrix

$$\hat{\mathbf{C}} = \mathbf{C}_t^- - \mathbf{C}_t^+ + \alpha\mathbf{I} - \alpha\mathbf{P}_{t-1}. \qquad (9)$$

The proof of this theorem is given in the Appendix. Please note that the solution to $\mathbf{A}_t$ is not unique, but the projection matrix $\mathbf{P}_t$ is. If we require that $\mathbf{A}_t$ is spanned by r orthogonal vectors, then $\mathbf{A}_t = \mathbf{U}$. Please also note the eigenvalues of $\hat{\mathbf{C}}$ may be negative.

By considering the data in previous time instants, we can use a forgetting factor $\beta < 1$, which can down-weight the influence of the data from previous times. This is equivalent to the use of an exponentially-weighted sliding window over time to calculate the covariance matrices. Thus, we can write

$$\begin{aligned}\mathbf{C}_t &= \sum_{k=1}^{t} \beta^{t-k}\mathtt{E}\left(\mathbf{z}_k\mathbf{z}_k^T\right) \\ &= \beta\mathbf{C}_{t-1} + \mathtt{E}\left(\mathbf{z}_t\mathbf{z}_t^T\right). \qquad (10)\end{aligned}$$

This way, we can update both $\mathbf{C}_t^+$ and $\mathbf{C}_t^-$.

C. Iterative Algorithm

Section III-B gives a closed-form solution to the subspace, but this solution involves the eigenvalue decomposition of a $m \times m$ matrix $\hat{\mathbf{C}}$, where m is the dimension of the visual observation vectors and thus can be quite large. To achieve a less demanding computation, we develop an iterative algorithm in this section, by formulating another optimization problem according to (7) as

$$\begin{aligned}\min_{U} J_2(\mathbf{U}) = \min_{U} \big\{ & \mathtt{E}(\|\mathbf{z}_t^+ - \mathbf{U}\mathbf{U}^T\mathbf{z}_t^+\|^2) \\ & +\mathtt{E}(\|\mathbf{U}\mathbf{U}^T\mathbf{z}_t^-\|^2) + \alpha\|\mathbf{U}\mathbf{U}^T - \mathbf{P}_{t-1}\|_F^2 \big\} \\ \text{s.t. } \mathbf{U}^T\mathbf{U} = \mathbf{I} \qquad & (11)\end{aligned}$$

where $\mathbf{U} \in \mathbb{R}^{m \times r}$ is constituted by r orthonormal columns. The gradient of J_2 is given by

$$\begin{aligned}\nabla J_2(\mathbf{U}) &= \frac{\partial J_2(\mathbf{U})}{\partial \mathbf{U}} \\ &\propto \left(\mathbf{C}_t^- - \mathbf{C}_t^+ + \alpha\mathbf{I} - \alpha\mathbf{P}_{t-1}\right)\mathbf{U}. \qquad (12)\end{aligned}$$

To find the optimal solution of $\mathbf{U}$, we can use the gradient descent iterations

$$\mathbf{U}^k \longleftarrow \mathbf{U}^{k-1} - \mu\nabla J_2(\mathbf{U}^{k-1}) \qquad (13)$$

during which the columns of $\mathbf{U}^k$ need to be orthogonalized after each update.

To speed up the iteration, we perform an approximation. When the subspace is to be updated by the positive data $\mathbf{z}_t^+$, the PAST algorithm [36] is applied for fast updating, which will be introduced in the next sub-section. When the updating is directed by the negative data $\mathbf{z}_t^-$, we can use the gradient decent method in (12).

D. Incrementally Updating by Past

In our approach we employ the Projection Approximation Subspace Tracking (PAST) algorithm [36] to incrementally update the subspace when new positive observations are arriving. To make this paper self-contained, we briefly introduce the basic idea of the PAST here. In PAST, the estimation of the subspace $\mathcal{L}$ delineated by a random vector process $\mathbf{z} \in \mathbb{R}^m$ is formulated as a scalar function optimization problem

$$\begin{aligned}J(\mathbf{A}) &= E(\|\mathbf{z} - \mathbf{A}\mathbf{A}^T\mathbf{z}\|^2) \\ &= \mathrm{tr}(\mathbf{C}) - 2\mathrm{tr}(\mathbf{A}^T\mathbf{C}\mathbf{A}) \\ &\quad + \mathrm{tr}(\mathbf{A}^T\mathbf{C}\mathbf{A} \cdot \mathbf{A}^T\mathbf{A}) \qquad (14)\end{aligned}$$

with a matrix argument $\mathbf{A} \in \mathbb{R}^{m \times r}$. Here we abuse the notation $\mathbf{A}$ a little bit, then we can see this matrix argument is the same as the linear transform $\mathbf{A}_t$ in the aforementioned formulation. We can see (14) is essentially as the same as the first term of (11) if we update the subspace only by the incoming positive data $\mathbf{z}_t^+$.

Yang [36] proved that the global minimum of $J(\mathbf{A})$ is achieved when $\mathbf{A}$ spans the r-dimensional subspace $\mathcal{L}$. Therefore, updating the projection matrix $\mathbf{P}$ of $\mathcal{L}$ translates to solve the optimization problem and calculate $\mathbf{P} = \mathbf{A}\mathbf{A}^\dagger$. Note that there is no orthonormal constraint on $\mathbf{A}$ and the eigenvectors of $\mathcal{L}$ are not necessarily contained in $\mathbf{A}$.

Replacing the expectation in (14) with the exponentially weighted sum of t steps, the optimization problem becomes

$$J(\mathbf{A}_t) = \sum_{i=1}^{t} \beta^{t-i}\left\|\mathbf{z}_t - \mathbf{A}_t\mathbf{A}_t^T\mathbf{z}_i\right\|^2 \qquad (15)$$

where β is a forgetting factor. Equation (15) is a fourth-order function of the elements in $\mathbf{A}_t$ and hard to optimize. The key point of the PAST algorithm is to approximate $\mathbf{A}_t^T\mathbf{z}_t$, the unknown projection of $\mathbf{z}_t$ onto $\mathbf{A}$, by $\mathbf{y}_i = \mathbf{A}_{t-1}^T\mathbf{z}_i$

$$J'(\mathbf{A}_t) = \sum_{i=1}^{t} \beta^{t-i}\|\mathbf{z}_t - \mathbf{A}_t\mathbf{y}_i\|^2. \qquad (16)$$

The benefits of this approximation are on twofold, the optimal $\mathbf{A}$ is guaranteed by quadratic optimization, and the exponential weighted least square matching in (16) is well studied in adaptive filtering and has the analytic solution as

$$\begin{aligned}\mathbf{A}_t &= \mathbf{C}_{xy}(t)\mathbf{C}_{yy}^{-1}(t)\\ \mathbf{C}_{xy}(t) &= \sum_{i=1}^{t}\beta^{t-i}\mathbf{x}_i\mathbf{y}_i^T\\ &= \beta\mathbf{C}_{xy}(t-1)+\mathbf{x}_t\mathbf{y}_t^T\\ \mathbf{C}_{yy}(t) &= \sum_{i=1}^{t}\beta^{t-i}\mathbf{y}_i\mathbf{y}_i^T\\ &= \beta\mathbf{C}_{yy}(t-1)+\mathbf{y}_t\mathbf{y}_t^T \end{aligned}\tag{17}$$

where $\mathbf{C}_{xy}(t)$ and $\mathbf{C}_{yy}(t)$ are the correlation matrices at time step t. Thus, the updating can be efficiently implemented with the incoming new observation $\mathbf{z}_t^+$.

E. Pair-Wise Constraints

Although the target cannot be detected directly, the low level image features which distinguish the target object from its neighborhood may give some hints about the target. The observation is that if the appearances of two target hypotheses are quite different, it is not possible that both are positive data. To utilize this kinds of pair-wise constraints, here we employ a graph cut algorithm [37] to roughly cluster some sample appearances collected within the predicted target regions. Then we may be able to find possible positive data and negative data from bottom-up. In case we cannot find the good positive data, i.e., no cluster has a small mean distance to the previous target subspace, we can determine some sample windows are not target at least.

Suppose the predicted region for the target is a rectangular region centered at (u, v) with width w and height h. We draw uniform samples (i.e., 15 × 15 image patches) to cover a rectangle region $(u \pm w, v \pm h)$. For each sample patch, the kernel-weighted [4] hue histogram $\mathbf{h}$ with 64 bins is calculated. The affinity matrix, obtained based on the similarity of all pairs of these histograms, is

$$\begin{aligned}\mathbf{S} = [S_{ij}], \quad &\text{where } S_{ij}\\ &= \texttt{exp}\left\{\frac{(\rho(\mathbf{h}_i,\mathbf{h}_j)-\mu)^2}{2\sigma^2}\right\}\end{aligned}\tag{18}$$

where $\rho(\cdot)$ is the Bhattacharya coefficient, μ is the mean of all coefficients, σ is their standard deviation. These sample patches can be grouped into 3–5 clusters by the eigenvalue decomposition of the affinity matrix and selection of the large eigenvalues.

It is not necessary to have a perfect clustering, as observed in our experiments. The image region delineated by the cluster with the minimum mean L^2 distance to the previous target subspace indicates the possible locations that the target may present. In practice, we can simply treat its geometric centroid as the possible location of the target and the corresponding appearance vector as the possible positive data $\mathbf{z}_t^+$. All the other clusters are regarded as negative clusters.

F. Selecting Negative Data

The negative data should be selected carefully. Because if the negative data are too far from the target, the data point may already lie in the orthogonal complement of the target subspace, then minimizing the projections of the negative data may not help. In addition, if the negative data are too close to the target, they may lie partly in the target subspace such that the estimated target subspace is pushed away from its true place.

Our selection of negative data is heuristic based on the clustering in Section III-E. After the positive cluster is selected as in Section III-E, all the other clusters are regarded as negative clusters. In the image regions spanned by all the negative clusters, we find the locations whose appearances (or features) are close to the previous target manifold, and treat these appearance data as negative data $\mathbf{z}_t^-$ in order to distinguish the target from the negative clusters. This heuristic works better in our experiments then some alternative schemes, e.g., selecting the means or geometrical centers of the negative clusters as negative data. The intrinsic idea of negative data is to push the target manifold as far as possible away from the nearby nontargets.

G. Summary of the Tracking Algorithm

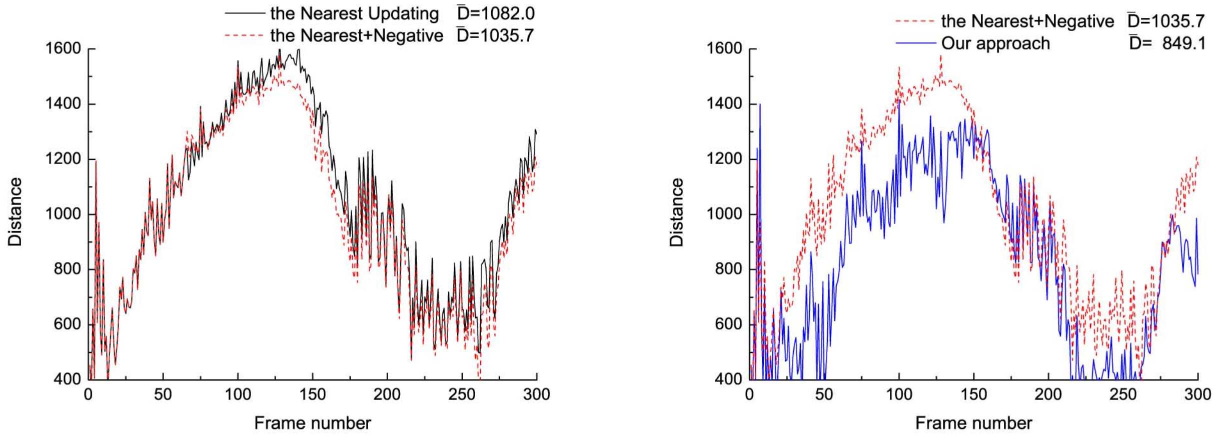
The entire procedure of the proposed algorithm is summarized as follows.

- **Initialization**: At $t = 0$, the target $\mathbf{x}$ is specified by the user input.
- **Iteration**: For $t \geq 1$, perform the following 4 steps iteratively.
- **Step 1**: Generate the affinity matrix by computing the similarities between the hue histograms of the 15 × 15 sample patches within windows with the same size as the target.
- **Step 2**: Cluster the sample windows and select the feature vector of the positive cluster's geometry centroid as $\mathbf{z}_t^+$ and the feature vectors with minimum distances to $\mathbf{P}_{t-1}$ in every negative cluster as $\mathbf{z}_t^-$.
- **Step 3**: Update the subspace described by the projection matrix $\mathbf{P}_t$ according to the methods in Sections III-C and III-D.
- **Step 4**: Draw S samples of $\mathbf{x}_i$ uniformly on the current frame, the one that gives minimum distance to $\mathbf{P}_t$ is the output of the tracker.

IV. Experiments

A. Setup and Comparison Baseline

In our experiments, we aim to recover the motion parameter $\mathbf{x} = \{u, v, s\}$, where (u, v) is the location of the target and s is its scale. The corresponding candidate region is normalized to a 20×20 window and the grey-level pixel intensities are rasterized to a feature vector $\mathbf{z} \in \mathbb{R}^{400}$. At Step 4 of the proposed algorithm, we uniformly sample 100 locations (u, v) on a pixel grid (every two pixels) at three scales s from 0.95, 1.0, and 1.05, so the total number of samples S is 300. The proposed method is implemented using C++ and runs at about 2–5 frames per second on a Pentium-IV 3-GHz desktop without code optimization. The computational intensive modules are the clustering process to identify possible positive and negative data in Section III-E and the iterative updating in Section III-C. Since



(a)

(b)

Fig. 1. Comparison of the distances of the ground truth data to the updated subspaces given by three schemes. (a) Nearest Updating versus Nearest+Negative Updating. (b) Nearest+Negative Updating versus our approach.

during tracking the target appearances may become totally different from what it is in the first frame, we do not apply the conventional remedy of always including the initial appearance in the model [21], [18].

For comparison, we implemented a subspace updating tracker similar to the method in [18], where the nearest appearance observation $\mathbf{z}_i$ to the previous target subspace $\mathbf{P}_{t-1}$ is used to update the orthonormal basis of the subspace by using Gram-Schmidt and dropping the oldest basis. We refer to this method as *Nearest Updating*. This scheme represents the essentials of most of the online adaptation approaches. The method is referred to as $\mathrm{Nearest} + \mathrm{Negative}$ when the positive data are collected by the *Nearest Updating* scheme, and the negative data are employed in updating the same way as in our approach. In all these methods, the adaptation applies every 4 frames.

B. Impact of the Positive and Negative Data

In this quantitative study, we show that the use of negative and possible positive data does help. We have manually annotated a video with 300 frames, in which a head presents a 180° out-of-plane rotation, and collect the ground truth appearance data for each frame (denoted by $\mathbf{z}_t^*$). The comparison is based on the L^2 distance D of the ground truth data $\mathbf{z}_t^*$ to the subspaces estimated by various methods. A smaller distance implies a better method.

As shown in Fig. 1(a), the distance curve for the *Nearest+Negative* scheme with the mean distance $\bar{D} = 1035.7$ is slightly lower than that for the *Nearest Updating* with $\bar{D} = 1082.0$, showing negative data can help to keep the adaptation away from the wrong subspaces. We also observed in our experiments that the negative data themselves may not be able to precisely drive the adaptation to the right places.

We compare the proposed method with the *Nearest+Negative* in Fig. 1(b), in which the curve of our approach is apparently lower than that of the *Nearest Updating*. The mean distance $\bar{D}$ drops from 1035.7 to 849.1. This verifies that the bottom-up positive data do help.

These two comparisons validate that the proposed approach are more capable of following the changes of the nonstationary appearances. Some sample frames are shown in Fig. 2, where the top row is the results of the proposed method, the middle row shows the locations of the possible positive cluster and the possible positive data is enlarged to 60×60 pixels and shown at the top-left corner of each frame, and the bottom row shows the results of the *Nearest Updating* and the nearest data is shown at the top-left corner as well. In this sequence, the proposed method consistently follows the head due to more accurate positive data selected to adapt the target appearance model. The details can be viewed in the video sequence `head180`.[1]

C. Impact of the Clustering Procedure

In this experiment, we compare our method with the *Nearest Updating* in the situation of partial occlusion. We need to track a face, but the partial occlusion makes it difficult when the person drinks and the face moves behind a computer.

When the face moves slowly behind the computer, the *Nearest Updating* drifts and erroneously adapts to a more stable appearance, i.e., a back portion of the computer. In Fig. 3, the top row illustrates this drift process in detail.

The middle row in Fig. 3 presents six appearance data from the possible positive cluster in our method at the 272th frame. Obviously, some of them are not faces, since the clustering is quite rough. But our heuristic of selecting the centroid of the cluster does help and leads to a correct adaptation. Similarly, the bottom row shows the situation of our method at the 284th frame. As the person moves upward, our method correctly follows the face.

This also illustrates that a rough clustering is sufficient for our method which is more robust than the *Nearest Updating*. Some sample frames are shown in Fig. 4, where the top row is our method and the bottom row is that of the *Nearest Updating*. The details of this demo can be viewed in the sequence `face`.

[1] All the test video sequences and the comparisons are available on-line at http://www.ece.northwestern.edu/~mya671/VideoAdaptTIP.htm.

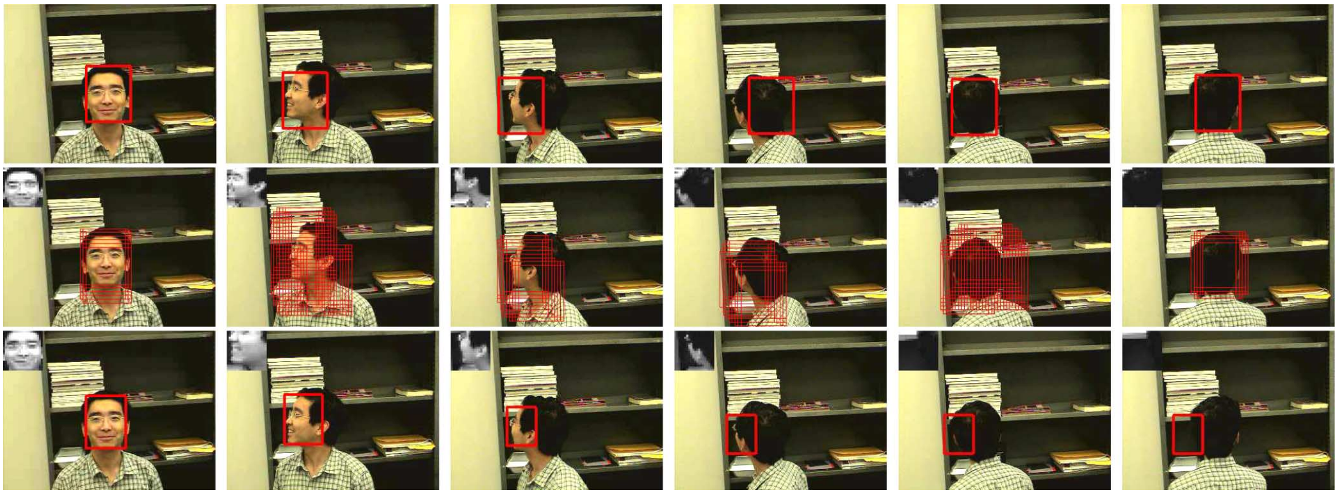


Fig. 2. Tracking a head with 180° rotation [head180]. (top) our method, (middle) the positive cluster, (bottom) Nearest Updating.

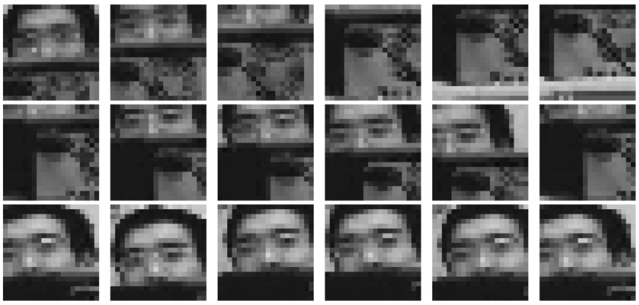


Fig. 3. Clustering performance in the video sequence face: top row shows the drift process of the *Nearest Updating* around frame 272; middle row lists six positive data at frame 272 in our method; bottom row lists six positive data at frame 284 in our method.

D. Tracking a Head With 360° Rotations

Fig. 5 shows the results of tracking a head presenting 360° out-of-plane rotation (The demo is in the sequence head360). The appearances of different views of the head are significantly different, which makes the tracking difficult and also challenges the adaptation. Our experiment shows that the *Nearest Updating* tends to stick to the past appearances and thus reducing the likelihood of including new appearances. For example, when the front face gradually disappears, this scheme is unable to adapt to the hairs to track the back head. In all of our experiments, this scheme loses track when the head fades away. In contrast, since the bottom-up information (i.e., the negative and possible positive data) hints the emerging appearances, our method can successfully track the head, although the bottom-up processing is quite rough.

E. Tracking a Watch With In-Plane Rotations

In general, 2-D in-plane rotation also induces significant changes to the visual appearance of the target. In this experiment, the black background is similar to the panel of the watch such that the adaptation in the *Nearest Updating* deviates from the true subspace and it drifts rapidly. In contrast, although the proposed method is also distracted at frame 444, it is able to recover quickly thanks to the help from the pair-wise constraints. Sample frames are shown in Fig. 6 and details in the sequence watch.

F. Tracking a Face With Large Illumination Changes

In video sequence lighting in Fig. 7, we demonstrate the performance of our algorithm for large illumination changes. The *Nearest Updating* quickly loses the face after the sudden lighting changes, since all observations are far from the target subspace; thus, the samples used in the *Nearest Updating* to update the subspace are kind of random. In contrast, in our method, selecting the centroid of the positive cluster to update the model ensures the samples used are consistent.

G. Tracking a Vehicle With Uneven Illumination Changes

Fig. 8 illustrates tracking a car undergoing uneven illumination changes in video sequence car. When this car is going through the shadow of the trees, the appearance changes substantially. The proposed method finds good positive samples from the clustering procedure to update the subspace model and keeps tracking the car quite robustly.

H. Tracking a Head in Real Environments

Fig. 9 shows the results of the experiment of tracking the head of a person walking (in the sequence walking) in a real environment. The appearance of the head undergoes large changes, and there are also scale changes. The *Nearest Updating* drifts to the background when the appearance of the black hair that the subspace has initially learned almost disappears. This happens when the person moves towards the camera. On the other hand, the proposed method can work comfortably and stably in the case. Further, in Fig. 10, we demonstrate the performance of tracking a head undergoing both large appearance and scale changes under a moving camera. The head rotates more than 180 degree while the scale changes are larger than 50% of the initial size. The proposed method successfully updates the appearance subspace model to follow the target changes.

I. Tracking Multiple Persons With Large Scale Changes

We also test the performance of tracking multiple people in surveillance videos using the CAVIAR video set [38], where

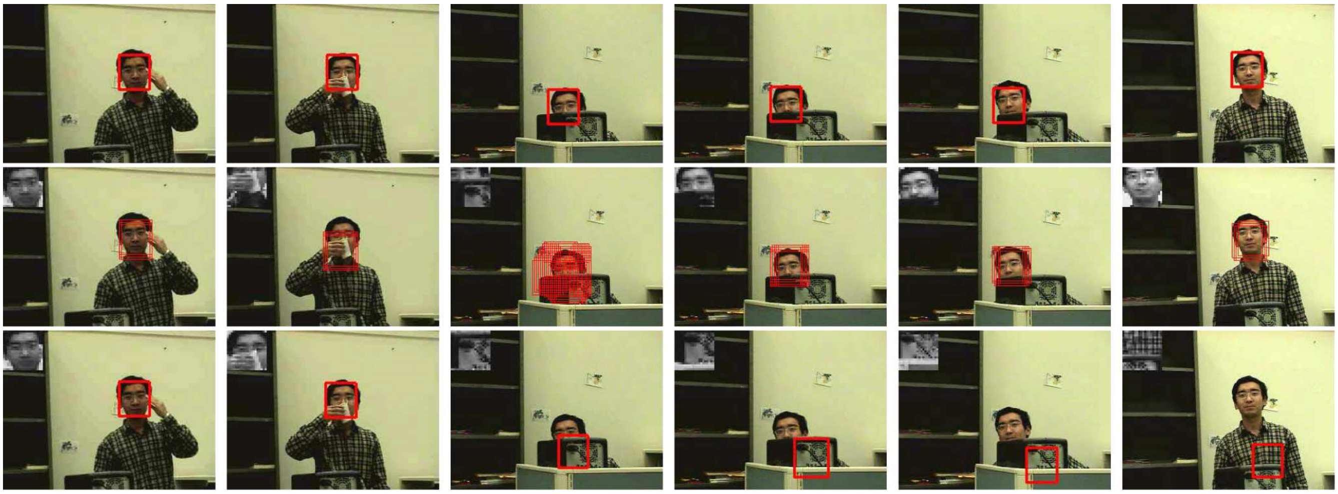


Fig. 4. Tracking a target with partial occlusions [face]. (top) our method, (middle) the positive cluster, (bottom) the Nearest Updating.

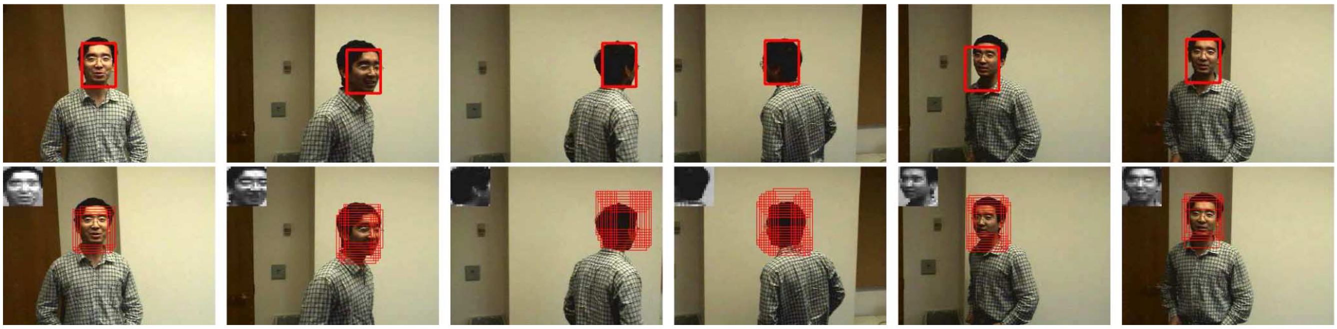


Fig. 5. Tracking a head with 360° out-of-plane rotation [head360]. (top) our method, (bottom) the positive cluster.

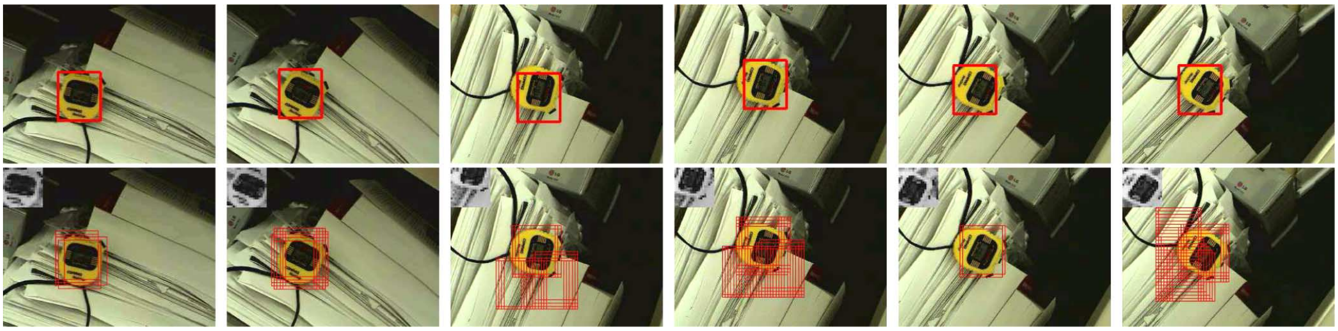


Fig. 6. Tracking a watch with in-plane rotations [watch]. (Top) our method; (bottom) the positive cluster.

the targets are subject to large scale changes but without appearance changes. As shown in Fig. 11, the proposed method can well handle scale changes induced by the people walking away from or towards the camera. In these cases, the *Nearest Updating* method shows comparable performance since the appearance of the target doesn't change too much.

J. Quantitative Experimental Results

We quantitatively evaluate the performance by comparing the proposed approach with the *Nearest Updating* method in terms of the relative position errors between the center of the tracking result and that of the manually labelled ground truth. The relative position error is defined as $d = \|(u,v)-(u_{gt},v_{gt})\|/s_{gt}$ where $\{u_{gt},v_{gt},s_{gt}\}$ is the labelled target location and scale. This measurement enables comparing the accuracy of tracking results for targets with different sizes. Perfect tracking performance should have position error around $0.d = 0.1$ indicates the tracker deviates away 10% of the true target size.

The results are summarized in Table I, where # of frm. indicates the number of frames annotated, and the mean relative position error and its standard deviation are denoted as $\bar{d}$ and d_σ for the proposed method, and $\bar{d}(NU)$ and $d_\sigma(NU)$ for the *Nearest Updating* method, respectively. Both the mean relative errors and the standard deviations of all sequences are consistently smaller than that of the *Nearest Updating* method, which indicates the proposed method is more accurate and stable (one exception is the tracking results of the person in a red jacket in the sequence corridor, i.e., the top row in Fig. 11, where the

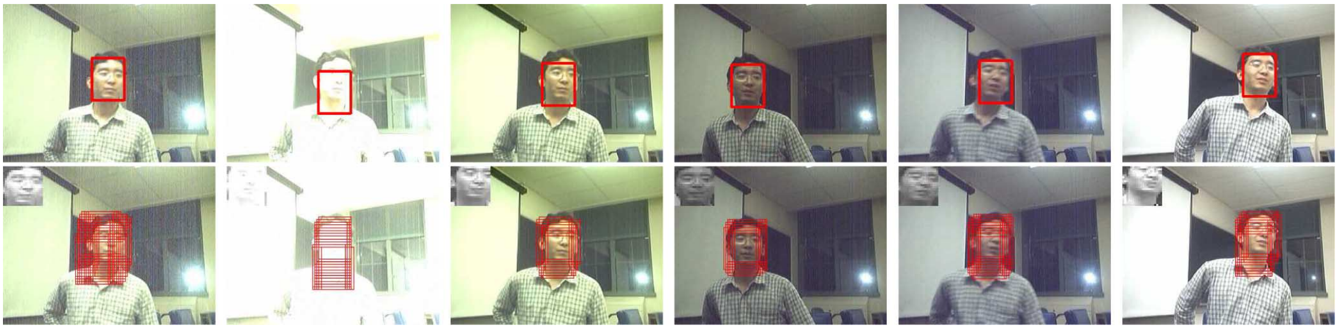


Fig. 7. Tracking a face with large illumination changes [`lighting`]. (top) our method, (bottom) the positive cluster.

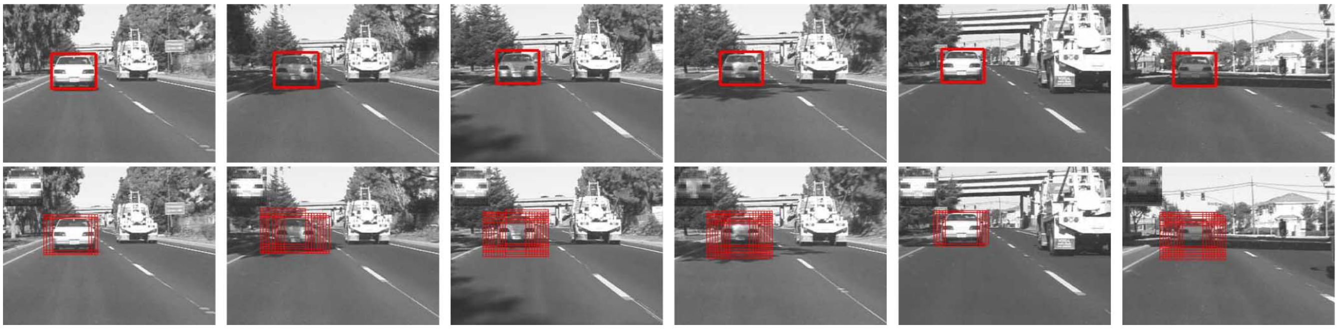


Fig. 8. Tracking a vehicle with uneven illumination changes [`car`]. (Top) our method; (bottom) the positive cluster.

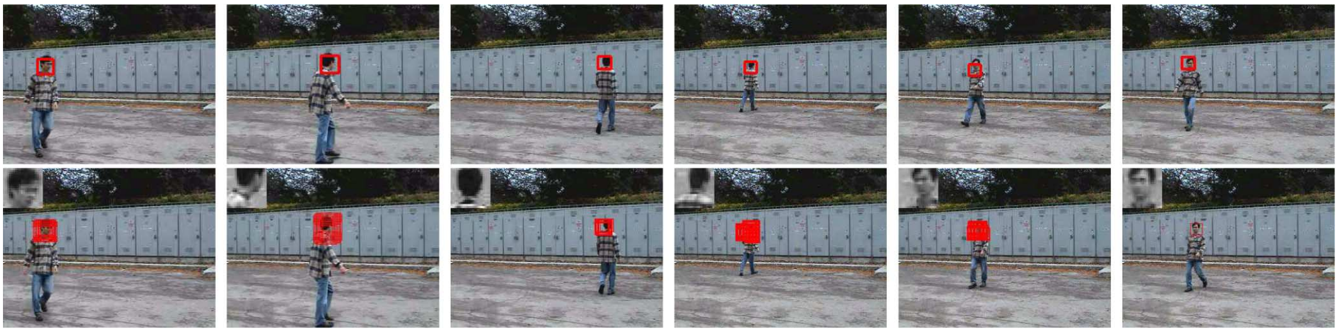


Fig. 9. Tracking a head in real environments [`walking`]. (Top) our method; (bottom) the positive cluster.

target has no appearance changes.). The frame-by-frame comparisons of the relative position errors d are illustrated in Fig. 12.

K. Discussions

All the above experiments have validated the proposed approach. When the target experiences drastic changes, we can explain the reason why the methods sharing the same nature as the *Nearest Updating* deteriorate in two aspects. First, these methods tend to adhere to the old model as much as possible and are reluctant to include the changes. When the model changes completely or the original features disappear, the updated model will drift away from the true one eventually. Second, when the drift starts, there is no mechanism in these methods to force them back; thus, the drift is unstable and catastrophic.

In contrast, since our method utilizes the information from bottom-up, it can be thought as feedbacks that guide the target observation adaptation to be distinguishable from the nearby environment. The combination of both bottom-up and top-down information makes our method stable and avoids catastrophic drift to a large extent. As a result, the proposed method can be more robust and stable to cope with the adaptation drift.

The main limitation is the dependence on good bottom-up low-level features with high discriminative power. For example, in case camouflage objects are in the close vicinity of the target, the color features employed may not yield good clustering results; thus, the selection of positive and negative data may not work well. One failure case is shown in Fig. 13 (tracking the person in a black jacket on the right in the sequence of `corridor`, i.e., the top row in Fig. 11). When two people with nearly identical appearances cross each other, the positive data are hard to select; thus, the appearance subspace erroneously adapts to the middle region of the two people.

V. Conclusion

We have investigated the adaptation problem in subspace tracking. If no constraints are imposed, this problem is ill-posed.

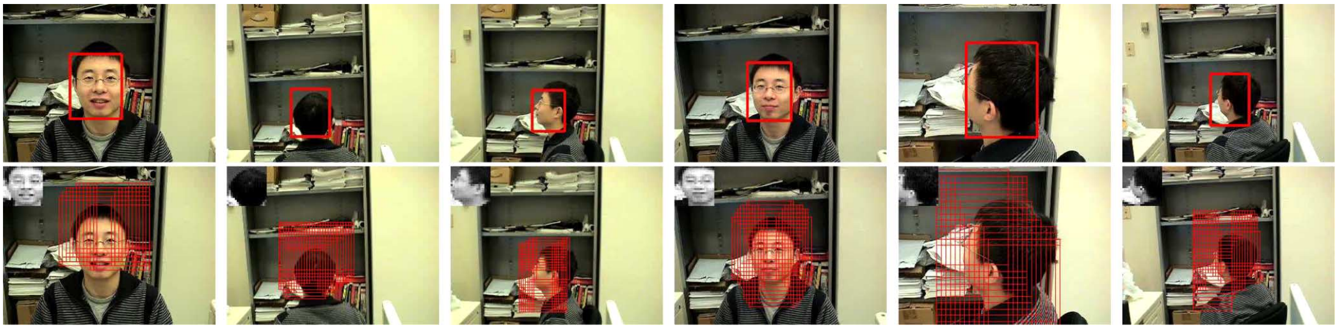


Fig. 10. Tracking a head with both appearance changes and scale changes [head180AB]. (Top) our method; (bottom) the positive cluster.

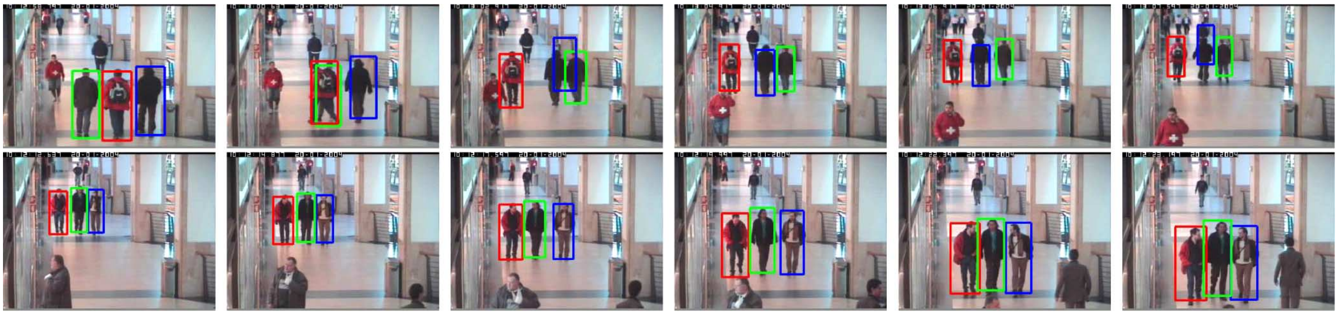


Fig. 11. Tracking multiple people with large scale changes [corridor]. (Top) Three people walking away from the camera; (bottom) three people walking towards the camera.

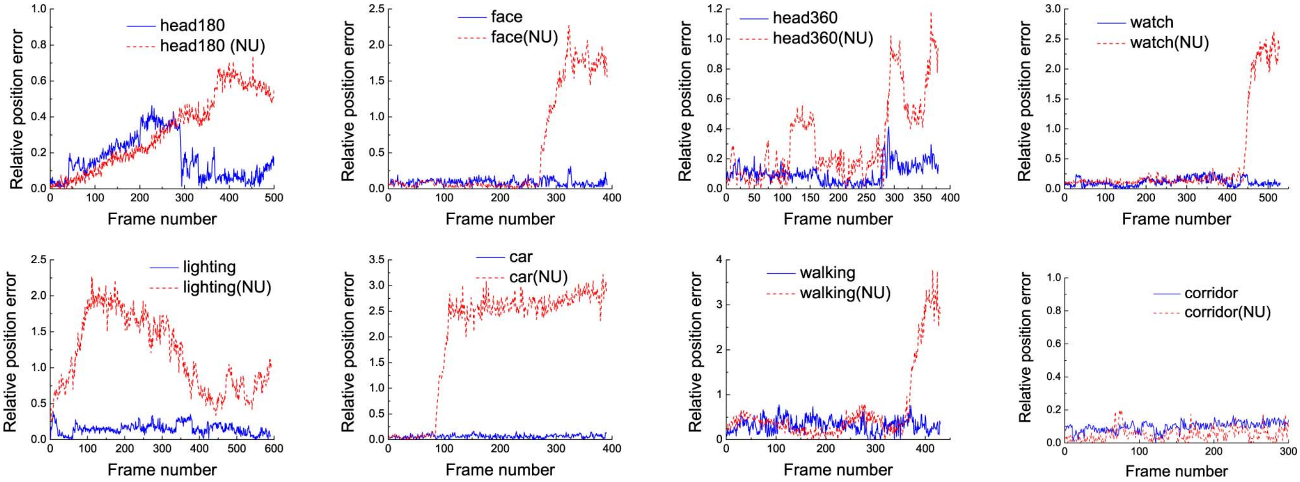


Fig. 12. Frame-by-frame comparisons of the relative position errors d.

TABLE I
SUMMARY OF THE QUANTITATIVE EXPERIMENTAL RESULTS

Sequence	# of frm.	d	d_σ	$d(NU)$	$d_\sigma(NU)$
head180	500	0.1663	0.1189	0.3172	0.2054
face	390	0.0899	0.0479	0.5105	0.7251
head360	380	0.1038	0.0662	0.3115	0.2717
watch	530	0.1130	0.0661	0.4589	0.7472
lighting	590	0.1500	0.0709	1.1834	0.5170
car	390	0.0652	0.0341	2.0132	1.0895
walking	430	0.3178	0.1552	0.6549	0.8321
corridor	300	0.0889	0.0288	0.0527	0.0397

Instead of the commonly used nearest updating scheme, we propose to impose both top-down smoothness constraints and the bottom-up data-driven constraints from current observations.

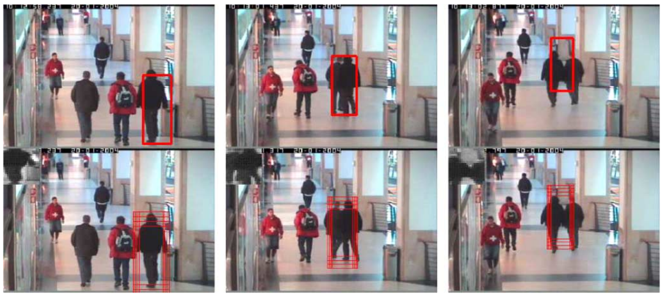


Fig. 13. Tracking a person with a camouflage object nearby [corridor]. (Top) our method; (bottom) the positive cluster.

Our method balances three factors: 1) distance of positive data to the subspace, 2) the projections of the negative data, and 3) the smoothness of two consecutive subspaces. The proposed method can largely alleviate the risk of adaptation drift and thus achieves better tracking performance.

Our further study will include the investigation of the situation when the smoothness constraint and bottom-up information are contradicted, and the best way of balancing and fusing these three types of constraints.

Appendix A

Lemma 1: The solution of the following problem:

$$\min_A \mathtt{tr}(\mathbf{A}^T\mathbf{C}\mathbf{A}), s.t., \mathbf{A}^T\mathbf{A} = I \tag{19}$$

where $\mathbf{A} \in \mathbb{R}^{m\times r}$, and $\mathbf{C} = \mathbf{Z}\mathbf{Z}^T \in \mathbb{R}^{m\times m}$, is given by the eigenvectors that corresponds to the r smallest eigenvalues of $\mathbf{C}$.

Proof: It is easy to figure it out. Actually this is the same as the proof of PCA.

Based on the Lemma, the Proof of Theorem 1 is given by the following: Performing SVD on $\mathbf{A}_t$, we have $\mathbf{A}_t = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{U} \in \mathbb{R}^{m\times r}$, $\mathbf{\Sigma} \in \mathbb{R}^{r\times r}, \mathbf{V} \in \mathbb{R}^{r\times r}$. It is easy to see: $\mathbf{P}_t = \mathbf{U}\mathbf{U}^T$. Then the optimization problem in (8) is equivalent to

$$\begin{aligned}\min_U J_3(\mathbf{U}) = \min_U \Big\{&\mathtt{tr}\left(\mathbf{U}^T\mathbf{C}_t^-\mathbf{U}\right)\\ &-\mathtt{tr}\left(\mathbf{U}^T\mathbf{C}_t^+\mathbf{U}\right)\\ &+\alpha\left\|\mathbf{U}\mathbf{U}^T - \mathbf{P}_{t-1}\right\|_F^2\Big\}\\ s.t.\mathbf{U}^T\mathbf{U} = \mathbf{I}.&\end{aligned} \tag{20}$$

The Lagrangian is given by

$$L(\mathbf{U}) = J_3(\mathbf{U}) + \lambda(\mathbf{U}^T\mathbf{U} - \mathbf{I}). \tag{21}$$

Let $\mathbf{U} = [\mathbf{e}_1, \ldots, \mathbf{e}_r]$, and we have

$$\begin{aligned}\frac{\partial L}{\partial \mathbf{e}} &= 2\left(\mathbf{C}_t^- - \mathbf{C}_t^+\right)\mathbf{e} + 2\alpha(\mathbf{e}\mathbf{e}^T - \mathbf{P}_{t-1})\mathbf{e} + 2\lambda\mathbf{e}\\ &= 2\left(\mathbf{C}_t^- - \mathbf{C}_t^+ + \alpha\mathbf{I} - \alpha\mathbf{P}_{t-1}\right)\mathbf{e} + 2\lambda\mathbf{e}.\end{aligned} \tag{22}$$

Thus, $\mathbf{e}$ is an eigenvector of $\hat{\mathbf{C}} = \mathbf{C}_t^- - \mathbf{C}_t^+ + \alpha\mathbf{I} - \alpha\mathbf{P}_{t-1}$. The minimization problem is solved by finding the r eigenvectors that correspond to the r smallest eigenvalues of $\hat{\mathbf{C}}$. **Q.E.D.**

Acknowledgment

The authors would like to thank the anonymous reviewers for their constructive comments. They would also like to thank S. Dai and J. Yuan for their help with recording the test video sequences.

References

[1] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," in *Proc. 4th Eur. Conf. Computer Vision*, Cambridge, U.K., 1996, pp. 343–356.

[2] Y. Wu and T. S. Huang, "A co-inference approach to robust visual tracking," in *Proc. IEEE Int. Conf. Computer Vision*, Vancouver, BC, Canada, Jul. 7–14, 2001, vol. 2, pp. 26–33.

[3] K. Nummiaro, E. Koller-Meierb, and L. V. Gool, "An adaptive color-based particle filter," *Image Vis. Comput.*, vol. 21, no. 1, pp. 99–110, Jan. 2003.

[4] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *IEEE Conf. Computer Vision and Pattern Recognition*, Hilton Head Island, SC, Jun. 13–15, 2000, vol. 2, pp. 142–149.

[5] G. Hager, M. Dewan, and C. Stewart, "Multiple kernel tracking with SSD," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Washington, DC, Jun. 27–Jul. 2 2004, vol. 1, pp. 790–797.

[6] Z. Fan, Y. Wu, and M. Yang, "Multiple collaborative kernel tracking," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, San Diego, CA, Jun. 20–26, 2005, vol. 2, pp. 502–509.

[7] M. Dewan and G. D. Hager, "Toward optimal kernel-based tracking," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, NYC, Jun. 17–22, 2006, vol. 1, pp. 618–625.

[8] P. N. Belhumeur and D. J. Kriegman, "What is the set of images of an object under all possible lighting conditions," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, San Francisco, CA, Jun. 18–20, 1996, pp. 270–277.

[9] F. Leymarie and M. D. Levine, "Tracking deformable object in the plane using an active contour model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 6, pp. 617–634, Jun. 1993.

[10] S. Birchfield, "Elliptical head tracking using intensity gradients and color histograms," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Santa Barbara, CA, Jun. 23–25, 1998, pp. 232–237.

[11] Y. Wu, J. Lin, and T. S. Huang, "Capturing natural hand articulation," in *Proc. IEEE Int. Conf. Computer Vision*, Vancouver, BC, Canada, Jul. 7–14, 2001, vol. 2, pp. 426–432.

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

[13] M. J. Black and A. D. Jepson, "Eigentracking: Robust matching and tracking of articulated objects using a view-based representation," in *Proc. 4th Eur. Conf. Computer Vision*, Cambridge, U.K., Apr. 1996, pp. 329–342.

[14] G. Hager and P. Belhumeur, "Real-time tracking of image regions with changes in geoemtry and illumination," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, San Francisco, CA, Jun. 18–20, 1996, pp. 403–410.

[15] K. Toyama and A. Blake, "Probabilistic tracking in a metric space," in *Proc. IEEE Int. Conf. Computer Vision*, Vancouver, BC, Canada, Jul. 7–14, 2001, vol. 2, pp. 50–57.

[16] A. Elgammal, "Learning to track: Conceptual manifold map for closed-form tracking," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, San Diego, CA, Jun. 20–26, 2005, vol. 1, pp. 724–730.

[17] A. D. Jepson, D. Fleet, and T. El-Maraghi, "Robust online appearance models for visual tracking," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Kauai, HI, Dec. 9–14, 2001, vol. 1, pp. 415–422.

[18] J. Ho, K.-C. Lee, M.-H. Yang, and D. J. Kriegman, "Visual tracking using learned linear subspace," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Jun. 27–Jul. 2 2004, vol. 1, pp. 782–789.

[19] J. Vermaak, P. Perez, M. Gangnet, and A. Blake, "Towards improved observation models for visual tracking: Selective adaptation," in *Proc. 7th Eur. Conf. Computer Vision*, Copenhagen, Denmark, May 2002, vol. 1, pp. 645–660.

[20] J. Lim, D. Ross, R.-S. Lin, and M.-H. Yang, "Incremental learning for visual tracking," in *Proc. Advances in Neural Information Processing Systems 17*, Vancouver, BC, Canada, Dec. 13–18, 2004, pp. 801–808.

[21] R. T. Collins and Y. Liu, "On-line selection of discriminative tracking features," in *Proc. IEEE Int. Conf. Computer Vision*, Nice, France, Oct. 13–16, 2003, vol. 1, pp. 346–352.

[22] J. Wang, X. Chen, and W. Gao, "Online selecting discriminative tracking features using particle filter," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, San Diego, CA, Jun. 20–26, 2005, vol. 2, pp. 1037–1042.

[23] S. Avidan, "Ensemble tracking," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, San Diego, CA, Jun. 20–25, 2005, vol. 2, pp. 494–501.

[24] A. P. Leung and S. Gong, "Online feature selection using mutual information for real-time multi-view object tracking," in *Proc. IEEE Int. Workshop Analysis and Modeling of Faces and Gestures*, Beijing, China, Oct. 16, 2005, pp. 184–197.

[25] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *Proc. Brit. Machine Vision Conf.*, Edinburgh, 4–7, 2006, vol. 1, pp. 47–56.

[26] S. K. Zhou, R. Chellappa, and B. Moghaddam, "Visual tracking and recognition using appearance-adaptive models in particle filters," *IEEE Trans. Image Process.*, vol. 13, no. 11, pp. 1491–1506, Nov. 2004.

[27] F. Tang and H. Tao, "Object tracking with dynamic feature graph," in *Proc. IEEE Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, Beijing, China, Oct. 15–16, 2005, pp. 25–32.

[28] Z. Yin and R. Collins, "On-the-fly object modeling while tracking," presented at the IEEE Conf. Computer Vision and Pattern Recognition, Minneapolis, MN, Jun. 17–22, 2007.

[29] M. Yang and Y. Wu, "Tracking non-stationary appearances and dynamic feature selection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, San Diego, CA, Jun. 20–26, 2005, vol. 2, pp. 1059–1066.

[30] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Seattle, WA, Jun. 21–23, 1994, pp. 593–600.

[31] B. Han and L. Davis, "On-line density-based appearance modelling for object tracking," in *Proc. IEEE Int. Conf. Computer Vision*, Beijing, China, Oct. 17–21, 2005, vol. 2, pp. 1492–1499.

[32] D. Ross, J. Lim, and M.-H. Yang, "Adaptive probabilistic visual tracking with incremental subspace update," in *Proc. 8th Eur. Conf. Computer Vision*, Prague, Czech Republic, May 2004, vol. 1, pp. 215–227.

[33] K.-C. Lee, J. Ho, M.-H. Yang, and D. Kriegman, "Video-based face recognition using probabilistic appearance manifolds," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Madison, WI, Jun. 18–20, 2003, vol. 1, pp. 313–320.

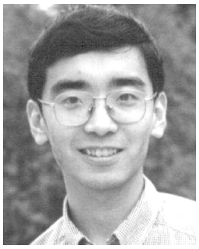
[34] H. Lim, V. I. Morariu, O. I. Camps, and M. Sznaier, "Dynamic appearance modeling for human tracking," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Jun. 17–22, 2006, vol. 1, pp. 751–757.

[35] K.-C. Lee and D. J. Kriegman, "Online learning of probabilistic appearance manifolds for video-based recognition and tracking," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, San Diego, CA, Jun. 20–25, 2005, vol. 1, pp. 852–859.

[36] B. Yang, "Projection approximation subspace tracking," *IEEE Trans. Signal Process.*, vol. 43, no. 1, pp. 95–107, Jan. 1995.

[37] J. Shi and J. Malik, "Normalized cuts and image segmentation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, San Juan, PR, Jun. 17–19, 1997, pp. 731–737.

[38] EC Funnded CAVIAR Project/IST 2001 37540 [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/caviar/



Ming Yang (M'08) received the B.E. and M.E. degrees in electronic engineering from Tsinghua University, Beijing, China, in 2001 and 2004, respectively, and the Ph.D. degree in electrical and computer engineering from Northwestern University, Evanston, IL, in June 2008.

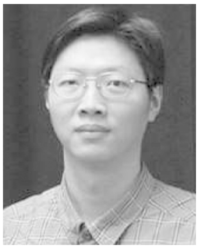
From 2004 to 2008, he was a Research Assistant with Prof. Y. Wu in the Computer Vision Group, Northwestern University. He joined NEC Laboratories America, Cupertino, CA, as a research staff member upon his graduation. His research interests include computer vision, machine learning, video communication, medical image analysis, and intelligent multimedia content analysis.

Dr. Yang was an excellent bachelor graduate of Tsinghua University, 2001. He was also awarded the excellent student fellowship from 1998 to 2003 at Tsinghua University.

Zhimin Fan received the B.S. and M.S. degrees with honors both from the Automation Department, Tsinghua University, Beijing, China, in 2001 and 2004, respectively, and he also received the M.S. degree from the Electrical Engineering and Computer Science Department, Northwestern University, Evanston, IL, in December 2005.

His research interests include computer vision, pattern recognition, and image processing.

Jialue Fan (S'08) received the B.S. and M.S. degrees from the Electronic Engineering Department, Tsinghua University, Beijing, China, in 2005 and 2007, respectively. He is currently pursuing the Ph.D. degree in the Computer Vision Group, Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL. His research interests include computer vision, pattern recognition, and image and video processing.



Ying Wu (SM'06) received the B.S. degree from the Huazhong University of Science and Technology, Wuhan, China, in 1994, the M.S. degree from Tsinghua University, Beijing, China, in 1997, and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign (UIUC), Urbana, in 2001.

From 1997 to 2001, he was a Research Assistant at the Beckman Institute for Advanced Science and Technology, UIUC. During summer 1999 and 2000, he was a research intern with Microsoft Research, Redmond, WA. In 2001, he joined the Department of Electrical and Computer Engineering, Northwestern University, Evanston, IL, as an Assistant Professor. He is currently an Associate Professor of electrical engineering and computer science at Northwestern University. His current research interests include computer vision, image and video analysis, pattern recognition, machine learning, multimedia data mining, and human-computer interaction.

Dr. Wu serves as an associate editor for IEEE TRANSACTIONS ON IMAGE PROCESSING, the SPIE *Journal of Electronic Imaging*, and the IAPR *Journal of Machine Vision and Applications*. He received the Robert T. Chien Award at UIUC in 2001 and the National Science Foundation CAREER Award in 2003.