Spatio-Temporal Context for Robust Multitarget Tracking

Hieu T. Nguyen, *Member*, *IEEE*, Qiang Ji, *Senior Member*, *IEEE*, and Arnold W.M. Smeulders, *Senior Member*, *IEEE*

Abstract—In multitarget tracking, the main challenge is to maintain the correct identity of targets even under occlusions or when differences between the targets are small. The paper proposes a new approach to this problem by incorporating the context information. The context of a target in an image sequence has two components: the spatial context including the local background and nearby targets and the temporal context including all appearances of the targets that have been seen previously. The paper considers both aspects. We propose a new model for multitarget tracking based on the classification of each target against its spatial context. The tracker searches a region similar to the target while avoiding nearby targets. The temporal context is included by integrating the entire history of target appearance based on probabilistic principal component analysis (PPCA). We have developed a new incremental scheme that can learn the full set of PPCA parameters accurately online. The experiments show robust tracking performance under the condition of severe clutter, occlusions, and pose changes.

Index terms—Multitarget tracking, context-based tracking, probabilistic PCA.

1 Introduction

Tracking an object can be seen as a dynamic classification of one target against everything else. When more than one object is being tracked, the problem evolves into dynamic multiclass classification.

The problem of jointly tracking multiple targets is as challenging as it is important. In video surveillance, a common interest is in the detection of suspicious behavior by patterns of movement of people. In team sports, the interest is usually in patterns of play. In traffic control, it usually requires the tracking of many vehicles in the road. All of these cases illustrate the importance of multiple target tracking in real-life applications.

The challenging aspect of multiobject tracking is in safeguarding the proper identity of all targets. This is especially hard when objects have little distinction in their appearance. Another aspect of the problem is occlusion between targets passing in front and behind each other and occlusion behind a part of the scene. In addition, there are the usual aspects of tracking such as change in pose and change in the illumination of the scene. We aim to maintain object identity in these conditions, here for the case of a fixed camera.

The traditional approach in resolving the ambiguous identity of several targets is to separate them whenever possible. The common principle is that, once a target is assigned to a position in the image, no more targets can occupy that place. The classical methods, including the joint probabilistic data association filter in [2] and the multiple hypotheses tracking algorithm in [20], enforce a data association variable into the target likelihood. It rules out configurations where multiple targets associate with the same image region. Recent methods [8], [26], [10] add a prior term to the likelihood to prevent any pair of targets from getting too close. Preventing proximity and eventually occlusion between targets by constraints is undesired, however, since that information is usually what one would like to know in tracking. In addition, by posing constraints on target positions, the references may succeed in avoiding coalescence of targets, but they still may yield an undesired switching of identity. In most references, each target is searched for by maximizing its likelihood while ignoring the others. The sensitivity of the likelihood to changes in appearance may then provoke false classifications. The joint likelihood models in [19], [9], [27] better describe the overlap between targets during occlusions, but they still minimize the likelihood of each individual target.

We believe that the accurate identification of targets should be based on the two following conditions:

1. the ability to distinguish between targets moving close to each other and between each target and the background and
2. an accurate appearance model of each target that should be robust to occlusions and other types of appearance changes occurring during tracking.

We propose to achieve these goals by incorporating the appearance information from the context of each target. Two types of context are considered: spatial and temporal. The spatial context of a target involves the local background and other foreground objects present in the current frame. The temporal context includes all prior appearances of the target.

We develop a new probabilistic framework for multitarget tracking by a built-in classifier for the distinction of targets against their spatial context. Robustness to occlusions is achieved by modeling the appearance of a target

- *H.T. Nguyen and Q. Ji are with the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY 12180. E-mail: nguyen@ecse.rpi.edu, jiq@rpi.edu.*
- *A.W.M. Smeulders is with the Intelligent Systems Lab Amsterdam (ISLA), University of Amsterdam, Faculty of Science, Kruislaan 403, NL-1098 SJ, Amsterdam, The Netherlands. E-mail: smeulders@science.uva.nl.*

Manuscript received 25 Aug. 2005; revised 11 May 2006; accepted 18 May 2006; published online 13 Nov. 2006.
Recommended for acceptance by P. Fua.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0464-0805.

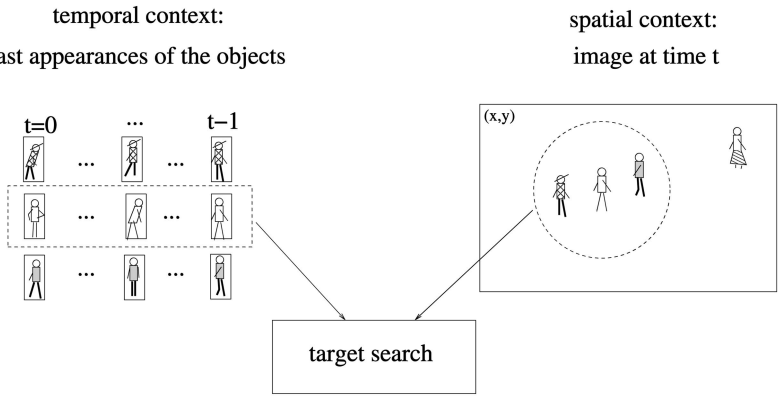


Fig. 1. Detection of a target object in a video frame relies on two types of context information. This includes the contrast of the object with respect to its surrounding and a memory of object's appearances.

using probabilistic principle component analysis (PPCA). The PPCA model integrates all appearances seen in the past. As we need to update the PPCA at each new frame, we have developed a new, incremental PPCA scheme which computes all parameters accurately and efficiently.

The paper is structured as follows: Section 2 presents our concept of spatio-temporal context and its role for tracking. Related work is discussed in Section 3. Section 4 presents our framework for multitarget tracking, including the probabilistic model and inference. In Section 5, we present a new method for incremental probabilistic PCA. This algorithm is used for the construction of the appearance model of each target. The tracking results are demonstrated in Section 6.

2 Spatio-Temporal Context-Based Tracking

When considering tracking as an object detection problem in a dynamically evolving environment, the detection model can be learned either offline or online, or by a combination thereof. Offline learning, in turn, may be target specific or general. Target-specific and complete models encompassing all appearances of the target severely limit the applicability of the model as it requires an explicit recording session. Offline learning of a general model is a much more realistic approach but it tends to miss important details in the appearance of the target. For example, one may have a detailed appearance model of humans, but it is still difficult to fill in the appearance of all possible clothing. In general, the detection model is best when constructed specifically for the online context where the tracking takes place. Incorporation of context information into the object detection model is therefore very important. This issue is the focus of this paper.

Since an image sequence has the spatio-temporal characteristic, the context information of a moving target has two components: the spatial context and the temporal context, see Fig. 1. These contexts naturally correspond to the two sources of information both used by the human vision system to localize a moving object:

1. the appearance distinction between the object and its surroundings at the specific moment of time and
2. the memory about the appearance of the object, which is acquired over time.

Both the spatial context and the temporal context should be taken into account.

In multitarget tracking, the spatial context of a target involves the appearance information of the background and the nearby targets. The detection model should be designed to best discriminate each target from its spatial context. Consequently, the algorithm should search for the image region similar to the target while avoiding regions of the background and other targets. In particular, if there are two regions where the target has a high likelihood, the algorithm should select the region which, at the same time has, a lower likelihood to be a part of the background or any of the other targets.

A good detection scheme in individual frames cannot last long with a poor memory of targets' appearance. This is why the temporal context is also needed. To incorporate the temporal context, for each target we use an appearance model covering all appearances of the object as seen in the past. The contribution of past appearances makes the model robust to occlusions or illumination changes. The model should then be able to recognize the object when past appearances return in the future. While the model should be adaptive to new appearances of the object, a long-term memory of all appearances will also help to reduce the drift usually happening in adaptive tracking.

3 Related Work

The spatial context has been considered in single target tracking, particularly by the trackers using the discriminative approach. For a single target, the spatial context information is the appearance of the surrounding background. In [5], the authors propose to select online color features most discriminating a target object from a local background window. The algorithm in [16] learns and maintains online a foreground-background discriminant function as the objective function in the target search. The papers indicate that the improvement of the distinction between the target and the surrounding context increases the robustness to varying appearances of the object. In [1], [25], the objective function is a classifier that is trained offline to discriminate the object class of interest from the nonobject class. The use of a substantial amount of prior knowledge in offline training can provide a powerful classifier, but does not take into account the specifics of an individual object. This could be a problem for tracking multiple objects of the same class since all targets would have the same objective function.

For modeling the target appearance over a long temporal context, the generative statistical models including eigenspace or mixture of Gaussians can be applied. An important requirement for such a model is that it can be learned incrementally upon the arrival of new tracking results and under the condition of a limited memory and a limited time. The algorithms for online learning of a mixture of Gaussians [24] require that the input samples be statistically independent and, furthermore, need time to converge. Recent tracking algorithms therefore focus on the eigenspace model [21], [13], [14], [7]. They rely on the recursive SVD algorithm [12] to update the eigenvectors of a data stream incrementally. Eigenvectors alone, however, do not provide a probabilistic measure to characterize object likelihood in the full feature space. The probabilistic formulation of the eigenspace model, well known as the PPCA (probabilistic PCA) [23], [15], requires an additional parameter being the variance of the noise in nonprincipal components. This parameter scales the distance from data to the subspace of the principal components, allowing for a natural combination with distance measures within that subspace. In an existing method [14], this noise parameter is predefined or set to a fraction of the eigenvalue of the smallest principal component. This ad hoc approach has no theoretical justification. Another incremental scheme of PPCA recently proposed in [4] is of a rather different vein. It first performs a batch PPCA on newly arrived samples and then merges the new PPCA and the existing PPCA using a plain incremental PCA method. The problem of this approach is inaccuracy of the estimation of PPCA for the small number of incoming additional samples. In particular, this method will not work when the number of new observations is smaller than the number of principal components. In addition, this method is not based on maximum likelihood, which is usually required in parameter estimation. In [11], PPCA is used for object tracking, but the model is learned offline.

Although not explicitly stated, the current tracking algorithms have used the context information for handling background clutter and appearance variations. However, the algorithms are restricted to the tracking of a single target. A probabilistic framework for context-based tracking in case of multiple targets is still lacking. Furthermore, while PPCA appears effective for modeling appearances of a target over a long temporal context, there is no incremental scheme that calculates the full set of the PPCA parameters online. These issues will be addressed in the presented paper.

Our context-based tracking model is learned solely online. As is frequently the case, in practice, we assume that no offline training is possible except for the initialization of the target region in the first frame of the sequence. Acknowledging the power of offline learning, we leave the interesting issue of combining offline and online models for future research.

4 Classification-Based Framework for Tracking Multiple Targets

We first present a novel classification-based framework for multitarget tracking.

Let M be the number of targets that we want to track, and $\boldsymbol{x}_i$ be the position of the ith target. For simplicity of the presentation, we consider only translational motion, although the method can also be extended for more sophisticated types of motion. The goal of the tracking is to estimate the concatenation of the position of all targets: $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M\}$.

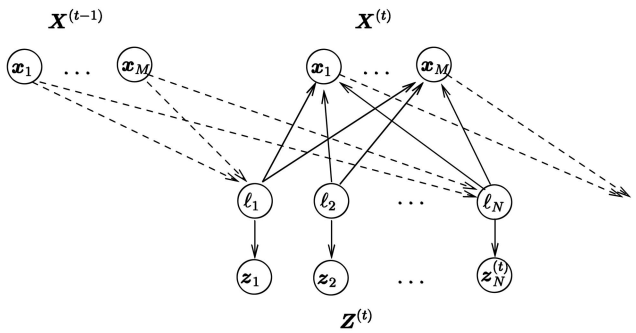


Fig. 2. The proposed probabilistic model for multitarget tracking. $\boldsymbol{x}_k$ is the position of kth target, $\boldsymbol{z}_i$ is the observation at position i, and ℓ_i is the class label of position i.

4.1 The Probabilistic Model

We propose the probabilistic model shown in Fig. 2. In this model, $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M\}$ is the state vector. Let $\mathcal{P} = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_N\}$ denote the set of all possible positions in the image. $\boldsymbol{Z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N\}$ is the set of measurements where $\boldsymbol{z}_i$ denotes the vectors assembled from the intensities in a neighborhood of position p_i. The size of this neighborhood will be elucidated in Section 5. To achieve accurate target identification, a classifier is integrated in the tracking by hidden class labels $\ell_1, \ldots, \ell_N$. Each class label $\ell_i \in \{0, 1, \ldots, M\}$ indicates the label of the target at location $\boldsymbol{p}_i$. The label 0 is the background label indicating that no target occupies the position. The main idea of the proposed approach is that the tracker first estimates the distribution of the label at every position and then locates each target at the position where the corresponding label has highest probability.

We use superscript (t) to denote time. For ℓ_i, however, we drop t as we use only labels at time t. Given the previous tracking result $\boldsymbol{X}^{(t-1)}$ and the current measurements $\boldsymbol{Z}^{(t)}$, inference about $\boldsymbol{X}^{(t)}$ is made based on three distributions: the predicted label distribution $P(\ell_i|\boldsymbol{X}^{(t-1)})$, the measurement distribution $p(\boldsymbol{z}_i^{(t)}|\ell_i)$, and the position distribution $p(\boldsymbol{x}_k^{(t)}|\ell_1, \ldots, \ell_N)$. The labels are assumed mutually independent conditioned on $\boldsymbol{X}^{(t-1)}$, implying that there is no dependence between the position of targets. This assumption may not be the case sometimes, for example, in a soccer play where the position of the keeper is always correlated with the defenders. However, it should not cause any serious problem as long as the current measurements and the predicted prior are sufficient to distinguish the targets. The assumption of the independence of the labels also implies that no constraint is placed on the target positions. In particular, this property allows the targets to move close to each other. The algorithm distinguishes the targets mainly by discriminating their appearance. The posterior distribution of each label $P(\ell_i|\boldsymbol{X}^{(t-1)}, \boldsymbol{z}_i^{(t)})$ can be calculated straightforwardly from $p(\ell_i|\boldsymbol{X}^{(t-1)})$ and $p(\boldsymbol{z}_i^{(t)}|\ell_i)$. The distribution of the position of each target $p(\boldsymbol{x}_k^{(t)} = \boldsymbol{p}_i|\boldsymbol{Z}^{(t)}, \hat{\boldsymbol{X}}^{(t-1)})$ is then independently inferred using $P(\ell_i|\boldsymbol{X}^{(t-1)}, \boldsymbol{z}_i^{(t)})$ and $p(\boldsymbol{x}_k^{(t)}|\ell_1, \ldots, \ell_N)$. This probability will be used for the target search.

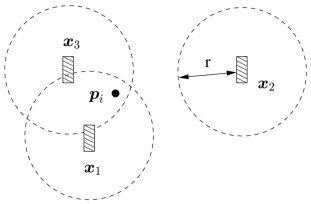


Fig. 3. The prediction of the label prior probability. In this example, only targets 1 and 3 contribute to the label prior at $\boldsymbol{p}_i$.

The three aforementioned distributions are defined as follows:

1. *The predicted label distribution* $P(\ell_i|\boldsymbol{X}^{(t-1)})$. The probability depends on the distance from $\boldsymbol{p}_i$ to the previous position of the targets. In particular, if $\boldsymbol{p}_i$ is close to $\boldsymbol{x}_k^{(t-1)}$, then the chance that the kth target occupies this position in the current frame should be high. We define:

$$p(\ell_i = k|\boldsymbol{X}^{(t-1)}) \propto \begin{cases} g(\boldsymbol{p}_i, \boldsymbol{x}_k^{(t-1)}) & \text{if } 1 \le k \le M \\ c & \text{if } k = 0, \end{cases} \quad (1)$$

where c is the prior of the background class, and $g(\boldsymbol{p}_i, \boldsymbol{x}_k^{(t-1)})$ is a function decreasing with the distance from $\boldsymbol{p}_i$ to $\boldsymbol{x}_k^{(t-1)}$. We use:

$$g(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} 1 & \text{if } |\boldsymbol{x} - \boldsymbol{y}| < r \\ 0 & \text{otherwise,} \end{cases} \quad (2)$$

where r is a predefined threshold representing the maximal displacement of a target between two successive frames. As result, if the distance from $\boldsymbol{p}_i$ to $\boldsymbol{x}_k^{(t-1)}$ exceeds r, $p(\ell_i = k|\boldsymbol{X}^{(t-1)})$ is zero, implying that $\boldsymbol{p}_i$ cannot be the position of the kth target in the current frame, see Fig. 3.
2. *The measurement distribution* $p(\boldsymbol{z}_i^{(t)}|\ell_i)$. The measurement distribution in each class is assumed to be Gaussian. The background distribution at each location is represented by an isotropic Gaussian learned a priori. A priori learning is possible as the camera is fixed. For the target distribution, we employ the probabilistic PCA model [23], a nonisotropic model which provides more flexibility in modeling appearance changes. Unlike the background, it is usually impossible to learn a target distribution a priori. Section 5 presents a method for the online construction of this distribution from the tracking results, requiring initialization of the target in the first frame only.
3. *The position distribution* $p(\boldsymbol{x}_k^{(t)}|\ell_1, \ldots, \ell_N)$. In the absence of any a priori bias, the probability of the kth target is uniformly distributed over the positions with the label k:

$$p(\boldsymbol{x}_k^{(t)} = \boldsymbol{p}_i|\ell_1, \ldots, \ell_N) = \frac{\delta(\ell_i - k)}{\sum_{j=1}^{N} \delta(\ell_j - k)}, \quad (3)$$

where $\delta()$ denotes the Dirac delta function. Thus, the target will have zero probability at pixels where the class label is different from k.

4.2 State Inference and Target Search

We search for the kth target by maximizing the posterior probability of the position $\boldsymbol{x}_k^{(t)}$ over all pixel sites. The probability is conditioned on the previous states and the current measurements:

$$\hat{\boldsymbol{x}}_k^{(t)} = \arg\max_{\boldsymbol{p}_i} \; p\Big(\boldsymbol{x}_k^{(t)} = \boldsymbol{p}_i|\boldsymbol{Z}^{(t)}, \hat{\boldsymbol{X}}^{(t-1)}\Big), \quad (4)$$

where $\hat{\boldsymbol{x}}_k^{(t)}$ is the estimate of $\boldsymbol{x}_k^{(t)}$, and $\hat{\boldsymbol{X}}^{(t-1)}$ is the estimate of the previous positions of all targets.

The posterior probability $p(\boldsymbol{x}_k^{(t)} = \boldsymbol{p}_i|\boldsymbol{Z}^{(t)}, \boldsymbol{X}^{(t-1)})$ can be inferred using the conditional independence of $\boldsymbol{X}^{(t)}$ from $\boldsymbol{X}^{(t-1)}$ and $\boldsymbol{Z}^{(t)}$ given the labels $\ell_1, \ldots, \ell_N$, as follows:

$$\begin{aligned} & p\Big(\boldsymbol{x}_k^{(t)} = \boldsymbol{p}_i|\boldsymbol{Z}^{(t)}, \boldsymbol{X}^{(t-1)}\Big) \\ &= \sum_{\ell_1=0}^{M} \cdots \sum_{\ell_N=0}^{M} p\Big(\boldsymbol{x}_k^{(t)} = \boldsymbol{p}_i, \ell_1, \ldots, \ell_N|\boldsymbol{Z}^{(t)}, \boldsymbol{X}^{(t-1)}\Big) \\ &= \sum_{\ell_1=0}^{M} \cdots \sum_{\ell_N=0}^{M} p\Big(\boldsymbol{x}_k^{(t)} = \boldsymbol{p}_i|\ell_1, \ldots, \ell_N\Big) \prod_{j=1}^{N} p\Big(\ell_j|\boldsymbol{z}_j^{(t)}, \boldsymbol{X}^{(t-1)}\Big). \end{aligned} \quad (5)$$

Substituting (3) into (5), we can represent the distribution of target position via the distribution of pixel labels. Moreover, the summation over ℓ_i is simplified as:

$$\begin{aligned} & p\Big(\boldsymbol{x}_k^{(t)} = \boldsymbol{p}_i|\boldsymbol{Z}^{(t)}, \boldsymbol{X}^{(t-1)}\Big) = p\Big(\ell_i = k|\boldsymbol{z}_i^{(t)}, \boldsymbol{X}^{(t-1)}\Big) \\ & \sum_{\ell_1=0}^{M} \cdots \sum_{\ell_{i-1}=0}^{M} \sum_{\ell_{i+1}=0}^{M} \cdots \sum_{\ell_N=0}^{M} \left\{ \frac{\prod_{j=1, j\neq i}^{N} p\Big(\ell_j|\boldsymbol{z}_j^{(t)}, \boldsymbol{X}^{(t-1)}\Big)}{1 + \sum_{j=1, j\neq i}^{N} \delta(\ell_j - k)} \right\}. \end{aligned} \quad (6)$$

The direct computation of this probability is intractable since it depends on the distribution of all labels in the field. Fortunately, the maximization of the probability in (6) can be done rather sufficiently using the following proposition:

Proposition 1. *The probability of the position of a target in (6) is monotonically increasing with the probability of the corresponding class. Specifically, for any pair of pixel sites* $\boldsymbol{p}_i$ *and* $\boldsymbol{p}_{i'}$*, the inequality*

$$p\Big(\boldsymbol{x}_k^{(t)} = \boldsymbol{p}_i|\boldsymbol{Z}^{(t)}, \boldsymbol{X}^{(t-1)}\Big) > p\Big(\boldsymbol{x}_k^{(t)} = \boldsymbol{p}_{i'}|\boldsymbol{Z}^{(t)}, \boldsymbol{X}^{(t-1)}\Big)$$

holds if and only if

$$p\Big(\ell_i = k|\boldsymbol{z}_i^{(t)}, \boldsymbol{X}^{(t-1)}\Big) > p\Big(\ell_{i'} = k|\boldsymbol{z}_{i'}^{(t)}, \boldsymbol{X}^{(t-1)}\Big).$$

The proof is given in the Appendix, which can be found at http://computer.org/tpami/archives.htm. It follows from the proposition that the maximization of the probability of the position of a target can be achieved by maximizing the probability of the corresponding class label:

$$\hat{\boldsymbol{x}}_k^{(t)} = \arg\max_{\boldsymbol{p}_i} \; p\Big(\ell_i = k|\boldsymbol{z}_i^{(t)}, \boldsymbol{X}^{(t-1)}\Big). \quad (7)$$

The intuitive explantation of this equation is that the image part that is best recognized as a target is the most likely position of the target. The analogy between the label distribution and the target position distribution is also confirmed by the simulated example shown in Fig. 4.

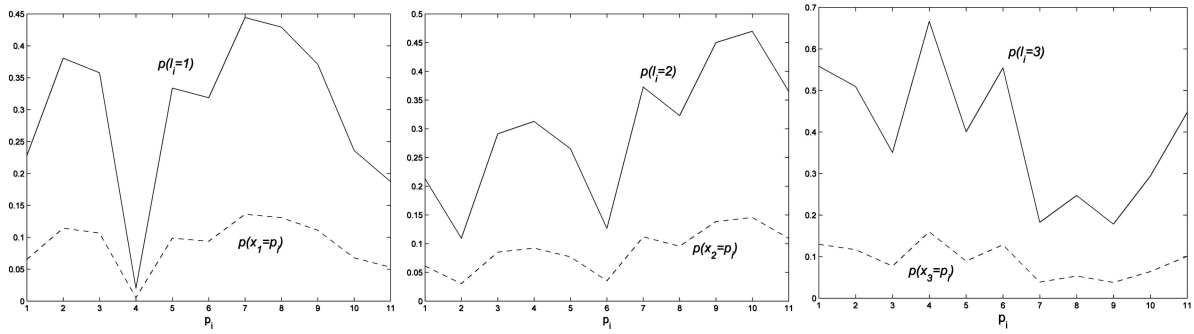


Fig. 4. The illustration of the analogy between the label probability and the target location probability. In this example, label distributions are randomly generated for an 11 pixels field with three classes. The solid lines represent the label probability while the dashed lines represent the true target position probability calculated by (6).

The probability of a class label is calculated using the Bayes formula as follows:

$$\begin{aligned} &p\Big(\ell_i = k|\boldsymbol{z}_i, \boldsymbol{X}^{(t-1)}\Big) = p(\boldsymbol{z}_i|\ell_i = k) \\ &p\Big(\ell_i = k|\boldsymbol{X}^{(t-1)}\Big)/p\Big(\boldsymbol{z}_i|\boldsymbol{X}^{(t-1)}\Big) \\ &\quad = \frac{p(\boldsymbol{z}_i|\ell_i = k)p\big(\ell_i = k|\boldsymbol{X}^{(t-1)}\big)}{\sum_{k=0}^{M} p(\boldsymbol{z}_i|\ell_i = k)p\big(\ell_i = k|\boldsymbol{X}^{(t-1)}\big)}. \end{aligned} \tag{8}$$

Substituting (1) into (8), the equation of the target search is elaborated as:

$$\hat{\boldsymbol{x}}_k^{(t)} = \arg\max_{\boldsymbol{p}_i} \frac{p(\boldsymbol{z}_i|\ell_i = k)g\Big(\boldsymbol{p}_i, \boldsymbol{x}_k^{(t-1)}\Big)}{c\ p(\boldsymbol{z}_i|\ell_i = 0) + \sum_{k'=1}^{M} p(\boldsymbol{z}_i|\ell_i = k')g\Big(\boldsymbol{p}_i, \boldsymbol{x}_{k'}^{(t-1)}\Big)}. \tag{9}$$

As observed in (9), while the numerator contains the likelihood of one target $p(\boldsymbol{z}_i^{(t)}|\ell_i = k)$, the denominator contains the likelihood of the background $p(\boldsymbol{z}_i^{(t)}|\ell_i = 0)$ and the likelihood of the other targets $p(\boldsymbol{z}_i^{(t)}|\ell_i = k')$. As a result, the tracker not only searches for the target k, but also avoids latching on the other targets or a background region. This is the major difference between the proposed method and the other methods which basically maximize the likelihood of individual targets.

There is no need to consider all targets while calculating (9). The weight $g(\boldsymbol{p}_i, \boldsymbol{x}_{k'}^{(t-1)})$ restricts the consideration in the neighborhood of $\boldsymbol{p}_i$. In particular, if the target is distant from the other targets, the algorithm needs to compute only the target likelihood and the background likelihood, and then maximize their ratio.

The estimation of the posterior class probability $p(\ell_i = k|\boldsymbol{z}_i, \boldsymbol{X}^{(t-1)})$, which is used for the subsequent maximization of the position probability $p(\boldsymbol{x}_k^{(t)} = \boldsymbol{p}_i|\boldsymbol{Z}^{(t)}, \boldsymbol{X}^{(t-1)})$, resembles the EM framework. The major difference with the standard EM is that the posterior class probability does not depend on the current position parameters, and moreover, the maximization of the probability of the position parameters can be done efficiently through the maximization of the class probability. As result, no iterative algorithm is needed and only one EM step is taken here.

Note that, for the proposed model, the computation of the state probability conditioned on the entire history of the observations $p(\boldsymbol{x}_k^{(t)} = \boldsymbol{p}_i|\boldsymbol{Z}^{(1:t)})$ is intractable due to the computational complexity of the probability in (6). In view of this, (4) is also a reasonable approach to locate the target. This approach works effectively in most tracking tasks and has been common in tracking [22], [27].

5 Online Construction of the Measurement Distribution Using the Incremental Probabilistic PCA

We now address the inclusion of the temporal context in the measurement model for each target.

The distribution of the measurement of the target k is represented by a Gaussian with the mean vector $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{C}_k$:

$$p\Big(\boldsymbol{z}_i^{(t)}|\ell_i = k\Big) = \mathcal{N}\Big(\boldsymbol{z}_i^{(t)}; \boldsymbol{\mu}_k, \boldsymbol{C}_k\Big). \tag{10}$$

The definition of measurement can be different among targets, depending on the target size. Each target is represented by a rectangular patch in the image. For the kth target, the measurement vector $\boldsymbol{z}_i$ is composed of the intensity values of the image patch which has the same size as the target and is centered at $\boldsymbol{p}_i$. The background likelihood at a pixel is evaluated by applying the model for the image patch centered at the pixel and having the size equal to the average of the size of the target windows. The assumption of the rectangular shape is not strict. Depending on the application, another shape such as ellipse can also be used if it better represents the target region.

The ability in representing complex data structures depends on the specifics of $\boldsymbol{C}_k$. The most simple model is the isotropic Gaussian, where $\boldsymbol{C}_k = \sigma_k^2\boldsymbol{I}$ and $\boldsymbol{I}$ is the identity matrix. This mode can only represent one snap shot of the object without any appearance variations. The full non-isotropic Gaussian with no constraint between the elements of $\boldsymbol{C}_k$ is most powerful but not computationally tractable when the dimensionality of the data is high. The common trade-off is the probabilistic PCA (PPCA) model [23]:

$$\boldsymbol{C}_k = \sigma_k^2\boldsymbol{I} + \boldsymbol{W}_k\boldsymbol{W}_k^T. \tag{11}$$

$\boldsymbol{W}_k$ is a $d_k \times q_k$ matrix, d_k is the dimensionality of $\boldsymbol{z}_i$, and $q_k \ll d_k$. This model provides a good balance between the representation accuracy and the complexity. In fact, the

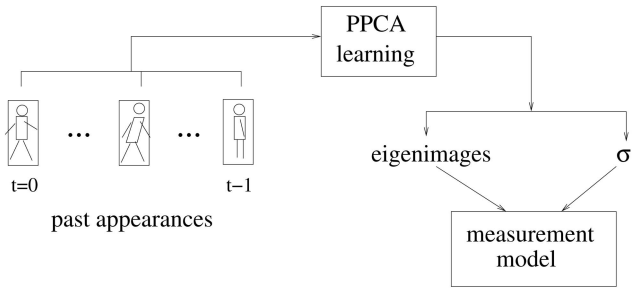


Fig. 5. The measurement distribution is built online by incrementally learning PPCA from the target's past appearances.

hyperplane spanned by the columns of W_k is the same hyperplane spanned by the first q_k eigenvectors of the covariance matrix. So, the model is rather similar to the classical eigenspace model, but has the advantage of the probabilistic interpretation.

In the presented method, PPCA for each target is estimated from the history of the past measurements $\boldsymbol{z}^{(1,k)}, \ldots, \boldsymbol{z}^{(t,k)}$ which are obtained from the beginning to frame t, see Fig. 5. Here, $\boldsymbol{z}^{(t,k)}$ is the vector of intensities of the image region at the estimated location of the kth target in frame t.

5.1 Maximum Likelihood Solution of Probabilistic PCA

According to [23], the maximum likelihood estimation of PPCA is:

$$\boldsymbol{\mu}_k = \frac{1}{t}\sum_{i=1}^{t} \boldsymbol{z}^{(i,k)}, \tag{12}$$

$$\sigma_k^2 = \frac{1}{d_k - q_k}\sum_{i=q_k+1}^{d_k} \lambda_{i,k}, \tag{13}$$

$$\boldsymbol{W}_k = \boldsymbol{V}_{q,k}(\boldsymbol{\Lambda}_{q,k} - \sigma_k^2 \boldsymbol{I})^{1/2}\boldsymbol{R}, \tag{14}$$

where $\lambda_{1,k}, \lambda_{2,k}, \ldots, \lambda_{d,k}$ are the eigenvalues arranged in the descending order of the observation covariance matrix:

$$\boldsymbol{S}_k = \frac{1}{t}\sum_{i=1}^{t}\left[\boldsymbol{z}^{(i,k)} - \boldsymbol{\mu}_k\right]\left[\boldsymbol{z}^{(i,k)} - \boldsymbol{\mu}_k\right]^T. \tag{15}$$

Let $\boldsymbol{v}_{1,k}, \ldots, \boldsymbol{v}_{d,k}$ be the corresponding eigenvectors. Here, $\boldsymbol{V}_{q,k}$ is the $d_k \times q_k$ matrix whose columns are $\boldsymbol{v}_{1,k}, \ldots, \boldsymbol{v}_{q,k}$, $\boldsymbol{\Lambda}_{q,k}$ is the diagonal matrix whose diagonal elements are $\lambda_{1,k}, \ldots, \lambda_{q,k}$, and $\boldsymbol{R}$ is an arbitrary $q_k \times q_k$ orthogonal matrix.

The estimated covariance matrix is:

$$\begin{aligned}\boldsymbol{C}_k &= \sigma_k^2 \boldsymbol{I} + \sum_{i=1}^{q_k}(\lambda_{i,k} - \sigma_k^2)\boldsymbol{v}_{i,k}\boldsymbol{v}_{i,k}^T \\ &= \sum_{i=1}^{q_k}\lambda_{i,k}\boldsymbol{v}_{i,k}\boldsymbol{v}_{i,k}^T + \sigma_k^2\sum_{i=q_k+1}^{d_k}\boldsymbol{v}_{i,k}\boldsymbol{v}_{i,k}^T.\end{aligned} \tag{16}$$

While $\lambda_{1,k}, \ldots, \lambda_{q,k}$ are the variances of the first q principal components, σ_k^2 is the average of the variances of the remaining $d_k - q_k$ components.

Note that (12), (13), and (14) should be used only in a batch mode, where all $\boldsymbol{z}^{(i,k)}, 1 \le i \le t$ are stored in memory and, in addition, when the data dimensionality d is low. The next section will present an efficient method for the estimation of the high-dimensional PPCA in the incremental mode without requiring the storage of all the past measurements.

5.2 Incremental Probabilistic PCA

In the incremental mode, the parameters are updated using the current parameters for each target k individually and the new coming measurement $\boldsymbol{z}^{(t+1,k)}$ for that target. In the sequel, we drop index k as this section holds for all targets. The full set of parameters of a target includes the mean vector $\boldsymbol{\mu}$, the first q eigenvectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_q$, the corresponding eigenvalues $\lambda_1, \ldots, \lambda_q$, and the noise parameter σ^2. Like before, we use the superscript (t) to denote the estimation of these parameters obtained at time t.

Upon the arrival of a new measurement $\boldsymbol{z}^{(t+1)}$, the mean vector is easily updated as:

$$\begin{aligned}\boldsymbol{\mu}^{(t+1)} &= \frac{1}{t+1}\sum_{i=1}^{t+1}\boldsymbol{z}^{(i)} \\ &= \frac{t}{t+1}\boldsymbol{\mu}^{(t)} + \frac{1}{t+1}\boldsymbol{z}^{(t+1)}.\end{aligned} \tag{17}$$

The new observation covariance matrix is:

$$\begin{aligned}\boldsymbol{S}^{(t+1)} &= \frac{1}{t+1}\sum_{i=1}^{t+1}\left[\boldsymbol{z}^{(i)} - \boldsymbol{\mu}^{(t+1)}\right]\left[\boldsymbol{z}^{(i)} - \boldsymbol{\mu}^{(t+1)}\right]^T \\ &= \frac{t}{t+1}\boldsymbol{S}^{(t)} + \frac{t}{t+1}\boldsymbol{y}\boldsymbol{y}^T,\end{aligned} \tag{18}$$

where $\boldsymbol{y} = \sqrt{\frac{1}{t+1}}[\boldsymbol{z}^{(t+1)} - \boldsymbol{\mu}^{(t)}]$.

We need to calculate the eigenvectors and the eigenvalues of $\boldsymbol{S}^{(t+1)}$ in order to obtain the new estimation of the parameters. The direct eigenvalue decomposition of $\boldsymbol{S}^{(t+1)}$ is impossible due to the high value of d.

The crucial point is to approximate $\boldsymbol{S}^{(t)}$ by its current estimation given in (16), yielding:

$$\boldsymbol{S}^{(t+1)} \approx \frac{t}{t+1}\left[\sigma^{(t)^2}\boldsymbol{I} + \sum_{i=1}^{q}(\lambda_i^{(t)} - \sigma^{(t)^2})\boldsymbol{v}_i^{(t)}\boldsymbol{v}_i^{(t)T} + \boldsymbol{y}\boldsymbol{y}^T\right]. \tag{19}$$

We remark that in related methods [12], [6], [3], matrix $\boldsymbol{S}^{(t)}$ is traditionally approximated as $\boldsymbol{S}^{(t)} = \sum_{i=1}^{q}\lambda_i\boldsymbol{v}_i^{(t)}\boldsymbol{v}_i^{(t)T}$. This approximation is less accurate than (16) since it completely removes the variances of the last $d - q$ principal components. Furthermore, it does not include σ. Therefore, they do not allow updating this parameter.

Let

$$\boldsymbol{L} = \left[\sqrt{\lambda_1^{(t)} - \sigma^{(t)^2}}\boldsymbol{v}_1^{(t)}, \ldots, \sqrt{\lambda_q^{(t)} - \sigma^{(t)^2}}\boldsymbol{v}_q^{(t)}, \boldsymbol{y}\right]. \tag{20}$$

Then, (19) becomes:

$$\boldsymbol{S}^{(t+1)} \approx \frac{t}{t+1}\left[\sigma^{(t)^2}\boldsymbol{I} + \boldsymbol{L}\boldsymbol{L}^T\right]. \tag{21}$$

From here, to obtain the eigenvectors and eigenvalues of $\boldsymbol{S}^{(t+1)}$, we need only the eigenvalue decomposition of the matrix $\boldsymbol{L}\boldsymbol{L}^T$. Again, the decomposition should not be applied directly to $\boldsymbol{L}\boldsymbol{L}^T$, which is $d \times d$.

Instead, we set the $(q+1)\times(q+1)$ matrix:

$$\boldsymbol{Q} = \boldsymbol{L}^T\boldsymbol{L} = \begin{pmatrix} \boldsymbol{\Sigma} & \boldsymbol{\beta} \\ \boldsymbol{\beta}^T & \alpha \end{pmatrix}, \tag{22}$$

where $\boldsymbol{\Sigma} = \mathrm{diag}\{\lambda_1^{(t)} - \sigma^{(t)^2}, \ldots, \lambda_q^{(t)} - \sigma^{(t)^2}\}$, $\alpha = \boldsymbol{y}^T\boldsymbol{y}$, and $\boldsymbol{\beta}$ is the $q\times 1$ vector whose elements are

$$\beta_i = \sqrt{\lambda_i^{(t)} - \sigma^{(t)^2}}\,\boldsymbol{v}_i^{(t)\,T}\boldsymbol{y}.$$

Let the eigenvalue decomposition of Q be:

$$\boldsymbol{Q} = \boldsymbol{U}\boldsymbol{\Gamma}\boldsymbol{U}^T, \tag{23}$$

where $\boldsymbol{\Gamma} = \mathrm{diag}\{\gamma_1, \ldots, \gamma_{q+1}\}$, and $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$. The eigenvectors of LL^T are the columns of the matrix:

$$\boldsymbol{V} = \boldsymbol{L}\boldsymbol{U}\boldsymbol{\Gamma}^{-1/2}. \tag{24}$$

Let $\boldsymbol{V} = [\boldsymbol{v}_1^{(t+1)}, \ldots, \boldsymbol{v}_{q+1}^{(t+1)}]$. Equation (19) is rewritten as:

$$\boldsymbol{S}^{(t+1)} \approx \frac{t}{t+1}\left[\sigma^{(t)^2}\boldsymbol{I} + \sum_{i=1}^{q+1}\gamma_i\boldsymbol{v}_i^{(t+1)}\boldsymbol{v}_i^{(t+1)\,T}\right]. \tag{25}$$

It follows that $\boldsymbol{v}_1^{(t+1)}, \ldots, \boldsymbol{v}_{q+1}^{(t+1)}$ are the first $q+1$ eigenvectors of $\boldsymbol{S}^{(t+1)}$. Only the first q eigenvectors are retained in memory. The first $q+1$ eigenvalues of $\boldsymbol{S}^{(t+1)}$ are:

$$\lambda_i^{(t+1)} = \frac{t}{t+1}[\sigma^{(t)^2} + \gamma_i]. \tag{26}$$

The $d-q-1$ remaining eigenvalues have the same value $\frac{t}{t+1}\sigma^{(t)^2}$. Using (13), σ is updated as:

$$\begin{aligned}\sigma^{(t+1)^2} &= \frac{1}{d-q}\left[\lambda_{q+1}^{(t+1)} + (d-q-1)\frac{t}{t+1}\sigma^{(t)^2}\right] \\ &= \frac{t}{t+1}\left[\frac{\gamma_{q+1}}{d-q} + \sigma^{(t)^2}\right].\end{aligned} \tag{27}$$

The incremental PPCA is summarized as follows for each target:

1. Update the mean $\boldsymbol{\mu}$, (17).
2. Update the matrix $\boldsymbol{W}$.
   a. Set up matrix $\boldsymbol{Q}$, (22), and decompose it in its eigenvectors and eigenvalues by (23).
   b. Then, compute the matrix $\boldsymbol{V}$ by (24). The first q columns of $\boldsymbol{V}$ are the new eigenvectors.
   c. The corresponding eigenvalues λ_i are calculated by (26). This yields all ingredients to compute (14).
3. Update the noise parameter, (27).

The initial PPCA model is learned from an initial set of measurements $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_k$ using the batch mode algorithm [23]. Note that we should have $k > q$, otherwise the estimated covariance matrix would be singular. The incremental PPCA can then start from frame $k+1$.

The most computationally consuming part of the proposed algorithm is Steps 2a and 2b. The complexity of the decomposition of matrix $\boldsymbol{Q}$ in Step 2a is $O(q^3)$. The complexity of the matrix multiplication in Step 2b is $O(dq)$. So, the overall complexity is $O(q^3) + O(dq)$ per each update. Since q is small, the algorithmic complexity is linear with the dimensionality of data. Therefore, the algorithm is efficient.

6 Experiments

We have performed experiments to evaluate the performance of the proposed appearance model and the tracking algorithm in Section 4.

6.1 Performance Evaluation of the Proposed Appearance Model

This first set of experiments demonstrates the accuracy of the proposed incremental PPCA algorithm and its applicability in the modeling appearance of a target.

To test the accuracy, we applied the algorithm for a synthetic data stream that was sequentially drawn from a Gaussian distribution of the form (11) with dimensionality $d = 300$, and number of principal components $q = 5$. The result is compared with the true distribution and the results of batch PPCA and the method of Lin et al. in [14]. We have implemented the last method for the case where the updating of PPCA parameters is performed upon the arrival of each new sample. Specifically, the new set of eigenvectors and eigenvalues are obtained by computing the Singular Value Decomposition (SVD) of the matrix:

$$E = \left[\sqrt{\gamma}\boldsymbol{V}_q^{(t)}\boldsymbol{\Lambda}_q^{(t)} | \sqrt{(1-\gamma)\gamma}\left(\boldsymbol{z}^{(t+1)} - \boldsymbol{\mu}^{(t)}\right)\right], \tag{28}$$

where $\boldsymbol{V}_q^{(t)}$ is the matrix whose columns are the q eigenvectors calculated at time t, $\boldsymbol{\Lambda}_q^{(t)}$ is the diagonal matrix created from the eigenvalues, and γ is a forgetting factor. We have set $\gamma = \frac{t}{t+1}$, which means that a newly arrived sample has the same weight as past samples. The result of SVD will have the form $E = \boldsymbol{V}_{q+1}^{(t+1)}\boldsymbol{\Lambda}_{q+1}^{(t+1)}\boldsymbol{U}_{q+1}^{(t+1)\,T}$. The q largest diagonal elements of $\boldsymbol{\Lambda}_{q+1}^{(t+1)}$ are the updated eigenvalues, and the corresponding columns in $\boldsymbol{V}_{q+1}^{(t)}$ are the updated eigenvectors. The noise variance is determined heuristically as fraction of the smallest eigenvalue:

$$\sigma^{(t)^2} = \kappa\lambda_q^{(t)}, \tag{29}$$

where κ is a predefined coefficient. Since the selection of optimal κ is not trivial, in the experiment we have used different values of κ, resulting in different estimates.

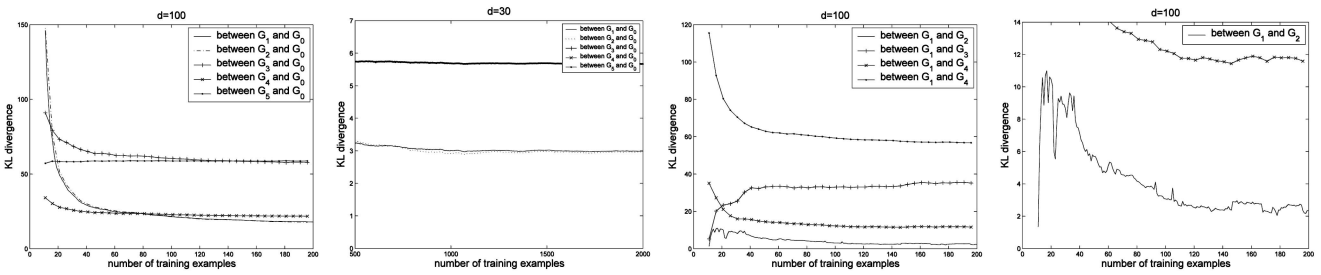
Every time a new sample arrives, the following Gaussians are compared:

1. G_0: the ground truth Gaussian distribution,
2. G_1: the Gaussian, updated by incremental PPCA,
3. G_2: the Gaussian, estimated by batch-mode PPCA for all drawn data points.
4. G_3, G_4, G_5: the Gaussians, estimated by the method in [14] with κ set to 0.05, 0.1, and 0.3, respectively.

The similarity measure between two Gaussians $\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{C}_1)$ and $\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{C}_2)$ is computed using the Kullback-Leibler divergence as:

$$\frac{1}{2}\left[\log\frac{|C_2|}{|C_1|} + \mathrm{tr}(C_2^{-1}C_1) + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T C_2^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) - d\right]. \tag{30}$$

The divergence between each pair of Gaussians are calculated and plotted as function of the number of training examples. Fig. 6a shows the divergence plots between G_0 and G_1, G_2, G_3, G_4, and G_5, respectively. Incremental PPCA



(a) (b) (c) (d)

Fig. 6. The Kullback-Leibler divergence between G_0: the ground truth Gaussian distribution, G_1: the Gaussian estimated by incremental PPCA, G_2: the Gaussian estimated by batch PPCA, and G_3, G_4, G_5: the Gaussians estimated by the method of [14] with $\kappa = 0.05, 0.1, 0.3$, respectively.

and batch PPCA have an equal accuracy in terms of the convergence to the ground truth distribution. The results of the method in [14] are sensitive to the setting of κ, showing the major problem of this ad hoc approach. In addition, G_3, G_4, and G_5 gain very little improvement in KL divergence as the number of samples increases and eventually have higher errors than G_1 and G_2. It can also be noted that, for small numbers of samples, the maximum likelihood estimation by batch PPCA is far from the best. This is because the maximum likelihood does not necessarily minimize the KL divergence from the true distribution. Especially, for small data sets, the error can be large due to overfitting. To verify, we repeated the experiment for lower dimensionality $d = 30$ and a larger number of samples. As shown in Fig. 6b, for $n > 1,000$, batch PPCA exhibits the best performance compared although the difference from incremental PPCA is small. Fig. 6c shows the divergence between the result of batch PPCA, G_2, and the other Gaussians. As observed, the result our incremental algorithm is most similar to G_2. Furthermore, as Fig. 6d shows, the results of incremental PPCA and batch PPCA become closer as more data arrive.

We have also verified the efficiency of the proposed PPCA scheme by measuring the computation time required by one update. Fig. 7 shows the plot of this time as function of the data dimensionality d. The computation was made in MATLAB and a 1.6 GHz laptop. The figure confirms the linearity of the computation time with respect to the data dimensionality.

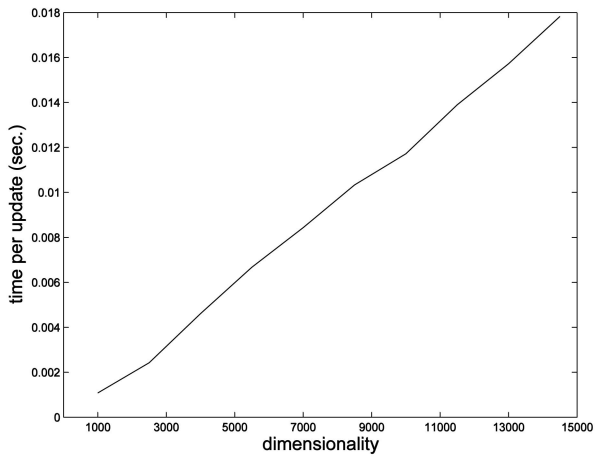


Fig. 7. Computation time of one iteration of PPCA as function of dimensionality. The computations were made in MATLAB and a 1.6 GHz laptop.

6.2 Multitarget Tracking

In this set of experiments, we demonstrate the result of the proposed methods for tracking multiple people.

The tracking is initialized by specifying the position and size of each target in the first frame. Although the subtraction from the background image could be a good source of information for the initialization, it usually includes false alarms or misses some part of the target object due to similarity to the background appearance. For automatic initialization, these errors need be corrected using some prior knowledge. Therefore, to focus on demonstrating the advantage of context information, in the reported experiments, all targets were initialized manually.

The parameters are set as follows: The background prior $c = 0.1$ ensuring that the probability of the background class is low in the vicinity of the targets. The value of c requires that the label distribution of a target is tight around its position. The threshold r indicates the maximum displacement in one time step, so that the algorithm can find the target in the next frame. In addition, to handle occlusion, it should also be larger than the size of the occluding object. This is reasonable a priori knowledge. When set to the maximum value, the image size, the computational burden will slow down the tracker too much. In the experiment, we track people with roughly the same size. r is set as $r = 3 \times$ the average target width. The measurement distribution for each target is represented by a PPCA-model with the first $q = 5$ principal components. The incremental update of PPCA starts after $k = 2q$ frames. Moreover, in the first k frames, targets are tracked independently and simply by intensity matching with the sample given in the first frame.

A target is considered occluded when the likelihood drops too low, namely,

$$\left(\boldsymbol{z}^{(t)} - \boldsymbol{\mu}_k\right)^T \boldsymbol{C}_k^{-1} \left(\boldsymbol{z}^{(t)} - \boldsymbol{\mu}_k\right) < t_o. \tag{31}$$

We have used $t_o = 6.0 d_k$. The inverting of $\boldsymbol{C}_k$ in (31) can be done efficiently using the Woodbury formulae [18]. During occlusion, the appearance model of the target is not updated.

The tracking of a target is stopped either when it is occluded for more than 100 successive frames or it moves outside the image border.

We have tested the algorithm on several video sequences. The results are shown in Figs. 8a, 9a, and 10a, respectively. For comparison, Figs. 8b, 9b, and 10b show the results of

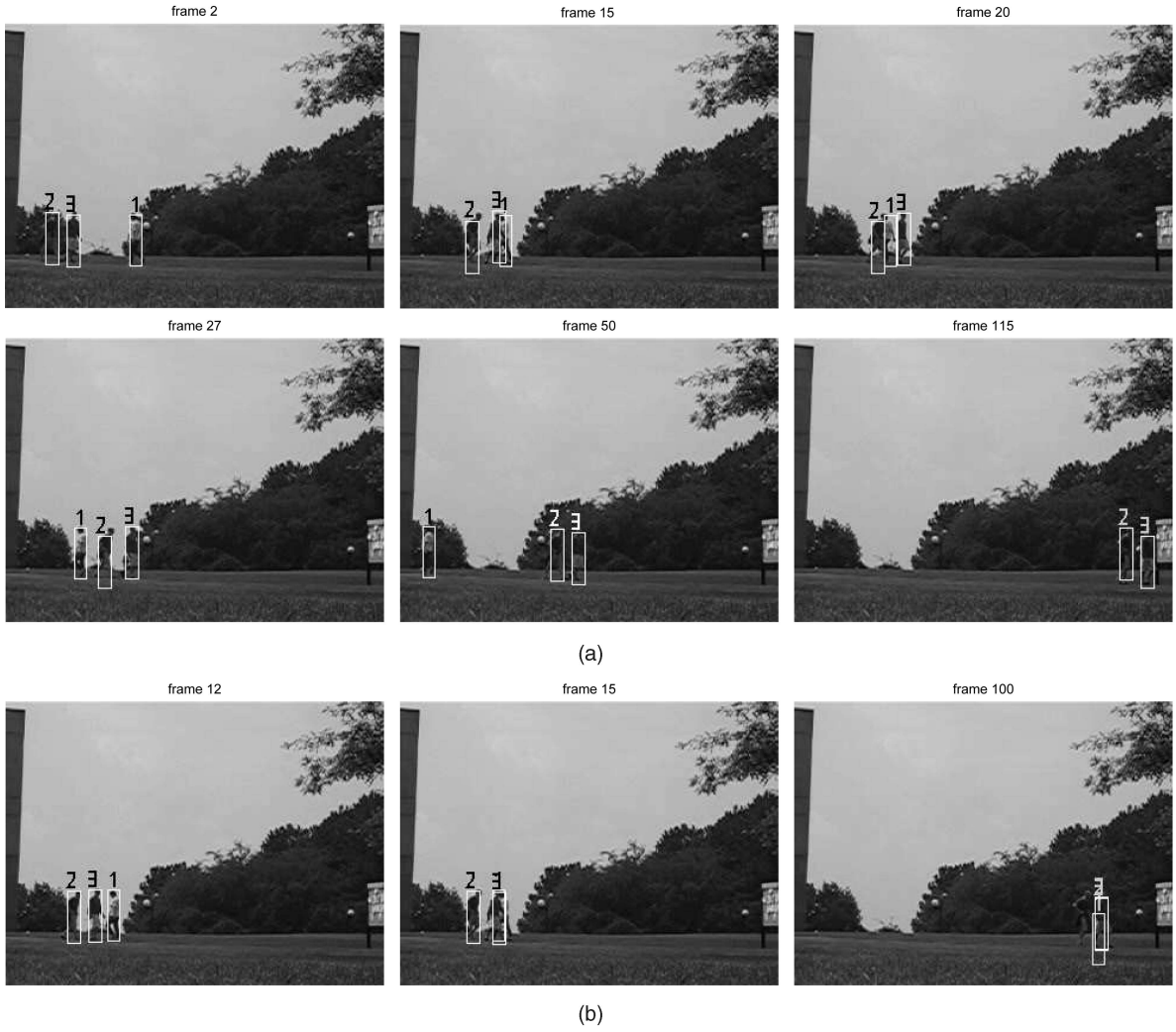


Fig. 8. (a) The result of the proposed algorithm for tracking multiple approaching targets with occlusions. The number on top of each target indicates its label. (b) The results of the single-target trackers.

independently applying multiple instances of a single target tracker. In this method, each target is searched for by maximizing the ratio of its likelihood to the background likelihood.

In Fig. 8, three persons are approaching each other from opposite directions. An occlusion takes place at frames 15-25 when they cross each other. Targets 2 and 3 have a slightly similar appearance and walk at a close distance. The single-target trackers quickly lose track of the first target at the occlusion. At frame 100, the three targets merge into one. The proposed algorithm tracks successfully and maintains the correct identity of the targets over the entire sequence.

Fig. 9 shows a similar situation but with four persons. Again, the proposed algorithm successfully tracks all targets until they leave the scene. The single-target trackers mix the targets even before the occlusion and all the four merge into one at the end, showing the drawback of the approach of maximizing the likelihood of individual targets.

A difficult example is shown in Fig. 10. The sequence was recorded by a fixed camera, located at a high window and looking down on people walking on a street. Due to the distance, all targets appear similar and small. Occlusions occur when people cross or pass behind trees. At some occlusions, three people coincide. In the figure, the proposed algorithm correctly tracks and classifies all targets except for the moment of occlusion when target windows merge. Immediately after, the correct identification of targets is restored. In the result of single-target trackers, the window of targets 1, 2, and 5 melts together at the first occlusion in frame 35. Erroneously, they stick to one target until frame 140. The same thing occurs for targets 3 and 4. The same thing occurs for two other targets. As a consequence, the independent trackers lose track and cannot recover.

The power of incremental PPCA in modeling object appearance is demonstrated in Fig. 11a. The figure shows the result of the proposed algorithm for tracking two faces under severe pose change and occlusion. A complete occlusion occurs in frame 250 when one person passes behind the other. Note that, during occlusion, the first person makes a pose change from frontal view to side view. The online training of a PPCA model for this person has taken into account different views of his head before the occlusion. Therefore, the algorithm successfully recognizes the profile view after occlusion since it has been seen earlier in frame 112. The eigenimages obtained are shown in Fig. 12. They also depict different views of the head. Fig. 11b shows the results

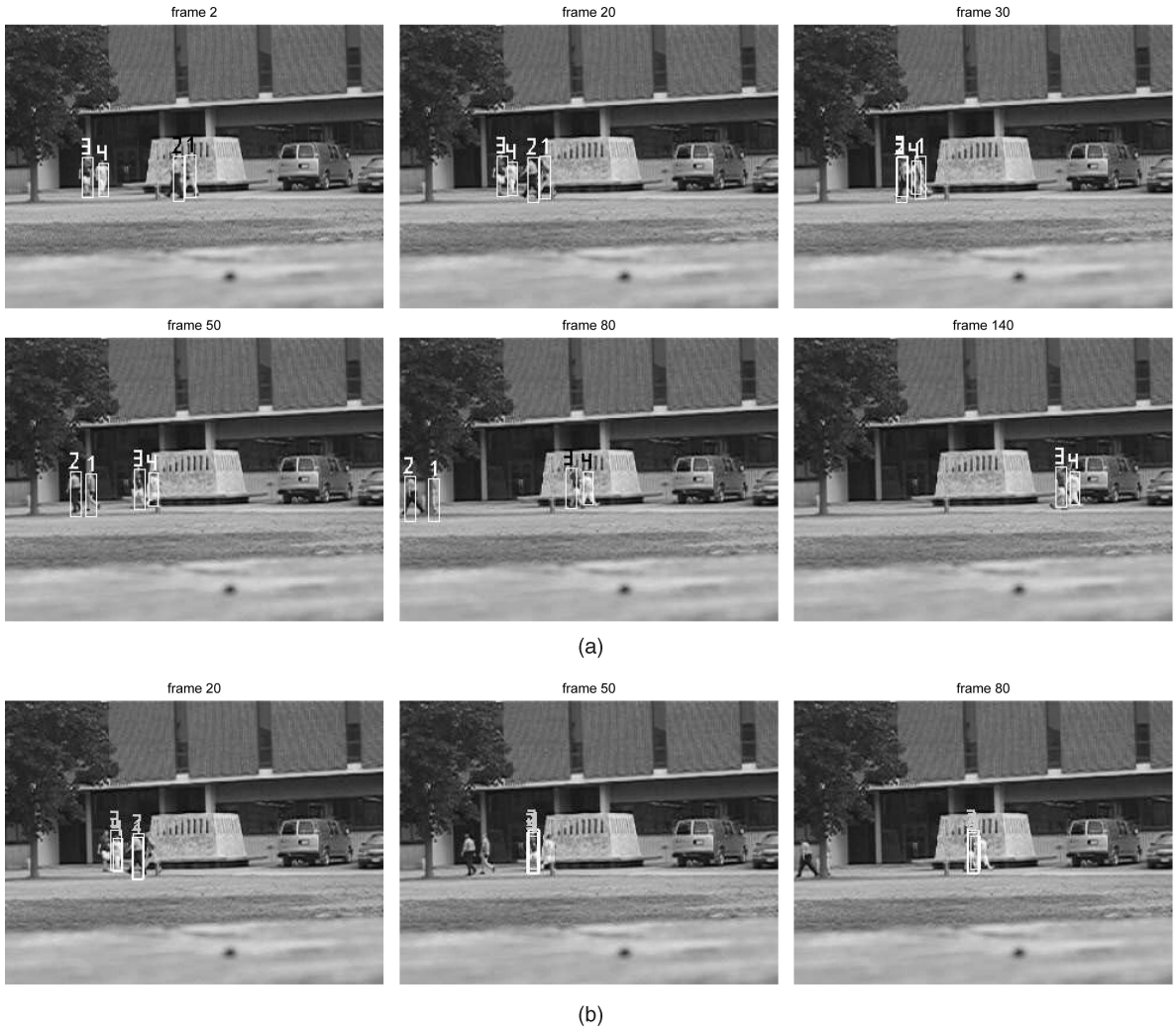


Fig. 9. The result of tracking multiple targets by (a) the proposed algorithm and (b) single-target trackers.

of a modified version of the proposed algorithm where the PPCA model is turned off. In this version, the likelihood model for each target is an isotropic Gaussian with the mean being a template image. The mean and the variance of this Gaussian is updated over time using the Kalman filter [17]. Unlike PPCA, the Kalman filter can represent object appearance over a short period of time only and will forget an object view that was observed some time in the past but is no longer visible. As a consequence, the algorithm failed to recognize the profile view of the person after the occlusion, see Fig. 11b, frame 275.

7 Conclusion and Future Work

A new approach has been proposed for tracking multiple targets, emphasizing the use of the context information. We have shown that the accuracy of the target identification can be improved by the incorporation of information from the spatial and temporal context of each target.

The tracker discriminates a target from nearby targets and the background by the pixel values in the target window. Before searching for the next target position, all targets are classified. Maximization of the probability of the target label, rather than the target likelihood, prevents the target from latching on image regions of the other targets or of the background. As long as the appearance of the targets is distinguishable, separating targets in appearance space and not in position space can effectively overcome the problem of target coalescence and identity switching. Moreover, this can be achieved without imposing ad hoc constraints on the targets' position, preserving purity in the location of the target.

The key element in the representation of target appearance is Probabilistic PCA, incrementally updated online without storage of past measurements. This permits the construction of a robust appearance model for each target. The model effectively represents the diversity in appearances as seen during the track, providing the long-term memory which is instrumental in redetecting an object after occlusions and severe pose changes.

Some issues remain for future research. First, since we discriminate the targets based on their appearance, their identity may not be determined correctly if their appearance is indistinguishable. Specifically for this case, constraints on the position of the target or constraints on the pixel labels might be helpful, although the correct identification of the targets would still not be guaranteed. Second, the current algorithm considers the translational motion model only, as

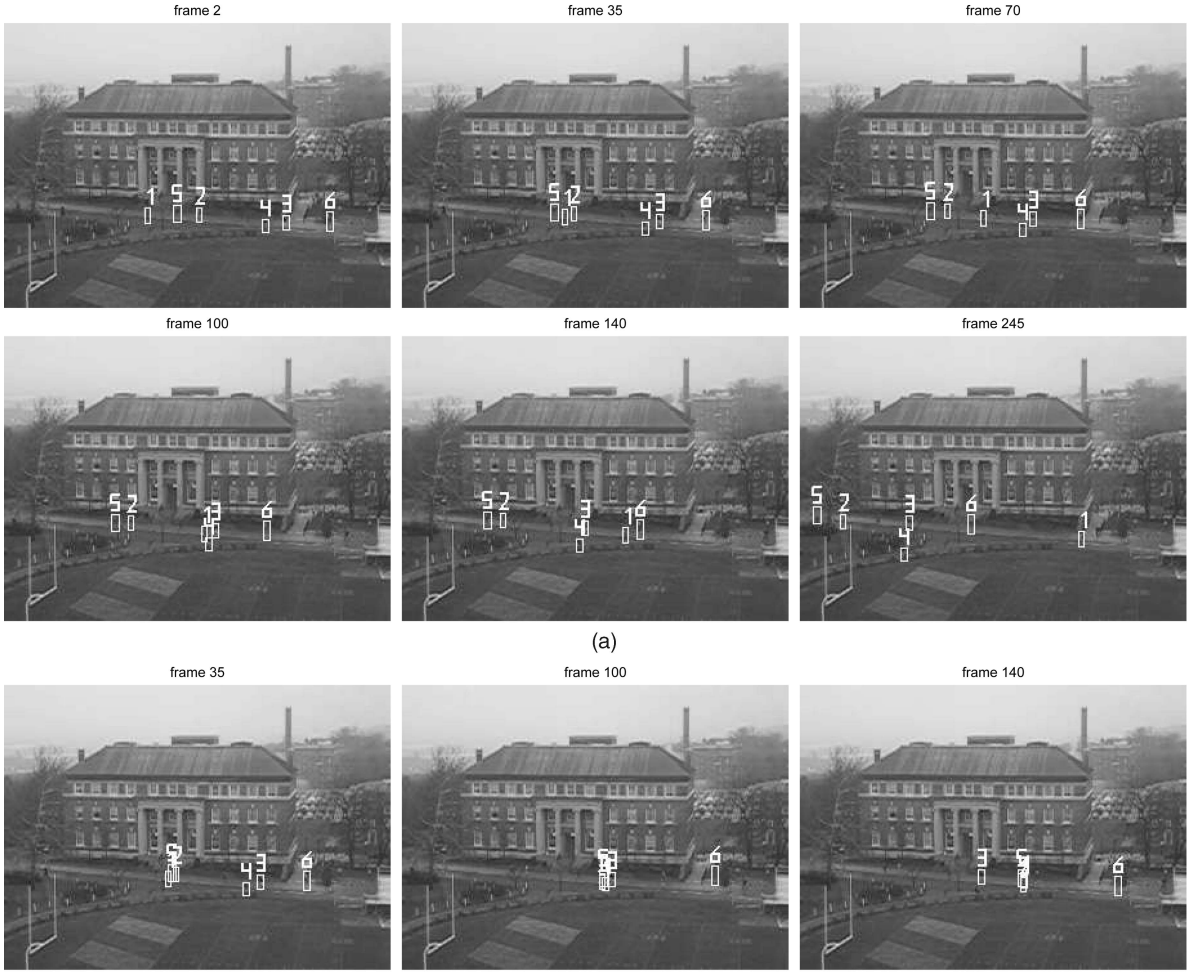


Fig. 10. The results of tracking multiple targets in the conditions of a low resolution and heavy occlusions by (a) the proposed algorithm and (b) single-target trackers.

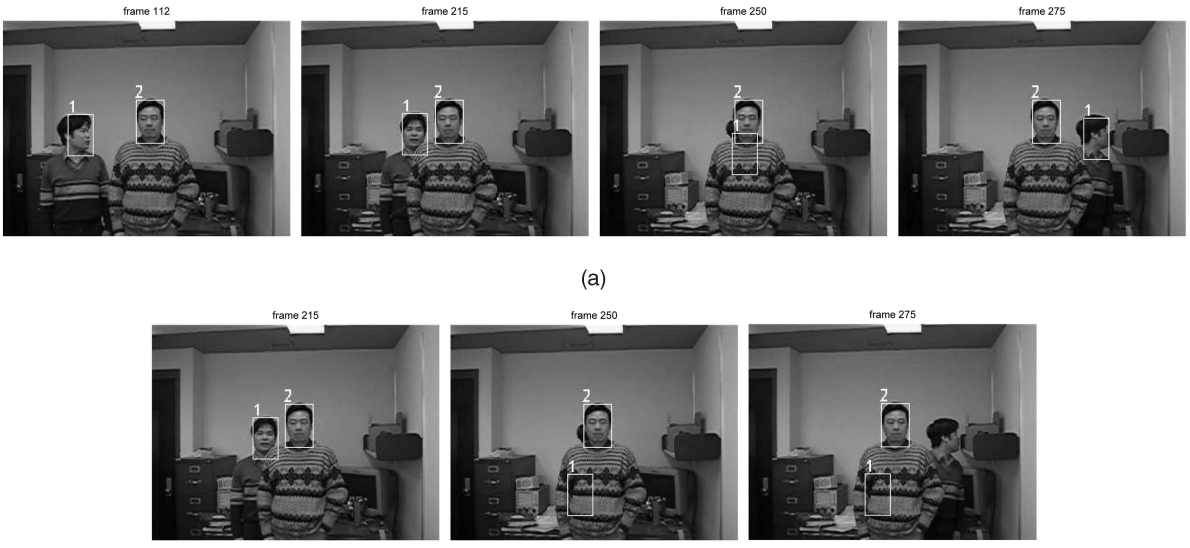


Fig. 11. (a) Tracking results of the proposed algorithm in the condition of occlusion and pose change. (b) The result of the algorithm that updates the template using the Kalman filter.

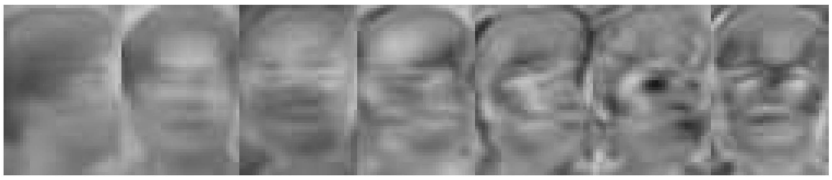


Fig. 12. The eigenimages obtained from the PPCA.

the maximization of the objective function is performed over the pixel sites. This limits the accuracy of tracking rotational and scaling motion. The extension of the algorithm for more sophisticated motion models is to be based on the definition of a new label field in a higher dimensional space of motion parameters and the corresponding extension of the likelihood model and the motion prediction model in (9) for that space. Third, while the current algorithm does allow targets to have different sizes, the normalization for targets having significantly different sizes can be a problem. The algorithm may be more sensitive to inaccuracies in the target likelihood model when the likelihood is not a Gaussian. Finally, online learning of the target model may have problems with incorporating incorrect data like other adaptive trackers. When the target window drifts or loses the object, the appearance of the background or of the other targets will be incorporated into the PPCA model of the target. When the tracking errors occur over a short period, the problem with incorrect data is not serious, since the PPCA model still remembers the original appearance of the object well. When the errors persist for a long time, incorrect data will become the most major principal components and the model will be damaged. This problem, fundamental to all adaptive trackers, can only be solved by a proper combination between a general a priori object model that is learned offline and the object model that is learned online.

ACKNOWLEDGMENTS

The authors are grateful to the anonymous reviewers for thoroughly reading and making very valuable comments and suggestions that have significantly improved the quality and accuracy of the paper. The research described in this paper was supported in part by a grant (N41756-03-C-4028) to the Rensselaer Polytechnic Institute from the Task Support Working Group (TSWG) of the United States.

REFERENCES

[1] S. Avidan, "Support Vector Tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 8, pp. 1064-1072, Aug. 2004.
[2] Y. Bar-Shalom and T. Fortmann, *Tracking and Data Association.* Academic Press, 1988.
[3] M. Brand, "Incremental Singular Value Decomposition of Uncertain Data with Missing Values," *Proc. European Conf. Computer Vision,* p. 707ff, 2002.
[4] E. Brunskill and N. Roy, "SLAM Using Incremental Probabilistic PCA and Dimensionality Reduction," *Proc. IEEE Int'l Conf. Robotics and Automation,* 2005.
[5] R. Collins and Y. Liu, "On-Line Selection of Discriminative Tracking Features," *Proc. IEEE Conf. Computer Vision,* pp. 346-352, 2003.
[6] P.M. Hall, D.R. Marshall, and R.R. Martin, "Merging and Splitting Eigenspace Models," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 9, pp. 1042-1049, Sept. 2000.
[7] J. Ho, K.C. Lee, M.H. Yang, and D.J. Kriegman, "Visual Tracking Using Learned Linear Subspaces," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 782-789, 2004.
[8] M. Isard, "PAMPAS: Real-Valued Graphical Models for Computer Vision," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 613-620, 2003.
[9] M. Isard and J.P. MacCormick, "BraMBLe: A Bayesian Multiple-Blob Tracker," *Proc. IEEE Conf. Computer Vision,* vol. 2, pp. 34-41, 2001.
[10] Z. Khan, T. Balch, and F. Dellaert, "An MCMC-Based Particle Filter for Tracking Multiple Interacting Targets," *Proc. European Conf. Computer Vision,* vol. 4, pp. 279-290, 2004.
[11] Z. Khan, T. Balch, and F. Dellaert, "A Rao-Blackwellized Particle Filter for Eigentracking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 980-986, 2004.
[12] A. Levy and M. Lindenbaum, "Sequential Karhunen-Loeve Basis Extraction and Its Application to Images," *IEEE Trans. Image Processing,* vol. 9, no. 8, pp. 1371-1374, 2000.
[13] J. Lim, D. Ross, R.S. Lin, and M.H. Yang, "Incremental Learning for Visual Tracking," *Proc. Neural Information Processing Systems,* 2004.
[14] R.S. Lin, D. Ross, J. Lim, and M.H. Yang, "Adaptive Discriminative Generative Model and Its Applications," *Proc. Neural Information Processing Systems,* 2004.
[15] B. Moghaddam and A.P. Pentland, "Probabilistic Visual Learning for Object Detection," *Proc. IEEE Conf. Computer Vision,* pp. 786-793, 1995.
[16] H.T. Nguyen and A.W.M. Smeulders, "Tracking Aspects of the Foreground against the Background," *Proc. European Conf. Computer Vision,* vol. 2, pp. 446-456, 2004.
[17] H.T. Nguyen, M. Worring, and R. van den Boomgaard, "Occlusion Robust Adaptive Template Tracking," *Proc. IEEE Conf. Computer Vision,* vol. 1, pp. 678-683, 2001.
[18] W.H Press, S.A. Teukolsky, W.T Vetterling, and B.P Flannery, *Numerical Recipes in C.* Cambridge Univ. Press, 1992.
[19] C. Rasmussen and G.D. Hager, "Probabilistic Data Association Methods for Tracking Complex Visual Objects," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 6, pp. 560-576, June 2001.
[20] D.B. Reid, "An Algorithm for Tracking Multiple Targets," *IEEE Trans. Automation and Control,* vol. 24, no. 6, pp. 843-854, 1979.
[21] D. Ross, J. Lim, and M.H. Yang, "Adaptive Probabilistic Visual Tracking with Incremental Subspace Update," *Proc. European Conf. Computer Vision,* vol. 2, pp. 470-482, 2004.
[22] H. Tao, H.S. Sawhney, and R. Kumar, "Dynamic Layer Representation with Applications to Tracking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 134-141, 2000.
[23] M.E. Tipping and C.M. Bishop, "Probabilistic Principal Component Analysis," *J. Royal Statistical Soc., Series B,* vol. 61, no. 3, pp. 611-622, 1999.
[24] D.M. Titterington, "Recursive Parameter Estimation Using Incomplete Data," *J. Royal Statistical Soc., Series B,* vol. 46, no. 2, pp. 257-267, 1984.
[25] O. Williams, A. Blake, and R. Cipolla, "A Sparse Probabilistic Learning Algorithm for Real-Time Tracking," *Proc. IEEE Conf. Computer Vision,* pp. 353-360, 2003.
[26] T. Yu and Y. Wu, "Collaborative Tracking of Multiple Targets," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 834-841, 2004.
[27] T. Zhao and R. Nevatia, "Tracking Multiple Humans in Crowded Environment," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 406-413, 2004.

Hieu T. Nguyen received the Eng and MSc degrees in computer technology from the National University "Lvivska Polytechnica" of Lviv, Ukraine, in 1994. He received the PhD degree in computer science from the University of Amsterdam, The Netherlands, in 2001. He is currently a research associate at the Intelligent Systems Lab at Rensselaer Polytechnic Institute, Troy, New York. His research interests include object tracking, object recognition, image segmentation, active learning, mathematical morphology, and content-based image retrieval. He is a member of the IEEE.

Qiang Ji received the PhD degree in electrical engineering from the University of Washington in 1998. He is currently an associate professor with the Department of Electrical, Computer, and Systems engineering at Rensselaer Polytechnic Institute (RPI). Prior to joining RPI in 2001, he was an assistant professor with the Department of Computer Science, University of Nevada at Reno. He also held research and visiting positions with Carnegie Mellon University, Western Research Company, and the US Air Force Research Laboratory. Dr. Ji's research interests are in computer vision, probabilistic reasoning with Bayesian Networks for decision making and information fusion, human computer interaction, pattern recognition, and robotics. He has published over 100 papers in peer-reviewed journals and conferences. His research has been funded by local and federal government agencies including NSF, NIH, AFOSR, ONR, DARPA, and ARO and by private companies including Boeing and Honda. Dr. Ji is a senior member of the IEEE.

Arnold W.M. Smeulders graduated from the Technical University of Delft in physics in 1977 (MSc) and in 1982 from Leiden University in medicine (PhD) on the topic of visual pattern analysis. He is the scientific director of the Intelligent Systems Lab Amsterdam, of Multimedia, the Dutch public-private partnership, and of ASCI, the national research school on computation and imaging. He is a fellow of the International Association of Pattern Recognition and honorary member of the Dutch Pattern Recognition Society. His research interests are in content-based image and video retrieval, tracking and cognitive vision, machine learning of vision, and, in the end, the picture-language question. He has written 300 papers in refereed journals and conferences and graduated 28 PhD students. The group has an extensive record in cooperation with Dutch institutions and industry in the area of multimedia and video analysis. He was an associated editor of the *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Currently, he is an associate editor of the *International Journal for Computer Vision* and the *IEEE Transactions on Multimedia*. He is a senior member of the IEEE.