Robust multi-target tracking using spatio-temporal context

Hieu T. Nguyen , Qiang Ji
Department of Electrical, Computer,
and Systems Engineering
Rensselaer Polytechnic Institute, USA
{nguyeh2, qji}@ecse.rpi.edu

Arnold W.M. Smeulders
Intelligent Sensory Information Systems
Faculty of Science, University of Amsterdam
The Netherlands
smeulders@science.uva.nl

Abstract

In multi-target tracking, the maintaining of the correct identity of targets is challenging. In the presented tracking method, accurate target identification is achieved by incorporating the appearance information of the spatial and temporal context of each target. The spatial context of a target involves local background and nearby targets. The first contribution of the paper is to provide a new discriminative model for multi-target tracking with the embedded classification of each target against its context. As a result, the tracker not only searches for the image region similar to the target but also avoids latching on nearby targets or on a background region. The temporal context of a target includes its appearances seen during tracking in the past. The past appearances are used to train a probabilistic PCA that is used as the measurement model of the target at the present. As the second contribution, we develop a new incremental scheme for probabilistic PCA. It can update accurately the full set of parameters including a noise parameter still ignored in related literature. The experiments show robust tracking performance under the condition of severe clutter, occlusions and pose changes.

1 Introduction

Tracking an object can be seen as a dynamic one-against-everything-else classification problem. When more than one object to track appears on the scene the problem evolves into a dynamic multi-class problem. The problem of jointly tracking of multiple targets is as challenging as it is interesting. Interesting are applications in video surveillance where one wishes to understand interactions between persons, and in sports where one aims to understand patterns of play, to name just two applications. Challenging is safeguarding the proper identity of all objects, especially hard when objects have little distinction in their appearance. The problem is further complicated by the mutual occlusion, change in pose, and change in lighting. We aim to maintain object identity in these conditions for the case of a fixed camera.

The traditional approach in resolving the ambiguous identity of 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 [1] and the multiple hypotheses tracking algorithm in [19], enforce a data association variable into the target likelihood, which rules out configurations where multiple targets associate with the same image region. Recent methods [10, 24, 13] add a prior term to the likelihood function in order to prevent any pair of targets from getting too close. This is not a realistic condition, where proximity between targets and occlusion in many application is the crux. In general, by putting constraints on target positions, the referenced methods succeed in preventing the coalescence of targets, but take no measure towards identity switching. In most cases, each target is searched for by maximizing its own likelihood computed without considering the appearance of the other targets. The sensitivity of the likelihood to appearance changes can then lead to a false classification among targets. Some joint likelihood models proposed in [18, 11, 25] can better describe the overlap between targets during occlusions but they still minimize the likelihood of individual targets.

To achieve an accurate identification of targets, we propose to incorporate the appearance information from the target context. 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 involves all prior knowledge regarding object appearance, which has been obtained up to this moment of time.

Recent work on single target tracking has pointed out the advantage of the spatial context for tracking. In [5], the authors propose to select online color features most discriminating a target object from a local background window. The algorithm of [17] learns and maintains online a foreground-

Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)
0-7695-2597-0/06 $20.00 © 2006 IEEE

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 changing appearances of the object. The same principle holds for multi-target tracking where superior distinction between targets leads to easier identification. *The first contribution of the presented work is to develop a new probabilistic framework for multi-target tracking by a built-in classifier for proper distinction of the targets against their spatial context.*

Another condition for accurate target identification is a robust appearance model of each target. One desirable property is the ability to represent a broad range of object appearances, including different views. A priori learning can provide good models [2, 6], but may not be possible in practice. So we focus on learning a model online from the past tracking results. Such a model should be able to detect the reappearance of an aspect of the object, which has been seen in the past. This will help to recover the track from an occlusion or a temporary failure. The traditional models that can represent multiple appearances include mixture of gaussians or eigenspace [23]. The difficulty is that the model needs be learnt incrementally upon the arrival of new tracking results and under the condition of a limited memory and a limited time. The algorithms for the online learning of a mixture of gaussians [22] require the input samples be statistically independent, and furthermore need time to converge. Recent tracking algorithms therefore focus on the eigenspace model [20, 12, 15, 9]. They rely on the recursive SVD algorithm [14] 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) [7, 16], requires an additional parameter being the variance of the noise in non-principal 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 the existing methods, this noise parameter is predefined or set to a fraction of the eigenvalue of the smallest principal component [15]. This adhoc approach has no theoretical justification. A rather different incremental scheme in [4] 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, the method will not work when the number of new observations is smaller than the number of principal components. *The second contribution of this paper is a new incremental scheme of the probabilistic PCA model accurately updating the full set of parameters.* Our PPCA solution is an approximation of the maximum likelihood estimation for the entire history of observation data, and can be updated upon the arrival of even one additional sample.

The paper is structured as follows. Section 2 presents our framework for multi-target tracking, including the probabilistic model and the inference. In section 3 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 4.

2 Classification-based framework for tracking multiple targets

This section presents a novel classification-based framework for multi-target tracking.

Let M be the number of the targets that we want to track, and $\boldsymbol{x}_i$ be the position of the ith target. For the 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: $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M\}$.

2.1 The probabilistic model

We propose the probabilistic model shown in Figure 1. 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 3. 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, implying that there are 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 since the label distribution usually can be estimated sufficiently accurately from current measurements and the predicted prior. The posterior distribution of each label $P(\ell_i|\boldsymbol{X}^{(t-1)}, \boldsymbol{z}_i^{(t)})$ can be calculated straightforward from $p(\ell_i|\boldsymbol{X}^{(t-1)})$ and $p(\boldsymbol{z}_i^{(t)}|\ell_i)$. The distribution of each $\boldsymbol{x}_k^{(t)}$ is then independently inferred using $P(\ell_i|\boldsymbol{X}^{(t-1)}, \boldsymbol{z}_i^{(t)})$ and $p(\boldsymbol{x}_k^{(t)}|\ell_1, \ldots, \ell_N)$.

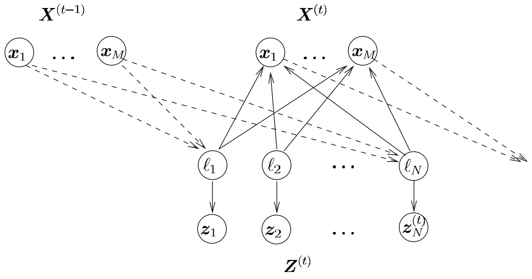


Figure 1. *The proposed probabilistic model for multi-target tracking. $\boldsymbol{x}_k$ is the position of kth target, $\boldsymbol{z}_i$ is the observation at position i, ℓ_i is the class label of position i.*

The three 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:

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

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:

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

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 Figure 2.
2) *The measurement distribution* $p(\boldsymbol{z}_i^{(t)}|\ell_i)$:
The measurement distribution in each class is assumed Gaussian. The background distribution at each location is represented by an isotropic Gaussian learnt a priori. A priori learning is possible as the camera is fixed. For the target distribution, we employ the probabilistic PCA model [7], a non-isotropic model which provides more flexibility in modelling appearance changes. Unlike the background, it is usually impossible to learn a target distribution a priori. Section 3 presents a method for the online construction of this distribution from the tracking results, requiring initialization of the target in the first frame only.

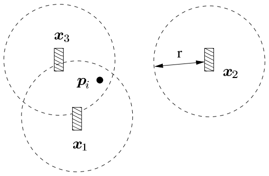


Figure 2. *The prediction of the label prior probability. In this example, only targets 1 and 3 contribute to the label prior at $\boldsymbol{p}_i$.*

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:

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

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

2.2 State inference and target search

We search for the k-th 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:

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

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:

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

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

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

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 eq. (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 $p(\boldsymbol{x}_k^{(t)} = \boldsymbol{p}_i | \boldsymbol{Z}^{(t)}, \boldsymbol{X}^{(t-1)}) > p(\boldsymbol{x}_k^{(t)} = \boldsymbol{p}_{i'} | \boldsymbol{Z}^{(t)}, \boldsymbol{X}^{(t-1)})$ holds if and only if $p(\ell_i = k | \boldsymbol{z}_i^{(t)}, \boldsymbol{X}^{(t-1)}) > p(\ell_{i'} = k | \boldsymbol{z}_{i'}^{(t)}, \boldsymbol{X}^{(t-1)})$.*

The proof will be presented in the journal version of this paper. It follows that the maximization of the probability of the position of a target can be achieved by maximizing the probability of the corresponding class label:

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

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

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

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

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

As observed in eq. (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.

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 eq. (6). In view of this, eq. (4) is also a reasonable approach to locate the target. This approach which works effectively in most tracking tasks, and has been common in tracking [21, 25].

3 Online construction of the measurement distribution using the incremental probabilistic PCA

In this section, we address the construction of 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 $\boldsymbol{C}_k$:

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

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 size of background measurements is the average of the size of the targets.

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 much 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 [7]:

$$\boldsymbol{C}_k = \sigma_k^2 \boldsymbol{I} + \boldsymbol{W}_k \boldsymbol{W}_k^T \quad (11)$$

$\boldsymbol{W}_k$ 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 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. 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. The model learnt from the past appearances will help to recognize the object in the future should these appearances return. Note however that the measurements obtained during occlusions are not reliable. We therefore collect only the measurements obtained when there is no overlap between the considered target and the other targets.

3.1 Maximum likelihood solution of probabilistic PCA

According to [7], 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:

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

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 the $\lambda_{1,k}, \ldots, \lambda_{q,k}$, and $\boldsymbol{R}$ is an arbitrary $q_k \times q_k$ orthogonal matrix.

The estimated covariance matrix is:

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

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 eq. (12)- (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.

3.2 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:

$$\boldsymbol{\mu}^{(t+1)} = \frac{1}{t+1}\sum_{i=1}^{t+1}\boldsymbol{z}^{(i)} = \frac{t}{t+1}\boldsymbol{\mu}^{(t)} + \frac{1}{t+1}\boldsymbol{z}^{(t+1)} \tag{17}$$

The new observation covariance matrix is:

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

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 eq. (16), yielding:

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

We remark that in related methods [14, 8, 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 eq. (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 an update 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:

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

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 $\boldsymbol{\Sigma} = \text{diag}\{\lambda_1^{(t)} - \sigma^{(t)^2}, \ldots, \lambda_q^{(t)} - \sigma^{(t)^2}\}$, $\alpha = \boldsymbol{y}^T\boldsymbol{y}$, 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 $\boldsymbol{\Gamma} = \text{diag}\{\gamma_1, \ldots, \gamma_{q+1}\}$, 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 $\boldsymbol{V} = [\boldsymbol{v}_1^{(t+1)}, \ldots, \boldsymbol{v}_{q+1}^{(t+1)}]$. Eq. (19) is rewritten as:

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

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 eq. (13), σ is updated as:

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

The incremental PPCA is summarized as follows:
For each target:

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

The initial PPCA model is learnt from an initial set of measurements $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_k$ using the batch mode algorithm [7]. 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 overall complexity is $O(q^3) + O(dq)$ per each update. Since q is small, the algorithm is efficient and can be applied to real time applications.

4 Experiments

The tracking is initialized by specifying the position and size of each target in the first frame. 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 threshold r indicates the maximum displacement in one time step, so that the algorithm can find the target in the next frame. In the experiment, 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.

The tracking results of the proposed method are shown in Figure 3 for several video sequences. For comparison, Figure 4 shows the results of independently applying multiple instances of a single target tracker, where each target is searched for by maximizing the ratio of its likelihood to the background likelihood.

In Figure 3a, 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 independent trackers quickly lost track of the first target at the occlusion, see Figure 4. 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.

A difficult example is shown in Figure 3b. 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 independent trackers, as shown in Figure 4, the window of targets 1,2,5 melts together at the

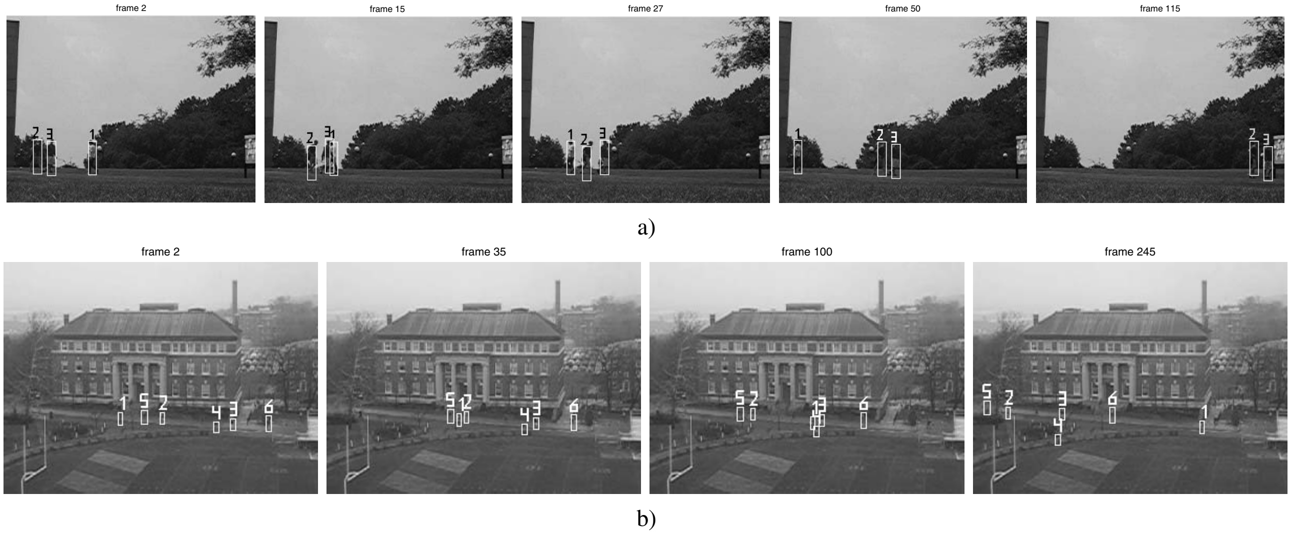


Figure 3. *The result of the proposed algorithm for tracking multiple approaching targets with occlusions. The number on top each target indicates its label. See also the enclosed videos.*

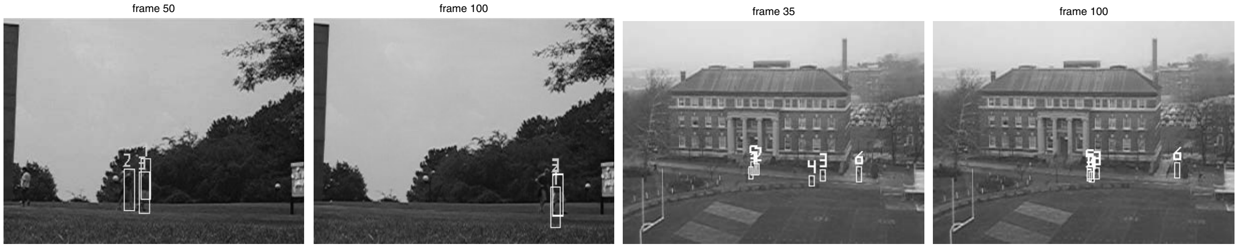


Figure 4. *The results of the independent trackers for the sequences in Figure 3.*

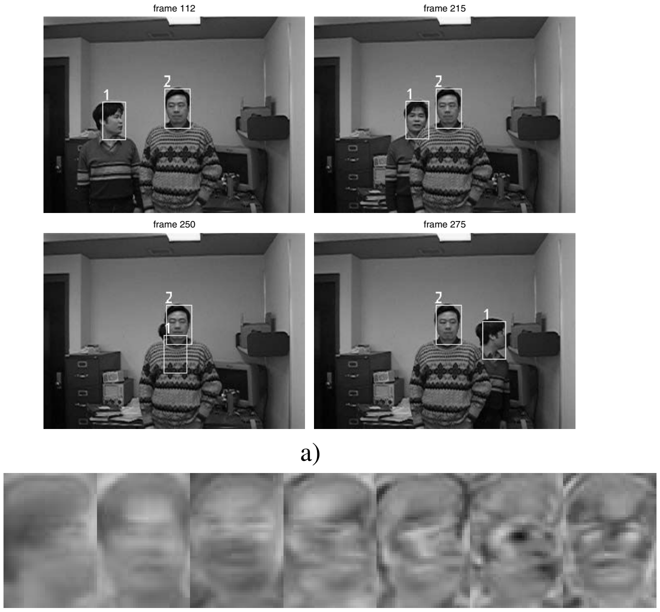
first occlusion in frame 35. Erroneously, they stick to one target until frame 140. The same thing occurs for targets 3 and 4. The same thing occurs for two other targets. As a consequence, the independent trackers lose track and cannot recover.

The power of incremental PPCA in modelling object appearance is demonstrated in Figure 5a. 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 recognized the profile view after occlusion since it has been seen earlier in frame 112. The eigenimages obtained are shown in Figure 5b. They also depict different views of the head. The algorithm failed to recapture the head however when we replaced the PPCA model by either a fixed template or an adaptive template obtained by frame averaging over a short period of time.

5 Conclusion

A new approach has been proposed for tracking multiple targets, emphasizing on 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 the target from nearby targets and the background by intensity of pixels in the target window. Before searching for the next target position, all targets will be classified. Maximization of the probability of the target label, rather than the target likelihood, avoids that the target latches on image regions with a similar appearance as the other targets or the background. By separating targets in appearance space and not in position space, the



a)

b)

Figure 5. *a) Tracking results of the proposed algorithm in the condition of occlusion and pose change; b) The eigenimages obtained from the PPCA.*

problem of target coalescence and identity switching has been solved effectively.

The representation which makes the method work is the Probabilistic PCA incrementally updated on line without storage of past measurements. A robust appearance model is constructed for each target. The model effectively represents a diverse set of appearances, effectively providing a long-term memory, instrumental in re-detecting an object after occlusions and severe pose changes.

References

[1] Y. Bar-Shalom and Th. Fortmann. *Tracking and data association*. Academic Press, 1988.

[2] M. J. Black and A. D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. In *Proc. of ECCV*, pages 329–342, 1996.

[3] M. Brand. Incremental singular value decomposition of uncertain data with missing values. In *Proc. of ECCV*, 2002.

[4] E. Brunskill and N. Roy. SLAM using incremental probabilistic PCA and dimensionality reduction. In *Proc. of ICRA*, 2005.

[5] R. Collins and Y. Liu. On-line selection of discriminative tracking features. In *Proc. ICCV*, 2003.

[6] T.F. Cootes, G.V. Wheeler, K.N. Walker, and C.J. Taylor. View-based active appearance models. *Image and Vision Computing*, 20(9-10):657–664, 2002.

[7] Tipping M. E. and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61(3):611–622, 1999.

[8] P.M. Hall, D.R. Marshall, and R.R. Martin. Merging and splitting eigenspace models. *PAMI*, 22(9):1042–1049, 2000.

[9] J. Ho, K.C. Lee, M.H. Yang, and D.J. Kriegman. Visual tracking using learned linear subspaces. In *CVPR*, 2004.

[10] M. Isard. PAMPAS: Real-valued graphical models for computer vision. In *Proc. CVPR*, pages I: 613–620, 2003.

[11] M. Isard and J.P. MacCormick. BraMBLe: A Bayesian multiple-blob tracker. In *Proc. ICCV*, II: 34–41, 2001.

[12] R.S. Lin M.H. Yang J. Lim, D. Ross. Incremental learning for visual tracking. In *Proc. of NIPS*, 2004.

[13] Z. Khan, T. Balch, and F. Dellaert. An MCMC-based particle filter for tracking multiple interacting targets. In *Proc. of ECCV*, pages Vol IV: 279–290, 2004.

[14] A. Levy and M. Lindenbaum. Sequential Karhunen-Loeve basis extraction and its application to images. *IEEE Trans. on Image Processing*, 9(8):1371–1374, 2000.

[15] R.S. Lin, D. Ross, J. Lim, and M.H. Yang. Adaptive discriminative generative model and its applications. In *Proc. of NIPS*, 2004.

[16] B. Moghaddam and A.P. Pentland. Probabilistic visual learning for object detection. In *Proc. of ICCV*, p. 786–793, 1995.

[17] H.T. Nguyen and A.W.M. Smeulders. Tracking aspects of the foreground against the background. In *ECCV*, 2004.

[18] C. Rasmussen and G.D. Hager. Probabilistic data association methods for tracking complex visual objects. *IEEE Trans. on PAMI*, 23(6):560–576, 2001.

[19] D.B. Reid. An algorithm for tracking multiple targets. *IEEE Trans. on Automation and Control*, 24(6):843–854, 1979.

[20] D. Ross, J. Lim, and M.H. Yang. Adaptive probabilistic visual tracking with incremental subspace update. In *Proc. of ECCV*, pages Vol II: 470–482, 2004.

[21] H. Tao, H.S. Sawhney, and R. Kumar. Dynamic layer representation with applications to tracking. In *Proc. CVPR00*, pages II:134–141, 2000.

[22] D. M. Titterington. Recursive parameter estimation using incomplete data. *J. Royal Stat. Soc. B*, 46(2):257–267, 1984.

[23] M. Turk and A. Pentland. Eigenfaces for recognition. *J. Cognitive Neuroscience*, 3(1):71–86, 1991.

[24] T. Yu and Y. Wu. Collaborative tracking of multiple targets. In *Proc. of CVPR*, pages I: 834–841, 2004.

[25] T. Zhao and R. Nevatia. Tracking multiple humans in crowded environment. In *Proc. of CVPR.*, pages II: 406–413, 2004.

0-7695-2597-0/06 $20.00 © 2006 **IEEE**