Single and Multiple Object Tracking Using Log-Euclidean Riemannian Subspace and Block-Division Appearance Model

Weiming Hu, Xi Li, Wenhan Luo, Xiaoqin Zhang, Stephen Maybank, and Zhongfei Zhang

Abstract—Object appearance modeling is crucial for tracking objects, especially in videos captured by nonstationary cameras and for reasoning about occlusions between multiple moving objects. Based on the log-euclidean Riemannian metric on symmetric positive definite matrices, we propose an incremental log-euclidean Riemannian subspace learning algorithm in which covariance matrices of image features are mapped into a vector space with the log-euclidean Riemannian metric. Based on the subspace learning algorithm, we develop a log-euclidean block-division appearance model which captures both the global and local spatial layout information about object appearances. Single object tracking and multi-object tracking with occlusion reasoning are then achieved by particle filtering-based Bayesian state inference. During tracking, incremental updating of the log-euclidean block-division appearance model captures changes in object appearance. For multi-object tracking, the appearance models of the objects can be updated even in the presence of occlusions. Experimental results demonstrate that the proposed tracking algorithm obtains more accurate results than six state-of-the-art tracking algorithms.

Index terms—Visual object tracking, occlusion reasoning, log-euclidean Riemannian subspace, incremental learning, block-division appearance model

1 Introduction

Visual object tracking [3] is one of the most fundamental tasks in applications of video motion processing, analysis, and data mining, such as human-computer interaction, visual surveillance, and virtual reality. Constructing an effective object appearance model to deal robustly with appearance variations is crucial for tracking objects, especially in videos captured by moving cameras and for reasoning about occlusions between multiple moving objects. Object appearance models for visual tracking can be based on region color histograms, kernel density estimates, GMMs (Gaussian mixture models) [6], conditional random fields, or learned subspaces [14], etc. Among these appearance models, subspace-based ones have attracted much attention because of their robustness.

1.1 Subspace-Based Appearance Models

In subspace-based appearance models, the matrices of the pixel values in image regions are flattened (i.e., rewritten) into vectors, and global statistical information about the pixel values is obtained by PCA (principal component analysis) for the vectors. Black and Jepson [2] present a subspace learning-based tracking algorithm. A pretrained, view-based eigenbasis representation is used for modeling appearance variations under the assumption that the different appearances are contained in a fixed subspace. However, the algorithm does not work well in cluttered scenes with large lighting changes because the subspace constancy assumption fails. Ho et al. [11] present a visual tracking algorithm based on linear subspace learning. In each subspace update, the subspace is recomputed using only recent batches of the tracking results. However, using the means of the tracking results in a number of consecutive frames as the learning samples may lose accuracy, and computing the subspace using only the recent batches of the tracking results may result in tracker drift if large appearance changes occur. Skocaj and Leonardis [13] present a weighted incremental PCA algorithm for subspace learning. Its limitation is that each update includes only one new sample, rather than multisamples, and as a result it is necessary to update the subspace at every frame. Li [12] proposes an incremental PCA algorithm for subspace learning. It can deal with multisamples in each update. However, it assumes that the mean vector of the vectors obtained by flattening the new arriving images is equal to the mean vector for the previous images. The subspace model cannot adapt to large changes in the mean. Ross et al. [14] propose a generalized tracking framework based on the incremental image-as-vector subspace learning method. It removes the assumption that the mean of the previous data is equal to the mean of the new data in [12]. However, it does not directly capture and model the spatial

- *W. Hu, X. Li, W. Luo, and X. Zhang are with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, No. 95, Zhongguancun East Road, PO Box 2728, Beijing 100190, P.R. China. E-mail: {wmhu, lixi whluo, xqzhang}@nlpr.ia.ac.cn.*
- *S. Maybank is with the Department of Computer Science and Information Systems, Birkbeck College, Malet Street, London WC1E 7HX, United Kingdom. E-mail: sjmaybank@dcs.bbk.ac.uk.*
- *Z. Zhang is with the Department of Computer Science, Watson School of Engineering and Applied Sciences, Binghamton University, N 16, Engineering Building, Binghamton, NY 13902-6000. E-mail: zhongfei@cs.binghamton.edu.*

Manuscript received 11 Feb. 2010; revised 31 Aug. 2011; accepted 8 Jan. 2012; published online 30 Jan. 2012.
Recommended for acceptance by P. Perez.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2010-02-0091.
Digital Object Identifier no. 10.1109/TPAMI.2012.42.

correlations between values of pixels in the tracked image region. Lee and Kriegman [9] present an online algorithm to incrementally learn a generic appearance model for video-based recognition and tracking. Lim et al. [10] present a human tracking framework using a robust identification of system dynamics and nonlinear dimension reduction techniques. Only image features are used in the algorithms in [9] and [10], but the spatial correlations in the tracked image region are not modeled. Furthermore, they use a number of predefined prior models whose training requires a large number of samples.

In summary, the general limitations of the current subspace-based appearance models include the following:

- They do not directly use object pixel values' local relations, which can be quantitatively represented by pixel intensity derivatives, etc. These local relations are, to a large extent, invariant to complicated environmental changes. For example, variances in lighting can cause large changes in pixel values, while the changes in the spatial derivatives of the pixel intensities may be much less.
- In applications to multi-object tracking with occlusion reasoning, it is difficult to update the object appearance models during occlusions.

It is a challenge for subspace-based appearance models to utilize the local relations between object pixels to increase the robustness of object tracking and to achieve multi-object tracking with updating of the object appearance models even during occlusions.

1.2 Riemannian Metrics

A covariance matrix descriptor [24], [29] which is obtained based on the features of intensity derivatives captures the spatial correlations of the features extracted from an object region. The covariance matrix descriptor is robust to variations in illumination, viewpoint, and pose. The nonsingular covariance matrix is contained in a connected manifold of symmetric positive definite matrices. Statistics for covariance matrices of image features can be constructed using an appropriate Riemannian metric [26], [27]. Researchers have applied Riemannian metrics to model object appearances. Porikli et al. [24] propose a Riemannian metric-based object tracking method in which object appearances are represented using the covariance matrix of image features. Tuzel et al. [25] propose an algorithm for detecting people by classification on Riemannian manifolds. Riemannian metrics have been applied to the modeling of object motions using matrices in an affine group. Kwon et al. [61] explore particle filtering on the 2D affine group for visual tracking. Porikli and Tuzel [62] propose a Lie group learning-based motion model for tracking combined with object detection.

The algorithms in [24] and [25] represent object appearances by points on a Riemannian manifold and utilize an affine-invariant Riemannian metric to calculate a Riemannian mean for the data. There is no closed form solution for the Riemannian mean. It is computed using an iterative numerical procedure [30]. Arsigny et al. [28] propose the log-euclidean Riemannian metric for statistics on the manifold of symmetric positive definite matrices. This metric is simpler than the affine-invariant Riemannian metric. In particular, the computation of a sample's Riemannian mean is more efficient than in the affine invariant case. Kwon et al. [61] propose a closed form approximation to the Riemannian mean of a set of particle offsets. In this paper, we apply the log-euclidean Riemannian metric to represent object appearances and construct a new subspace-based appearance model for object tracking.

1.3 Our Work

Based on the log-euclidean Riemannian metric, we propose an incremental log-euclidean Riemannian subspace learning algorithm [1] which is the basis of our new object tracking algorithms whose main components include a block-division appearance model, Bayesian state inference for single object tracking, and multi-object tracking with occlusion reasoning. In our incremental subspace learning algorithm, covariance matrices of image features are transformed into log-euclidean Riemannian matrices which are then unfolded into vectors whose dominant projection subspace is found and updated. In the block-division appearance model, the object appearance region is divided into several blocks. For each block, a low dimensional log-euclidean Riemannian subspace model is learned online. The likelihood of a candidate block given the learned log-euclidean subspace model is computed, and then a block-related likelihood matrix is obtained. This matrix is locally filtered by spatial relations between blocks and globally filtered by a spatial Gaussian kernel. In the Bayesian state inference for single object tracking, the object state is estimated using a particle filter. The block-related image features associated with the optimal state are used to update the appearance model. In our algorithm for tracking multi-objects, the changes in block appearances are used to reason about the occlusion relations between objects. The appearance models for the unoccluded blocks are updated but those for the occluded blocks are unchanged.

Our work is original in the following ways:

- Our incremental log-euclidean Riemannian subspace learning algorithm captures the local correlations at the pixel level. The vector space properties of the log-euclidean Riemannian space make the linear subspace analysis very effective in exploring the information in the covariance matrices.
- Our log-euclidean block-division appearance model captures both the global and local spatial correlations of object appearances at the block level due to the spatial filtering scheme.
- A single object tracking algorithm for videos captured by nonstationary or stationary cameras is proposed. Incremental updating of the object appearance model captures variations in illumination, pose, and view.
- An algorithm for tracking multi-objects with occlusion reasoning in videos captured by stationary or nonstationary cameras is proposed. The object appearance models can be updated even during occlusions.
- The experimental results show that our tracking algorithm obtains more accurate results than six state-of-the-art object tracking algorithms.

The remainder of the paper is organized as follows: Section 2 discusses related work. Section 3 proposes our incremental log-euclidean Riemannian subspace learning algorithm. Section 4 presents our log-euclidean block-division appearance model. Section 5 describes the Bayesian state inference for single object tracking. Section 6 covers our algorithm for multi-object tracking with occlusion reasoning. Section 7 demonstrates experimental results. Section 8 summarizes the paper.

2 Related Work

In Section 1, we reviewed the work closely related to subspace-based appearance modeling in order to motivate the paper. In order to give the broad context, we further briefly review the related work on appearance modeling-based single object tracking and multi-object tracking with stationary or nonstationary cameras.

2.1 Single Object Tracking

A number of algorithms focus on specific types of appearance changes. As change in illumination is the most common cause of object appearance variation, many algorithms focus on such changes. Hager and Belhumeur [37] propose a typical tracking algorithm which uses an extended gradient-based optical flow method to track objects under varying illuminations. Zhao et al. [22] present a fast differential EMD (Earth Mover's Distance) tracking method which is robust to illumination changes. Silveira and Malis [17] present an image alignment algorithm to cope with generic illumination changes during tracking. Some algorithms focus on dealing with object appearance deformations. For example, Li et al. [8] use a generalized geometric transform to handle object deformation, articulated objects, and occlusions. Ilic and Fua [20] present a nonlinear beam model for tracking large appearance deformations. There exists work on dealing with appearance changes in scale and orientation. For example, Yilmaz [16] proposes an object tracking algorithm based on adaptively varying scale and orientation of a kernel. The above algorithms are robust to the specific appearance changes for which they are designed, but they are oversensitive to other appearance changes.

More attention has been paid to the construction of general appearance models which are adapted to a wide range of appearance variations [23], [51]. Black et al. [4] and Jepson et al. [5] employ mixture models to explicitly represent and recover object appearance changes during tracking. Zhou et al. [6] embed appearance models adaptive to various appearance changes into a particle filter to achieve visual object tracking. Yu and Wu [7] propose a spatial appearance model which captures nonrigid appearance variations efficiently. Yang et al. [44] use negative data constraints and bottom-up pairwise data constraints to dynamically adapt to the changes in the object appearance. Kwon and Lee [45] use a local patch-based appearance model to maintain relations between local patches by online updating. Mei and Ling [50] sparsely represent the object in the space spanned by target and trivial image templates. Babenko et al. [36] use a set of image patches to update a classifier-based appearance model. These general appearance models can adaptively handle a wide range of appearance changes. However, they are less robust to specific types of appearance changes than the algorithms which are designed for these specific appearance changes.

There are algorithms that use invariant image features or key points to implicitly represent object appearance. For example, Tran and Davis [21] propose robust regional affine invariant image features for visual tracking. Grabner et al. [18] describe key points for tracking using an online learning classifier. He et al. [52] track objects using the relations between local invariant feature motions and the object global motion. Ta et al. [53] track scale-invariant interest points without computing their descriptors. The affine-invariance properties make these algorithms more robust to large local deformations and effective in tracking textured appearances. Partial occlusions can be dealt with by partial matching of key points. However, they are sensitive to large appearance changes and background noise.

All of the aforementioned specific model-based methods, the general model-based methods, and the key-point-based methods share a problem in that the appearance model is constructed using the values of the pixels in an image region, without any direct use of local relations between the values of neighboring pixels.

2.2 Multi-Object Tracking

There has been much work on tracking multi-objects using object appearance models in videos captured by stationary or nonstationary cameras.

2.2.1 Multi-Object Tracking with Stationary Cameras

For stationary cameras, background subtraction, image calibration, and homography constraints between multi-cameras, etc., are often employed to obtain prior information about the positions of moving objects [40], [47]. Khan and Shah [41] use spatial information in a color-based appearance model to segment each person into several blobs. Occlusions are handled by keeping track of the visible blobs belonging to each person. Ishiguro et al. [46] classify the type of object motion using a few distinct motion models. A switching dynamic model is used in a number of object trackers. The algorithms in [40], [41], [46], and [47] depend on background subtraction. Zhao and Nevatia [42] adopt a 3D shape model as well as camera models to track people and handle occlusions. Mittal and Davis [33] use appearance models to detect people and an occlusion likelihood is applied to reason about occlusion relations between people. Joshi et al. [43] track a 3D object through significant occlusions by combining video sequences from multiple nearby cameras. The algorithms in [33], [42], and [43] depend on a costly camera calibration. Fleuret et al. [48] use multicameras to model positions of multiobjects during tracking. Their algorithm depends on a discrete occupancy grid, besides camera calibration. Khan and Shah [49] track multiple occluding people by localizing them on multiple scene planes. The algorithm depends on the planar homography occupancy constraint between multicameras.

Although the above algorithms achieve good performances in multi-object tracking, the requirement for stationary cameras limits their applications.

2.2.2 Multi-Object Tracking with Nonstationary Cameras

For nonstationary cameras, background subtraction, calibration, and homography constraints cannot be used. As a

result, multi-object tracking with nonstationary cameras is much more difficult than with stationary cameras. Wu and Nevatia [38], [39] use four detectors for parts of a human body and a combined human detector to produce observations during occlusions. Wu et al. [15] track two faces through occlusions using multiview templates. Qu et al. [54] use a magnetic-inertia potential model to carry out the multi-object labeling. Yang et al. [55] track multi-objects by finding the Nash equilibrium of a game. Jin and Mokhtarian [56] use a variational particle filter to track multi-objects. One limitation in current algorithms for tracking multi-objects in videos taken by nonstationary cameras is the assumption that the object appearance models are unchanged in the presence of occlusions. When there are large changes in object appearances during occlusions, the objects cannot be accurately tracked.

In recent years, many detection-based tracking methods have been proposed for multipedestrians [57], [58], [59], [60]. These methods first detect the pedestrians and then assign the detection responses to the tracked trajectories using different data association strategies, such as cognitive feedback to visual odometry, min-cost flow networks [57], the hypothesis selection [58], the Hungarian algorithm [59], and continuous segmentation [60]. The performance of these detection-based tracking methods greatly depends on the accuracy of pedestrian detection.

In summary, the algorithms introduced in Sections 2.2.1 and 2.2.2 still have limitations. It is a challenge to track multi-objects in a general way without prior knowledge about objects, with appearance model updating during occlusions, and with either stationary or nonstationary cameras.

3 Incremental Log-Euclidean Riemannian Subspace Learning

First, the image covariance matrix descriptor and the Riemannian geometry for symmetric positive definite matrices are briefly introduced for the convenience of readers. Then, the proposed incremental log-euclidean Riemannian subspace learning algorithm is described.

3.1 Covariance Matrix Descriptor

Let f_i be a d-dimensional feature vector of pixel i in an image region. The vector f_i is defined by $(x, y, (E_j)_{j=1,\ldots,\mathfrak{V}})$, where (x, y) are the pixel coordinates, $\mathfrak{V}$ is the number of color channels in the image, and

$$E_j = \left(I^j, |I^j_x|, |I^j_y|, \sqrt{(I^j_x)^2 + (I^j_y)^2}, |I^j_{xx}|, |I^j_{yy}|, \arctan\frac{|I^j_y|}{|I^j_x|} \right), \tag{1}$$

where I^j is the intensity value in the jth color channel, I^j_x, I^j_{xx}, I^j_y, and I^j_{yy} are the first and second order intensity derivatives in the jth color channel, and the last term is the first-order gradient orientation. For a grayscale image, f_i is a 9D feature vector (i.e., $\mathfrak{V} = 1$ and $d = 9$). For a color image with three channels, f_i is a 23D vector (i.e., $\mathfrak{V} = 3$ and $d = 23$). The calculation of the intensity derivatives depends on the intensity values of the pixels neighboring to the pixel i. So, the local relation between values of neighboring pixels is described by the intensity derivatives in the feature vector.

Given an image region R, let L be the number of pixels in the region and let μ be the mean of $\{f_i\}_{i=1,2,\ldots,L}$. The image region R is represented using a $d \times d$ covariance matrix C_R [29] which is obtained by

$$C_R = \frac{1}{L-1}\sum_{i=1}^{L}(f_i - \mu)(f_i - \mu)^T. \tag{2}$$

The covariance matrix descriptor of a grayscale or color image region is a 9×9 or 23×23 symmetric matrix. The pixels' coordinates are involved in the computation of the covariance matrix in order to include the spatial information about the image region and the correlations between the positions of the pixels and the intensity derivatives into the covariance matrix.

3.2 Riemannian Geometry for Symmetric Positive Definite Matrices

As discussed in Section 1.2, the nonsingular covariance matrix lies in a connected manifold of symmetric positive definite matrices. The Riemannian geometry for symmetric positive definite matrices is available for calculating statistics of covariance matrices. The Riemannian geometry depends on the Riemannian metric, which describes the distance relations between samples in the Riemannian space and determines the computation of the Riemannian mean.

In the space of $d \times d$ symmetric positive definite matrices, the exponential and the logarithm of matrices are fundamental matrix operations. Given a symmetric positive definite matrix A, the SVD (singular value decomposition) for A ($A = U\Sigma U^T$) produces the orthogonal matrix U and the diagonal matrix $\Sigma = Diag(\lambda_1, \lambda_2, \ldots, \lambda_d)$, where $\{\lambda_i\}_{i=1,2,\ldots,d}$ are the eigenvalues of A. Then, the matrix exponential of A is defined by:

$$\begin{aligned}\exp(A) &= \sum_{k=0}^{\infty}\frac{A^k}{k!} \\ &= U \cdot Diag(\exp(\lambda_1), \exp(\lambda_2), \ldots, \exp(\lambda_d)) \cdot U^T.\end{aligned} \tag{3}$$

The matrix logarithm of A is defined by

$$\begin{aligned}\log(A) &= \sum_{k=1}^{\infty}\frac{(-1)^{k+1}}{k}(A - I_d)^k \\ &= U \cdot Diag(\log(\lambda_1), \log(\lambda_2), \ldots, \log(\lambda_d)) \cdot U^T,\end{aligned} \tag{4}$$

where I_d is the $d \times d$ identity matrix.

The affine-invariant Riemannian metric is widely used on the space of symmetric positive definite matrices. The limitations of the affine-invariant Riemannian metric are 1) there is no closed form solution for the Riemannian mean of a set of symmetric positive definite matrices [30]; 2) the distance from matrix X to matrix Y is given by $\|\log(X^{-\frac{1}{2}} \cdot Y \cdot X^{-\frac{1}{2}})\|_F$, where $\|\cdot\|_F$ is a Frobenius norm. It is complicated to evaluate due to the matrix inverse operation for the computation of X to the negative power of $1/2$, and the multiplication of three matrices.

The log-euclidean Riemannian metric was proposed in [28] and [61]. The symmetric positive definite matrices are a subset of a Lie group. Under the log-euclidean Riemannian metric, the tangent space at the identity element in the Lie group forms a Lie algebra, which has a vector space

structure [31]. In the Lie algebra, the mean μ of matrix logarithms obtained using the matrix logarithmic operation in (4) is simply their arithmetic mean [61]. Given N symmetric positive definite matrices $\{X_i\}_{i=1}^N$, the mean μ in the Lie algebra is explicitly computed by

$$\mu = \frac{1}{N}\sum_{i=1}^{N}\log(X_i). \quad (5)$$

The mean μ can be mapped into the Lie group using the matrix exponential operation in (3), forming the Riemannian mean μ^R in the Lie group. Corresponding to (5), μ^R is obtained by

$$\mu^R = \exp\left(\frac{1}{N}\sum_{i=1}^{N}\log(X_i)\right). \quad (6)$$

Moreover, under the log-euclidean Riemannian metric, the distance between two points X and Y in the space of symmetric positive definite matrices is measured by $\|\log(X) - \log(Y)\|_F$. The Riemannian mean and distance under the log-euclidean metric are simpler to compute than those under the affine-invariance metric. In this paper, the log-euclidean Riemannian metric is used to calculate statistics of covariance matrices of image features.

3.3 Incremental Log-Euclidean Riemannian Subspace Learning

As the log-euclidean Riemannian space (i.e., the tangent space at the identity element of the space of symmetric positive definite matrices) is a vector space in which the mean and squared distance computations are simply linear arithmetic operations, linear subspace analysis can be performed in this space. We map covariance matrices into the log-euclidean Riemannian space to obtain log-euclidean covariance matrices which are then unfolded into vectors. A linear subspace analysis of the vectors is then carried out.

A covariance matrix of the image features inside an object block is used to represent this object block. A sequence of N images in which this object block exists yields N covariance matrices $\{C^t \in R^{d\times d}\}_{t=1,2,\ldots,N}$ which constitute a covariance matrix sequence $A \in R^{d\times d\times N}$. In order to ensure that C^t is not a singular matrix, we replace C^t with $C^t + \varepsilon I_d$, where ε is a very small positive constant and I_d is the $d \times d$ identity matrix. By the log-euclidean mapping which is implemented using the matrix logarithmic operation in (4), we transform the covariance matrix sequence A into a log-euclidean covariance matrix sequence: $\alpha = (\log(C^1), \ldots, \log(C^t), \ldots, \log(C^N))$. We unfold the matrix $\log(C^t)$ into a d^2-dimensional column vector v^t ($1 \le t \le N$) in either the row first order or the column first order, i.e., matrix $\log(C^t)$ is represented by the column vector v^t. Then, the log-euclidean unfolding matrix $\Upsilon = (v^1 v^2 \ldots v^t \ldots v^N) \in R^{d^2\times N}$ (the tth column is v^t) is obtained. The merit of unfolding $\log(C^t)$, in contrast to directly unfolding C^t, is that the set of possible values of $\log(C^t)$ forms a vector space in which classic vector space algorithms (e.g., PCA) can be used.

We apply the SVD technique to find the dominant projection subspace of the column space of the log-euclidean unfolding matrix Υ. This subspace is incrementally updated when new data arrive. The mean vector μ is obtained by taking the mean of the column vectors in Υ. We construct a matrix X whose columns are obtained by subtracting μ from each column vector in Υ. The SVD for X is carried out: $X = UDV^T$, producing a $d^2 \times d^2$ matrix U, a $d^2 \times N$ matrix D, and an $N \times N$ matrix V, where U's column vectors are the singular vectors of X, and D is a diagonal matrix containing the singular values. The first k ($k \le N$) largest singular values in D form the $k \times k$ diagonal matrix D^k and the corresponding k columns in U form a $d^2 \times k$ matrix U^k which defines the eigenbasis. The log-euclidean Riemannian subspace is represented by $\{\mu, U^k, D^k\}$.

The incremental SVD technique in [14] and [32] is applied to incrementally update the log-euclidean Riemannian subspace. Let $\{\mu_{t-1}, U_{t-1}^k, D_{t-1}^k\}$ be the previous log-euclidean Riemannian subspace at stage $t-1$. At stage t, a new covariance matrix sequence $A^* \in R^{d\times d\times N^*}$ which contains N^* covariance matrices is added and the new sequence A^* is transformed into a log-euclidean covariance matrix sequence which is then unfolded into a new log-euclidean unfolding matrix $\Upsilon^* \in R^{d^2\times N^*}$. Then, the new subspace $\{\mu_t, U_t^k, D_t^k\}$ at stage t is estimated using $\{\mu_{t-1}, U_{t-1}^k, D_{t-1}^k\}$ and Υ^*. This incremental updating process is outlined as follows:

- **Step 1:** Update the mean vector:

$$\mu_t = \frac{\Gamma \cdot N}{(N^* + \Gamma \cdot N)}\mu_{t-1} + \frac{N^*}{(N^* + \Gamma \cdot N)}\mu^*, \quad (7)$$

  where μ^* is the mean column vector of Υ^* and Γ is a forgetting factor which is used to weight the data streams in order that recent observations are given more weights than historical ones.
- **Step 2:** Let Υ^* have the zero mean: $\Upsilon^* \leftarrow \Upsilon^* - \mu^*$.
- **Step 3:** Construct the combined matrix Υ':

$$\Upsilon' = \left(\Gamma U_{t-1} D_{t-1} | \Upsilon^* | \sqrt{\frac{NN^*}{N+N^*}}(\mu_{t-1} - \mu^*)\right), \quad (8)$$

  where the operation "|" merges its left and right matrices.
- **Step 4:** Compute the QR decomposition for the combined matrix: $\Upsilon' = QR$, producing matrices Q and R.
- **Step 5:** Compute the SVD for matrix R: $R = UDV^T$, producing matrices U, D, and V.
- **Step 6:** Compute singular vectors U_t' and singular values D_t' by

$$U_t' = QU,\ D_t' = D \cdot \sqrt{N/(N^* + \Gamma \cdot N)}. \quad (9)$$

- **Step 7:** The k largest singular values in D_t' are selected to form the diagonal matrix D_t^k, and the k columns corresponding to the elements in D_t^k are chosen from U_t' to form U_t^k.

The above subspace updating algorithm tracks the changes in the column space of the unfolding log-euclidean matrix when new covariance matrix sequences emerge, and identifies the new dominant projection subspace. The vector space properties of the log-euclidean Riemannian space

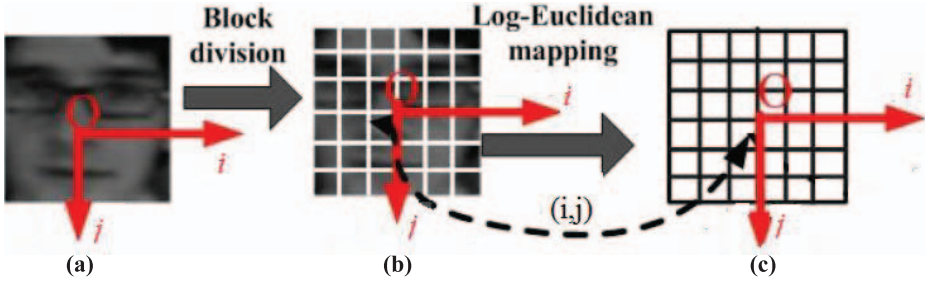


Fig. 1. Division of an object appearance region into blocks. (a) An object image region. (b) The divided image blocks. (c) The array of the log-euclidean covariance matrices (each block corresponds to a matrix).

ensure the effectiveness of the identified dominant projection subspace.

3.4 Likelihood Evaluation

The likelihood of a test sample is evaluated given the learned subspace. Let $C^t \in R^{d\times d}$ be the covariance matrix of features inside the test image region. Let v^t be the column vector obtained by unfolding $\log(C^t)$. Given the learned log-euclidean Riemannian subspace $\{\mu, U, D\}$, the square of the euclidean vector distance between v^t and $\{\mu, U, D\}$ is calculated as the subspace reconstruction error:

$$\mathbb{Z} = \|(v^t - \mu) - U \cdot (U^T \cdot (v^t - \mu))\|^2, \tag{10}$$

where $\|.\|$ is the euclidean vector norm. The likelihood of C^t given $\{\mu, U, D\}$ is evaluated by

$$p(C^t|\mu, U, D) \propto \exp(-\mathbb{Z}). \tag{11}$$

The smaller the $\mathbb{Z}$, the larger the likelihood.

3.5 Theoretical Comparison

In the following, we theoretically compare our algorithm with Kwon et al.'s algorithm [61] and Porikli and Tuzel's algorithm [62]. There are three main differences between Kwon et al.'s algorithm [61] and ours:

- In Kwon et al.'s algorithm, the Riemannian metric is used to approximate the importance density function for optimally sampling 2D affine motion matrices. In our algorithm, the Riemannian metric is applied to construct the Riemannian feature space for appearance feature description.
- The Riemannian metric in Kwon et al.'s algorithm handles 2×2 affine group matrices. The log-euclidean Riemannian metric in our algorithm deals with nonsingular covariance matrices.
- Kwon et al.'s algorithm computes the average offset from the best particle before resampling in the tangent space. The exponential operation projects the offset onto the 2D affine group Riemannian manifold. The desired Riemannian mean after resampling is finally returned. Our algorithm directly uses the arithmetic mean of log-euclidean covariance matrices.

The main differences between Porikli and Tuzel's algorithm [62] and ours are as follows:

- As in Kwon et al.'s algorithm, Porikli and Tuzel use the Riemannian metric to handle affine motion matrices and then build 2D affine motion models.
- Porikli and Tuzel's algorithm computes the geodesic distance between two affine motion matrices using a first order approximation. In our algorithm, the distance between two nonsingular covariance matrices is directly calculated as the logarithm matrix norm without any approximation.

4 Log-Euclidean Block-Division Appearance Model

We divide the object appearance region into nonoverlapping blocks whose log-euclidean Riemannian subspaces are learned and updated incrementally, in order to incorporate more spatial information into the appearance model. Local and global spatial filtering operations are used to tune the likelihoods of the blocks in order that local and global spatial correlations at the block level are contained in the appearance model.

4.1 Appearance Block Division

Given an object appearance sequence $\{F^t\}_{t=1,2,\ldots,N}$, we divide the parallelogram appearance F^t of an object in an image at time t into $m \times n$ blocks. For each block (i, j) $(1 \le i \le m, 1 \le j \le n)$, the covariance matrix feature $C^t_{ij} \in R^{d\times d}$ is extracted using (1) and (2). Covariance matrices $\{C^t_{ij}\}_{t=1,2,\ldots,N}$ corresponding to block (i, j) constitute a covariance matrix sequence $A_{ij} \in R^{d\times d\times N}$. By the log-euclidean mapping using (4), the covariance matrix sequence A_{ij} is transformed into the log-euclidean covariance matrix sequence α_{ij}, which is then unfolded into a log-euclidean matrix $\Upsilon_{ij} \in R^{d^2\times N}$. Fig. 1 illustrates the division of an object appearance region into blocks whose covariance matrices are mapped into the log-euclidean covariance matrices, where "O" is the center of the appearance region. A log-euclidean subspace model $\{\mu_{ij}, U_{ij}, D_{ij}\}$ for Υ_{ij} is learned using our incremental log-euclidean Riemannian subspace learning algorithm.

The square $\mathbb{Z}_{ij}$ of the euclidean vector distance between the block (i, j) of a test sample and the learned log-euclidean subspace model $\{\mu_{ij}, U_{ij}, D_{ij}\}$ is determined by (10), and then the likelihood p_{ij} for block (i, j) in the test sample is estimated using (11). Finally, a matrix $M = (p_{ij})_{m\times n} \in R^{m\times n}$ is obtained for all the blocks.

4.2 Local Spatial Filtering

In order to remedy occasional inaccurate estimation of the likelihoods for a very small fraction of the blocks, the matrix M is filtered to produce a new matrix $M_l = (p^l_{ij})_{m\times n} \in R^{m\times n}$

based on the prior knowledge that if the likelihoods of the blocks neighboring to a given block are large, then the likelihood of the given block is also likely to be large. This local spatial filtering is formulated as

$$p_{ij}^{l} \propto p_{ij} \cdot \exp\left(\frac{N_{ij}^{+} - N_{ij}^{-}}{\sigma_l}\right), \tag{12}$$

where N_{ij}^{+} is the number of block (i,j)'s neighboring blocks whose likelihoods are not less than p_{ij}, N_{ij}^{-} is the number of block (i,j)'s neighboring blocks whose likelihoods are less than p_{ij}, and σ_l is a positive scaling factor. The exponential function in (12) is a local spatial filtering factor which measures the influence of the neighboring blocks on the given block. If N_{ij}^{+} is smaller than N_{ij}^{-}, the factor decreases the likelihood of block (i,j), and the larger the difference between N_{ij}^{+} and N_{ij}^{-}, the more the likelihood is decreased; otherwise the likelihood of block (i,j) is increased. Although the p_{ij} values are from different subspace projections, they are comparable. The reasons for this include the following points:

- As shown in (10) and (11), the likelihood is a similarity measurement which is unaffected by changes in the mean.
- The sizes of all the blocks in an object appearance region are the same.
- The dimensions of the covariance matrices describing the blocks are the same, and the definitions of each corresponding element in all the covariance matrices are the same.
- The order in which the log-euclidean covariance matrices are unfolded is the same in every case.
- The dimensions of the dominant projection subspaces for all the blocks are the same.

4.3 Global Spatial Filtering

Global spatial filtering is carried out based on the prior knowledge that the blocks nearer to the center of the appearance region have more dependable and more stable likelihoods, and the likelihoods for boundary blocks are prone to being influenced by the exterior of the appearance region. A spatial Gaussian kernel is used to globally filter the matrix $M_l = (p_{ij}^{l})_{m\times n}$ to produce a new matrix $M_g = (p_{ij}^{g}) \in R^{m\times n}$:

$$p_{ij}^{g} \propto p_{ij}^{l} \cdot \exp\left(-\frac{(x_{ij}-x_o)^2 + (y_{ij}-y_o)^2}{2\sigma_g^2}\right), \tag{13}$$

where x_{ij} and y_{ij} are the positional coordinates of block (i,j), x_o and y_o are the positional coordinates of the center of the appearance region, and σ_g is a scaling factor. The nearer the block is to the center of the appearance region, the more weight it is given.

4.4 Observation Likelihood

The overall likelihood $p_{overall}$ of a candidate object appearance region given the learned block-division appearance model positively correlates with the product of all the corresponding block-specific likelihoods after the local and global spatial filtering:

$$p_{overall} \ddot{\propto} \prod_{i=1}^{m}\prod_{j=1}^{n} p_{ij}^{g}, \tag{14}$$

where the symbol $\ddot{\propto}$ means that the left-hand side and the right-hand side of (14) either increase together or decrease together. The log version of (14) is used to transform the product of likelihoods to the sum of log likelihoods

$$\log(p_{overall}) \ddot{\propto} \sum_{i=1}^{m}\sum_{j=1}^{n} \log(p_{ij}^{g}). \tag{15}$$

4.5 Remark

Local and global spatial correlations of object appearance blocks are represented via local and global spatial filtering. Local spatial relations between the values of the pixels in each block and the temporal correlations between the image regions corresponding to the same block in the image sequence are reflected in the log-euclidean Riemannian subspace of the block. This makes our appearance model robust to environmental changes.

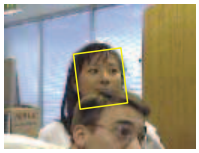
5 Single Object Tracking

The object motion between two consecutive frames especially for videos captured by nonstationary cameras is usually modeled by affine warping, which is defined by parameters $(x_t, y_t, \eta_t, s_t, \beta_t, \phi_t)$, where x_t, y_t, η_t, s_t, β_t, and ϕ_t denote the x, y translations, the rotation angle, the scale, the aspect ratio, and the skew direction, respectively [14]. The state X_t of a tracked object in frame t is described by the affine motion parameters $(x_t, y_t, \eta_t, s_t, \beta_t, \phi_t)$. In the tracking process, an observation model $p(O_t|X_t)$ and a dynamic model $p(X_t|X_{t-1})$ are used to obtain the optimal object state in frame t given its state in frame $t-1$, where O_t is the observation in frame t. In our algorithm, the observation model $p(O_t|X_t)$ reflects the similarity between the image region specified by X_t and the learned log-euclidean block-division appearance model, and it is defined as: $p(O_t|X_t) \propto p_{overall}$, where $p_{overall}$ is defined in (14) and (15). A Gaussian distribution [14] with a diagonal covariance matrix with diagonal elements σ_x^2, σ_y^2, σ_η^2, σ_s^2, σ_β^2, and σ_ϕ^2 is employed to model the state transition distribution $p(X_t|X_{t-1})$. A standard particle filtering approach [3] is applied to estimate the optimal state (please refer to [14] for details). The image region associated with the optimal state is used to incrementally update the block-related log-euclidean appearance model.

During tracking, each image region is warped into a normalized rectangular region [14] using the estimated affine parameters. Covariance matrix computation, subspace projection, likelihood evaluation, subspace update, and smoothing with Gaussian kernel are carried out on the normalized rectangular region.

6 Multi-Object Tracking with Occlusion Reasoning

Our task of tracking multi-objects is, especially for videos captured by nonstationary cameras, to localize multiple moving objects even when tracked objects are occluded by another tracked objects, and to explicitly determine their



0.5533	0.2974	0.7509	0.3832
0.4002	0.2475	0.4501	0.5745
1.4403	0.5524	0.2771	1.4751
2.5272	3.0886	3.6015	5.4790

(a) (b)

Fig. 2. Reconstruction errors of blocks. (a) Original image, where a man's face occludes the bottom part of a girl's face. (b) Values of reconstruction errors of blocks corresponding to the girl's face.

occlusion relations. Our algorithm for multi-object tracking is an extension of our single object tracking algorithm. When there are no occlusions in the previous frame, the extent of any occlusion in the current frame is not large and the single object tracking algorithm is robust enough to track the objects accurately. So, under the condition that there are no occlusions in the previous frame, each of the objects in the current frame can be tracked using the single object tracking algorithm. If there is occlusion in the previous frame, then each object is tracked using one particle filter as before, except that the appearance subspaces of the blocks with large appearance changes are unchanged, while those for the remaining blocks are updated in the current frame.

In the following, we describe occlusion detection, subspace reconstruction error changes during occlusion, appearance model updating during occlusion, occlusion reasoning, and appearance and disappearance handling.

6.1 Occlusion Detection

Occlusion existence is deduced from the tracking results. Given the optimal state of an object, the object is represented by a parallelogram which is determined by its center coordinates, height, width, and skew angle. If parallelograms of two objects intersect, then there is an occlusion between the two objects.

6.2 Subspace Reconstruction Error Changes during Occlusions

If a block is occluded, then the subspace reconstruction error (10) of its log-euclidean unfolded covariance is extremely high due to drastic appearance changes resulting from the occlusion. The effects of occlusion on the reconstruction errors are illustrated in Fig. 2, where Fig. 2a shows an exemplar video frame in which the bottom part of a girl's face is occluded by a man's face, and Fig. 2b shows the reconstruction errors of blocks of the girl's face. As shown in Fig. 2b, blocks corresponding to the occluded part of the girl's face have much larger reconstruction errors than the unoccluded blocks. The blocks with reconstruction errors less than a given threshold $\mathbb{Z}_{threshold}$ are used to evaluate the likelihood. Equation (15) is replaced by

$$\log(p_{overall}) \dddot{\propto} \frac{\sum_{i\in\Omega} \log(p_i)}{|\Omega|}, \tag{16}$$

where Ω is the set of blocks with reconstruction errors less than $\mathbb{Z}_{threshold}$ and $|\Omega|$ is the number of blocks in Ω.

6.3 Appearance Model Updating during Occlusions

If the appearance variations caused by large occlusions are learned by the appearance model, large appearance errors from occluded blocks may result in inaccurate or incorrect tracking results. During occlusions, we only update the subspaces for blocks whose reconstruction errors are less than the threshold $\mathbb{Z}_{threshold}$. The subspaces for blocks whose reconstruction errors exceed the threshold remain unchanged. In this way, the appearance variations in blocks which are not occluded are learned effectively. As a result, the appearance model can be updated even in the presence of occlusions.

6.4 Occlusion Reasoning

The task of occlusion reasoning is to determine the occlusion relations between objects. A number of sophisticated probabilistic mechanisms have been developed for occlusion reasoning. For example, Sudderth et al. [63] augment a nonparametric belief propagation algorithm to infer variables of self-occlusions between the fingers of a hand. Zhang et al. [64] handle long-term occlusions by adding occlusion nodes and constraints to a network which describes the data associations. Wang et al. [65] carry out object tracking with occlusion reasoning using rigorous visibility modeling within a Markov random field. Herbst et al. [66] reason about the depth ordering of objects in a scene and their occlusion relations. Gay-Bellile et al. [67] construct a probability self-occlusion map to carry out image-based nonrigid registration. However, the current probabilistic mechanisms for occlusion reasoning are very complicated. In practice, assumptions or simplifications are always utilized to reduce the search space.

We found that, given the states of the objects, their occlusion relations are fixed. So, the occlusion relations between objects are dependent on the current states of the objects and independent of their previous occlusion relations. Instead of sophisticated probabilistic mechanisms, we propose a simple and intuitive mechanism which deduces the occlusion relations from the current states of the objects and the current observations, using the observation model which corresponds to subspace reconstruction errors. We utilize variations of reconstruction errors of blocks to find which objects are occluded. When it is detected that two objects a and b are involved in occlusion, the overlapped region between the parallelograms corresponding to these two objects is segmented. For each of these two parallelograms, the blocks within this overlapped region and the blocks overlapped with this region are found. Let $\overline{\mathbb{Z}}_{o\subset a}$ and $\overline{\mathbb{Z}}_{o\subset b}$ be, respectively, the average reconstruction errors of such overlapped blocks in objects a and b. Let $\overline{\mathbb{Z}}_a$ and $\overline{\mathbb{Z}}_b$ be, respectively, the average reconstruction errors of all the blocks in objects a and b. Let $\phi_{(a,b)}$ represent the occlusion relation between objects a and b:

$$\phi_{(a,b)} = \begin{cases} -1, & if \quad a \quad occludes \quad b, \\ 0, & if \quad no \quad occlusion, \\ 1, & if \quad b \quad occludes \quad a. \end{cases} \tag{17}$$

The occlusion relation between objects a and b at frame t is determined by

$$\phi^t_{(a,b)} = \begin{cases} -1, & if \ \ \overline{\mathbb{Z}}_{o\subset a} - \overline{\mathbb{Z}}_a < \overline{\mathbb{Z}}_{o\subset b} - \overline{\mathbb{Z}}_b, \\ 1, & if \ \ \overline{\mathbb{Z}}_{o\subset a} - \overline{\mathbb{Z}}_a > \overline{\mathbb{Z}}_{o\subset b} - \overline{\mathbb{Z}}_b, \\ \phi^{t-1}_{(a,b)}, & otherwise. \end{cases} \tag{18}$$

To suppress the effects of noise, any occlusion relation which lasts for less than three consecutive frames is omitted and it is replaced with the previous relation. Using this strategy, occlusion relations between more than two people who occlude each other can be reasoned about.

6.5 Appearance and Disappearance

We handle the appearance and disappearance of objects for both stationary and nonstationary cameras.

For videos taken by stationary cameras, background subtraction is used to extract motion regions. The extracted motion regions in the current frame are compared with the motion regions in the previous frame according to their positions and appearances. If the current frame contains a motion region which does not correspond to any motion region in the previous frame, then a new object is detected as a tracked object. The appearance model for the object is initialized according to the new motion region. A particle filter is initialized according to a prior probability distribution on the state vector of the new tracked object and the prior distribution is assumed to be a Gaussian distribution. If a motion region gradually becomes smaller to the point where it can be ignored, then object disappearance occurs. The particle filter corresponding to the object is removed.

For videos taken by nonstationary cameras, object detection methods should be introduced for handling object entering. There are a number of face or pedestrian detection algorithms [35], [57], [58], [59], [60] with low computational complexity. However, for these algorithms, mistaken detections are frequent. In this paper, we use the estimated optical flows with ego-motion compensation to find motion regions in which pixels not only have large optical flow magnitudes, but also coherent optical flow directions. Candidate motion regions are defined by moving a rectangle over the image and changing its size [19]. However, the found motion regions are usually inaccurate. Then, we use the boundaries of these detected motion regions as the objects' initial contours, which are then evolved using the region-based level set contour evolution algorithm in [19] to obtain the final object contours. The region-based contour algorithm can evolve a simple and rough initial contour to closely fit the edge of the object. We detect objects in the bounding boxes of the contours. In this way, objects such as faces [35] can be accurately detected and located. The optical flow estimation is slow. We make some assumptions to increase the speed. For example, we assume that the motion regions corresponding to an object entering are connected with the image boundaries. In this way, the area that is required to search for the new motion regions is reduced.

Object disappearance for videos taken by nonstationary cameras is handled by checking the reconstruction errors. If the reconstruction error of the object appearance gradually becomes larger and there is no other object which occludes the object, then it is determined that the object is disappearing.

6.6 Remark and Extension

Traditional centralized methods for multi-object tracking with occlusion handling carry out particle filtering in a joint state space for all the objects, i.e., the state vectors of all the objects are concatenated into a vector, and a particle is defined for the objects. Due to the high dimension of the joint state vector, the computational cost is very high. Our algorithm handles occlusions according to the reconstruction errors in object appearance blocks. This ensures that our algorithm can track individual objects in their own state spaces during occlusions, i.e., there is one particle filter for each object. This makes the state inference more computationally efficient, in contrast to centralized particle filters.

Our method for occlusion detection and handling at the block level can be used for single object tracking when the tracked object is occluded by untracked moving objects or scene elements, e.g., static objects: Block-wise appearance outliers of the object are monitored and the subspaces of the unoccluded blocks are updated online. Although our single object tracking algorithm without special occlusion handling is robust to partial occlusions and fast illumination changes due to the log-euclidean Riemannian appearance model, introducing occlusion detection and handling into single object tracking can increase the tracking accuracy during occlusions or fast illumination changes while more runtime is required.

7 Experiments

In order to evaluate the performance of the proposed tracking algorithms, experiments were carried out using Matlab on the Windows XP platform. The experiments covered 10 challenging videos, five of which were taken by nonstationary cameras and five of which were taken by stationary cameras. The experiments on these videos consisted of four face tracking examples and six examples of tracking pedestrians. For the face tracking examples, tracking was initialized using the face detection algorithm [35]. For the videos captured by stationary cameras, tracking was initialized using background subtraction [34]. For tracking a pedestrian in the video taken by a nonstationary camera, tracking was initialized using optical flow region analysis [19].

The tuning parameters in our algorithm are set empirically in the experiments. For example, the number of blocks was chosen to maintain both the accuracy and robustness of the tracking. If a larger number of blocks is used, the object can be tracked more accurately when the changes in the object appearance are small or moderate, but when there are large appearance changes, the tracker is more likely to drift. In our experiments, we found that when each object region was uniformly divided into 36 blocks, the objects in all the examples are successfully and accurately tracked. But when fewer blocks are used, some results with unacceptable accuracy are obtained. So, it is appropriate to set the number of blocks equal to 36. We set the dimension k of the subspace according to the reconstruction quality which is defined as the ratio of the sum of the k largest singular values in D'_t defined in (9) to the sum of all the singular values in D'_t. The number k is the least number such that the reconstruction quality is above 98 percent. The number of particles was set to 200 for each object in the absence of occlusions, and set to 500 in the presence of occlusions. It is found that when fewer particles are used, there are frames for which the results are obviously inaccurate, and the runtime is only slightly decreased. The log-euclidean block-division appearance model was updated every three frames. The six diagonal elements (σ_x^2, σ_y^2, σ_η^2, σ_s^2, σ_β^2, σ_φ^2) in the dynamic model were given the values of 52, 52, 0.032, 0.032, 0.0052, and 0.0012, respectively. The

Fig. 3. Example 1: Tracking a face with drastic illumination changes. From left to right, the frame numbers are 140, 150, 158, 174, and 192, respectively: the first, second, third, fourth, fifth, and sixth rows are, respectively, the results from our algorithm, the algorithm based on the affine-invariant metric, the vector subspace-based algorithm, Jepson's algorithm, Yu's algorithm, and the MIL-based algorithm.

forgetting factor Γ in (7), (8), and (9) was set to 0.99. The factor σ_l in (12) was set to 8. The factor σ_g in (13) was set to 3.9. The occlusion threshold for an object at the current frame is set to three times the mean of the reconstruction errors of its unoccluded blocks in the previous three frames.

In the experiments, we compared our single object tracking algorithm with the following five state-of-the-art representative and typical tracking algorithms

- The algorithm based on the affine-invariant Riemannian metric [24]: A baseline for our algorithm.
- The vector subspace-based algorithm [14]: Also a baseline for our algorithm.
- Jepson et al.'s algorithm [5]: The most typical one which learns the appearance model online.
- Yu and Wu's algorithm [7]: The most typical one for part-based appearance modeling for visual tracking, in contrast to the above competing algorithms which use holistic appearance representations.
- The multiple instance learning (MIL)-based algorithm [36]: The typical one which can deal effectively with accumulation of small tracking inaccuracies in consecutive frames.

The algorithm based on the affine-invariant Riemannian metric [24] was extended to track multi-objects with occlusion reasoning according to the principles of handling occlusions in our multi-object tracking algorithm. Then, our multi-object tracking algorithm was compared with the extended algorithm. We also compared our multi-object tracking algorithm with Yang et al.'s algorithm [55], which is a typical appearance-based multi-object tracking algorithm.

7.1 Example 1

The video for this archetypal example is available on http://www.cs.toronto.edu/~dross/ivt/. This video was recorded with a nonstationary camera. It consists of 8-bit grayscale images in which a man moves in a dark outdoor scene with drastically varying lighting conditions.

In this example, the face of the man is tracked. Fig. 3 shows the results for this example. It is shown that our algorithm tracks the object successfully in all 497 frames, even in poor lighting conditions. In a few frames in which the face moves rapidly, there are some deviations between the localized positions of the person and the true positions. In comparison, the algorithm based on the affine-invariant Riemannian metric loses the track in many frames. The tracking using the vector subspace-based algorithm breaks down after frame 300 when there is a large variation in illumination and a pose change. Jepson's algorithm loses track from frame 316 to frame 372, after which the track is recovered. It loses the track again from frame 465 onward. Yu's algorithm overall continuously tracks the face, but in a number of frames the results are inaccurate. The MIL-based algorithm loses the track from frame 195 onward because its use of Haar-like features makes it sensitive to changes in illumination.

In each frame, we manually label four benchmark points corresponding to the four corners of the image region of the face. These benchmark points characterize the location of the face and are used to evaluate the accuracy of the results of the tracking algorithms. During the tracking, four validation points corresponding to the four benchmark points were obtained in each frame according to the object's

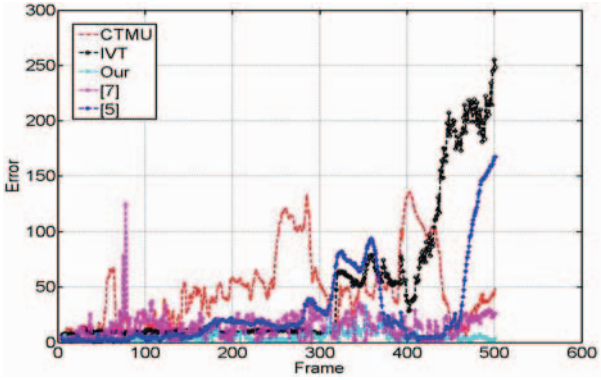


Fig. 4. The quantitative comparison between our algorithm and the competing algorithms for Example 1. Our algorithm corresponds to the cyan curve, the algorithm based on the affine-invariant Riemannian metric the red curve, the vector subspace-based algorithm the black curve, Jepson's algorithm the blue curve, and Yu's algorithm the magenta curve.

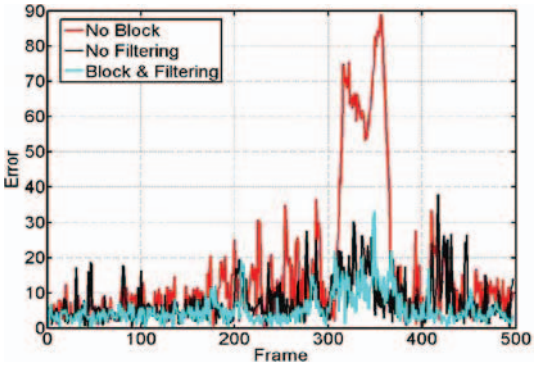


Fig. 6. The quantitative comparison results for Example 1 with and without nonoverlapping blocks or local and global filtering: The red, black, and blue curves correspond, respectively, to the results without nonoverlapping blocks, the results with nonoverlapping blocks but without local and global filtering, and the results with nonoverlapping blocks and local and global filtering.

affine motion parameters. In each frame, the location deviation (also called the tracking error) between the validation points and the benchmark points is defined as the average of the pixel distances between each validation point and its corresponding benchmark point. This tracking error is a quantitative measure of the tracking accuracy. Fig. 4 shows the tracking error curves of our algorithm and the competing algorithms. It is seen that the tracking errors of our algorithm are lower than the errors of the competing algorithms. It is noted that Jepson's algorithm and Yu's algorithm are much faster than ours, and the other competing algorithms have similar runtimes to ours. As the affine parameters used to represent the state of the object in our algorithm are not used in the MIL-based algorithm, a quantitative accuracy comparison between our algorithm and the MIL-based algorithm is omitted.

Fig. 5 shows the results obtained by omitting nonoverlapping blocks or local and global filtering from our algorithm. Fig. 6 shows tracking error curves with and without nonoverlapping blocks or local and global filtering. The mean errors without nonoverlapping blocks, with nonoverlapping blocks but without local and global filtering, and with nonoverlapping blocks and local and global filtering, are 15.25, 7.47, and 5.29, respectively. It is apparent that the tracking results with nonoverlapping blocks and local and global filtering are overall more accurate than the results with nonoverlapping blocks but without local and global filtering. The tracking results without nonoverlapping blocks are much less accurate than the results with nonoverlapping blocks but without local and global filtering. So, the division of the object appearance into blocks is more important than the local and global filtering.

As stated in Section 6.6, our blockwise occlusion monitoring method can trigger in case of fast illumination changes. Fig. 7 shows the results of tracking the face using our occlusion handling method. It is seen our occlusion monitoring method successfully handles fast illumination changes throughout the video. Fig. 8 quantitatively compares the results with and without occlusion monitoring. The mean tracking error without occlusion handling is 5.31 pixels per frame and that with occlusion handling is 4.86 pixels per frame. Occlusion handling obtains more accurate results.

7.2 Example 2

The video for this example is in the PETS 2004 database, which is an open database for research on tracking. It is available at http://homepages.inf.ed.ac.uk/rbf/CAVIAR/. The video, which was taken from a stationary camera, is composed of 24-bit RGB color images. In this video, a pedestrian moves along a corridor. In the middle of the video, his body is severely occluded by the bodies of two other pedestrians. Fig. 9 shows the results of tracking this pedestrian. It is apparent that our algorithm succeeds in tracking the pedestrian in all the frames, whether or not occlusion handling is used. The algorithm based on affine-invariant Riemannian metric fails in several frames. Jepson's algorithm does not correctly track the pedestrian after he is occluded by another pedestrian with similar clothing. Yu's algorithm continuously tracks the occluded pedestrian, but in some frames the results are inaccurate.

Fig. 5. The results for Example 1 without nonoverlapping blocks or local and global filtering. The first row shows the results without nonoverlapping blocks. The second row shows the results with nonoverlapping blocks but without local and global filtering.

Fig. 7. Example 1: Tracking the face of the boy using occlusion handling.

7.3 Example 3

The video for this example was recorded with a nonstationary camera. It consists of 8-bit grayscale images. In this video, a man walks from left to right on a bright road, and his body pose varies over time. In the middle of the video, there are drastic motion and pose changes: bowing down to reach the ground and standing up back again. Fig. 10 shows the results of tracking the human body. It is apparent that our algorithm tracks the target successfully even with drastic pose and motion changes. Both the algorithm based on the affine-invariant Riemannian metric and the vector subspace-based algorithm lose track during the drastic pose and motion changes.

7.4 Example 4

The video used in this example was chosen from the open PETS2001 database. It was taken from a stationary camera and consists of 8-bit grayscale images. In this video, a pedestrian moves down a road in a dark scene. The pedestrian has a very small apparent size in all the frames. Fig. 11 shows the results of tracking the pedestrian. It can be seen that our algorithm succeeds in tracking the small object throughout the video. The algorithm based on the affine-invariant Riemannian metric does not track the small object accurately in many frames. The vector subspace-based algorithm loses the track from frame 503 onward.

7.5 Example 5

This example is widely used for testing face tracking algorithms. The video is available at http://www.cs.toronto.edu/vis/projects/dudekfaceSequence.html. It was recorded with a nonstationary camera and consists of 8-bit grayscale images. In this video, a man who sits in a chair changes his pose and facial expression over time and from time to time his hand occludes his face.

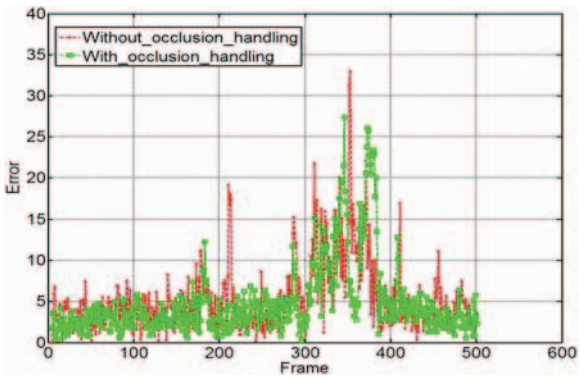


Fig. 8. The quantitative comparison between the results with and without occlusion handling for Example 1: The green and red curves correspond to the results with and without occlusion handling, respectively.

Fig. 12 shows the results of tracking the face of the man. It is seen that our algorithm (without occlusion handling) tracks the object accurately. The algorithm based on the affine-invariant Riemannian metric loses the track in many frames. The vector subspace-based algorithm continually tracks the face, but the occlusion which occurs from frame 104 to 108 produces a slight tracker drift away from the face and this drift causes the results of the following tracking to only partly overlap the face. The tracking results of Jepson' algorithm, Yu's algorithm, and the MIL-based algorithm are not accurate in many frames.

In this example, each frame contains seven manually labeled benchmark points [14]. Fig. 13 shows the tracking error curves of our algorithm and the competing algorithms. It is seen that the tracking errors of our algorithm are always lower than the tracking errors of all the competing algorithms.

Fig. 14 shows the results obtained by omitting nonoverlapping blocks, or local and global filtering from our algorithm. Fig. 15 shows the tracking error curves with and without nonoverlapping blocks or local and global filtering. The mean errors without nonoverlapping blocks, with nonoverlapping blocks but without local and global filtering, and with nonoverlapping blocks and local and global filtering, are 54.57, 14.2, and 12.14, respectively. It is apparent that the algorithm with nonoverlapping blocks and local and global filtering obtains more accurate results than the algorithm without nonoverlapping blocks and the algorithm with nonoverlapping blocks but without local and global filtering.

7.6 Example 6

The video for this example is available at http://vision.stanford.edu/~birch/headtracker/. This is a widely used face tracking example. The video was recorded with a nonstationary camera. A girl changes her facial pose over time under varying lighting conditions. In the middle of the video, the girl's face is severely occluded by a man's head.

For this example, the 8-bit grayscale image sequence and the 24-bit RGB color image sequence were both considered for tracking the girl's face. The tracking results for the grayscale image sequence are shown in Fig. 16. It can be seen that our algorithm tracks the object successfully in the case of severe occlusions, whether or not special occlusion handling is used, and the results with occlusion handling are more accurate than the results without occlusion handling. The algorithm based on the affine-invariant Riemannian metric loses the track after severe occlusions or does not track the face accurately. Jepson's algorithm loses track for a few frames when the girl's face is occluded by the man's face. Both Yu's algorithm and the MIL-based algorithm continuously track the face, but the results are not accurate in some frames. The tracking results for the color image sequence are shown in Fig. 17. It can be seen that our

Fig. 9. Example 2: Tracking an occluded human body. From left to right, the frame numbers are 22, 26, 28, 32, and 35, respectively. The first and second rows show the results of our algorithm without and with occlusion handling, respectively, the third, fourth, and fifth rows show, respectively, the results of the algorithm based on the affine-invariant Riemannian metric, Jepson's algorithm, and Yu's algorithm.

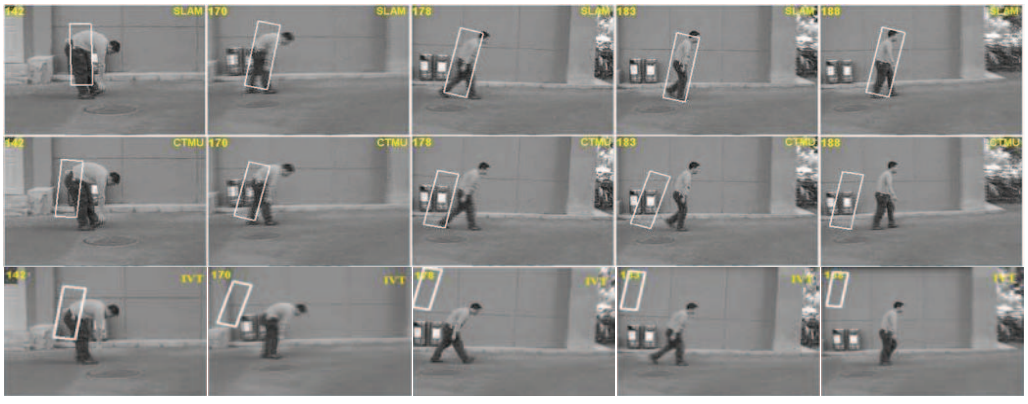


Fig. 10. Example 3: Tracking a person with drastic body pose variations. From left to right, the frame numbers are 142, 170, 178, 183, and 188, respectively, where the first, second, and third rows are, respectively, the results of our algorithm, the algorithm based on the affine-invariant Riemannian metric, and the vector subspace-based algorithm.

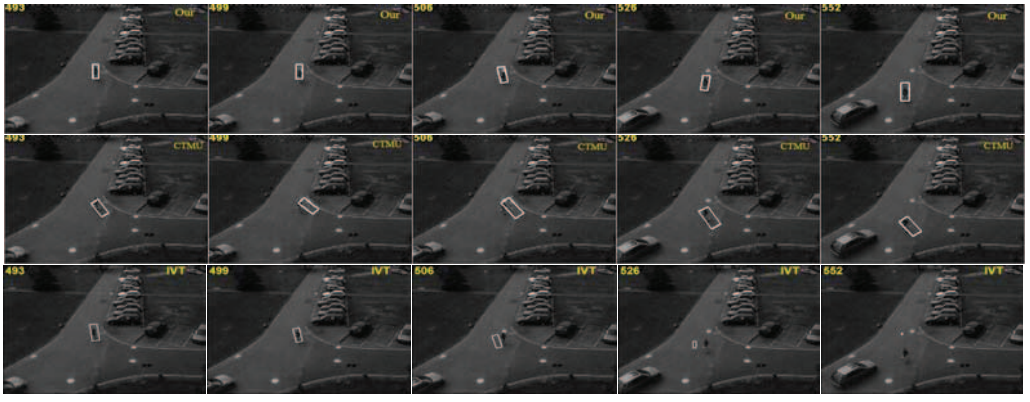


Fig. 11. Example 4: Tracking a pedestrian with a small apparent size in a dark scene. From left to right, the frame numbers are 493, 499, 506, 526, and 552, respectively, where the first row is the results of our algorithm, the second row is the results of the algorithm based on the affine-invariant Riemannian metric, and the third row is the results of the vector subspace-based algorithm.

Fig. 12. Example 5: Tracking a face under partial occlusions and pose variations. From left to right, the frame numbers are 49, 108, 117, 185, and 289, respectively: the first, second, third, fourth, fifth and sixth rows are, respectively, the results of our algorithm, the algorithm based on the affine-invariant Riemannian metric, the vector subspace-based algorithm, Jepson's algorithm, Yu's algorithm, and the MIL-based algorithm.

algorithm tracks the object successfully and accurately in all the frames while the algorithm based on the affine-invariant Riemannian metric loses track during severe occlusions.

We also use this color sequence to test multi-object tracking with occlusion reasoning. Fig. 18 shows the results of simultaneously tracking both the two faces using our algorithm and the algorithm based on the affine-invariant Riemannian metric. It is seen that, overall, both the faces are successfully tracked by our algorithm, but the algorithm based on the affine-invariant Riemannian metric loses track of the severely occluded face. Fig. 19 shows the recovered occlusion relations obtained using our algorithm, where the x-coordinate is the frame number and the y-coordinate is the occlusion relation where "-1" indicates that the man's face occludes the girl's face, "0" means that there is no occlusion between them, and "1" indicates that the girl's face occludes the man's face. It is seen that the occlusion reasoning result is correct when the man's face occludes the girl's face. Fig. 20 shows the variance curve of the sum of reconstruction errors for all the blocks for the girl's face over time. The two peaks in the curve correspond to the two stages that the man's face severely occludes the girl's face.

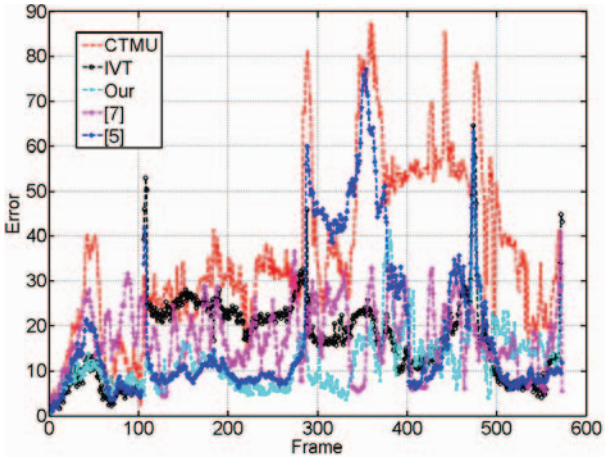


Fig. 13. The quantitative comparison between our algorithm and the competing algorithms for Example 5. Our algorithm corresponds to the cyan curve, the algorithm based on the affine-invariant Riemannian metric the red curve, the vector subspace-based algorithm the black curve, Jepson's algorithm the blue curve, and Yu's algorithm the magenta curve.

It is noted that from frame 439 to frame 455, the man's face gradually disappears from the scene and then after frame 455 his face appears in the scene again. The disappearance and reappearance are successfully detected and handled by our algorithm. Fig. 21 shows the process for handling the face entrance in frame 457. It is seen that tracking of the face is successfully initialized when the face reappears in the scene. The occlusion relations are correctly deduced even during occlusion, disappearance, and reappearance.

7.7 Example 7

The video for this example was taken by a nonstationary camera. It is composed of 24-bit color frames. In several frames, one face is nearly completely occluded by the other face. There also exist pose variations in the video. Fig. 22 shows the results of tracking these two faces using our algorithm and the algorithm based on the affine-invariant

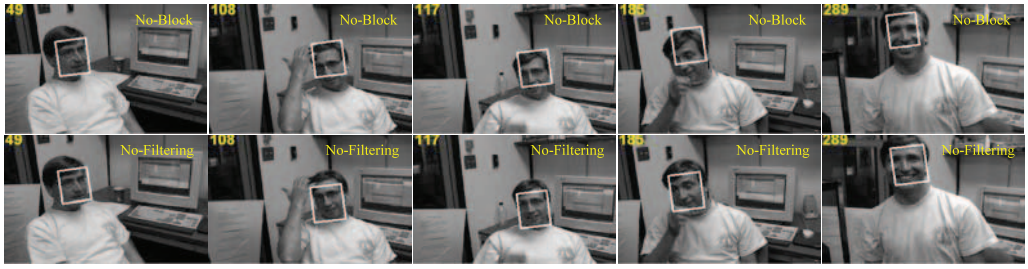


Fig. 14. The results for Example 5 without nonoverlapping blocks or local and global filtering. The first row shows the results without nonoverlapping blocks and the second row shows the results with nonoverlapping blocks but without local and global filtering.

Riemannian metric. The results show that our algorithm successfully tracks both the faces, especially when the back face is severely occluded by the front face from frame 10294 to frame 10338. The algorithm based on the affine-invariant Riemannian metric continuously tracks the two faces, but its tracking accuracy is lower than our algorithm's. As shown in Fig. 23, the occlusion relations between these two faces are recovered correctly by our algorithm.

7.8 Example 8

The color video for this example is in the PETS 2004 database. It shows a shopping center. In the video, there is occlusion between two pedestrians. During the occlusion, both the occluded and the unoccluded pedestrians turn their bodies and thus undergo gradual appearance changes. This example is used to illustrate the effectiveness of our strategy for updating object appearance models in the presence of occlusions. Fig. 24 shows the tracking results using our appearance model updating strategy and the conventional updating strategy which keeps the appearance model unchanged in the presence of occlusions. The conventional strategy loses track of the occluded pedestrian during the occlusion and does not recover the track after the occlusion because the appearance changes of the occluded person during the occlusion are not recorded. In contrast, our strategy successfully tracks the two people throughout the occlusion. It is noted that one of the pedestrians enters the scene from a door in the first several frames. The entrance was successfully detected using the background subtraction.

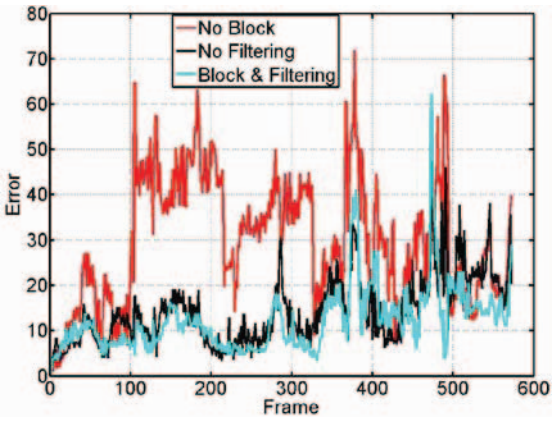


Fig. 15. The quantitative comparison results for Example 5 with and without nonoverlapping blocks or local and global filtering: The red, black, and blue curves correspond, respectively, to the results without nonoverlapping blocks, the results with nonoverlapping blocks but without local and global filtering, and the results with nonoverlapping blocks and local and global filtering.

7.9 Example 9

In this video, a male pedestrian and a female pedestrian walk together. The woman is occluded by the man between frames 192 and 257. During the occlusion, the woman is sometimes completely occluded by the man, i.e., she is nearly invisible. There also exists nonplanar rotation in that the man turns his body from time to time.

Fig. 25 shows the results of tracking these two pedestrians. It is seen that our algorithm successfully tracks the two pedestrians, the algorithm based on the affine-invariant Riemannian metric loses track of the occluded woman, and Yang's algorithm [55] fails to track the two pedestrians during the occlusion. Fig. 26 shows that the occlusion relations recovered using our algorithm are correct.

There are ground truth data for this video. A quantitative evaluation of the tracking accuracy is conducted, using the following criteria:

- the number of successfully tracked frames (tracking is considered to be successful if the estimated box's center is in the box of the ground truth),
- the mean of tracking errors in all the frames.

The quantitative results in Table 1 show that our method outperforms both Yang's algorithm and the algorithm based on the affine-invariant Riemannian metric.

7.10 Example 10

In the video for this example, mutual occlusions occur between three pedestrians. Fig. 27 shows the results for tracking these three pedestrians, using our algorithm and Yang's algorithm. Although the three pedestrians overlap each other, our algorithm still tracks them robustly and recovers the true occlusion relations, while Yang's algorithm loses the tracks of two of them. In the second image in the first row, the pedestrian with a blue bounding box is not tracked so accurately. The reason is that this pedestrian is occluded in the initial frame and, as a result, her appearance model was not learned accurately.

There are ground truth data for this video. Table 2 shows the results of quantitative comparisons between our algorithm and Yang's algorithm for tracking these three pedestrians (persons A, B, and C). It is seen that the mean tracking error of Yang's algorithm for person A is very large. This is because Yang's algorithm quickly loses the track of this person. From the table, it is apparent that the results of our algorithm are more accurate than the results of Yang's algorithm.

Fig. 16. Example 6: Tracking a face under severe occlusions in the grayscale sequence. From left to right, the frame numbers are 153, 162, 165, 180, and 187, respectively: The first and second rows are the results of our algorithm without and with occlusion handling, respectively, the third, fourth, fifth, and sixth rows are, respectively, the results of the algorithm based on the affine-invariant Riemannian metric, Jepson's algorithm, Yu's algorithm, and the MIL-based algorithm.

Fig. 17. Example 6: Tracking an occluded face in the color sequence. From left to right, the frame numbers are 158, 160, 162, 168, and 189, respectively. The first row shows the results of our algorithm, and the second row shows the results of the algorithm based on the affine-invariant Riemannian metric.

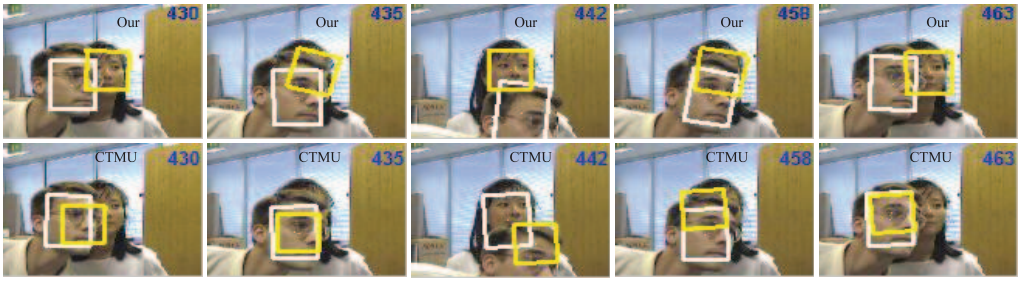


Fig. 18. Example 6: Tracking two faces simultaneously in the color sequence. From left to right, the frame numbers are 430, 435, 442, 458, and 463, respectively. The first row shows the results of our algorithm, and the second row shows the results of the algorithm based on the affine-invariant Riemannian metric.

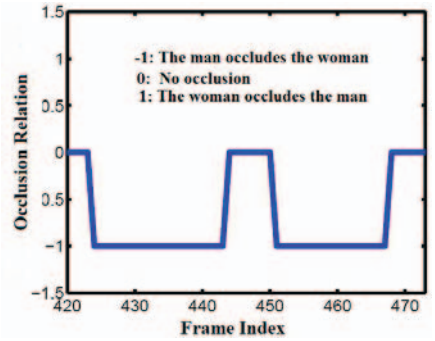


Fig. 19. Recovered occlusion relations for Example 6.

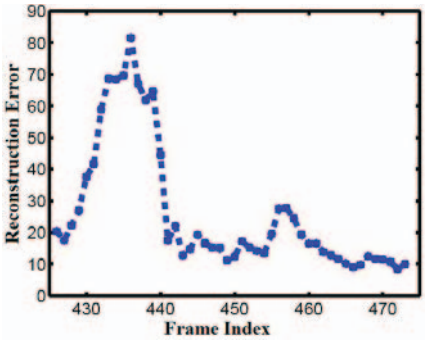


Fig. 20. Variance of the sum of reconstruction errors for all the blocks for the girl's face.

7.11 Analysis of Results

In the following, we analyze the reasons why our algorithm obtains more accurate results than the competing algorithms.

We compare the following points between our algorithm and the algorithm based on the affine-invariant Riemannian metric:

- The log-euclidean block-division appearance model in our algorithm captures both the global and local spatial properties of object appearance at the block level. Even if the subspace information on some blocks is partially lost or drastically varies, our appearance model recovers the missing information using the cues of the subspace information from nearby blocks. As a result, small inaccuracies in the localization of the object do not accumulate. In comparison, the algorithm based on the affine-invariant Riemannian metric [24] only captures statistical properties of the object appearance in the whole object appearance region. More local spatial information inside the object region is lost.
- Our algorithm constructs a robust log-euclidean Riemannian subspace representation for each object appearance block. Covariance matrices of image features are mapped into the log-euclidean space, which is a vector space. The log-euclidean Riemannian subspace representation effectively summarizes geometric and structural information (e.g., mean, variance, and dominant projection directions) in the covariance matrices of the image features. However, the algorithm based on the affine-invariant Riemannian metric relies heavily on an intrinsic mean in the Lie group structure. The information in the covariance matrices of the image features is more likely to be lost for modeling object appearances.
- The algorithm based on the affine-invariant Riemannian metric [24] does not include online learning.

The vector subspace-based algorithm [14] does not model the spatial correlation between pixel values. As a result, the global or local variations in a scene may substantially change the vector subspace, resulting in tracking errors. Furthermore, the vector subspace-based tracking algorithm only models the total object appearance region using one subspace. This makes it susceptible to losing the track when there are local drastic changes in object appearance. However, our algorithm directly encodes spatial information, local correlations between pixel values and the object appearance variations, and thus makes the appearance model stable against global or local variations in the scene.

Fig. 21. An example of face entering detection. (a) The estimated optical flow field. (b) The detected motion region. (c) The evolved contour of the face. (d) The localization of the detected face.

Fig. 22. Example 7: Tracking two faces. From left to right, the frame numbers are 10271, 10286, 10300, 10318, and 10348, respectively. The first and second rows correspond to our algorithm and to the algorithm based on the affine-invariant Riemannian metric, respectively.

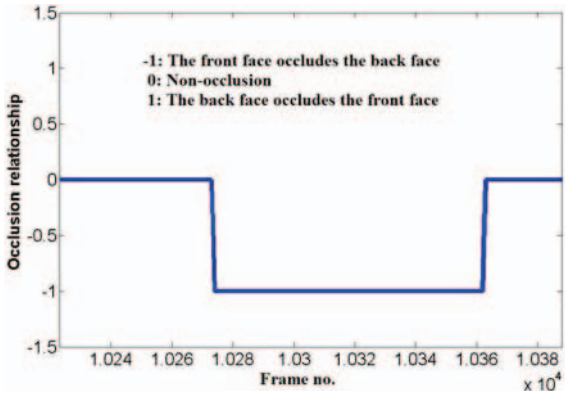


Fig. 23. Recovered occlusion relations for Example 7.

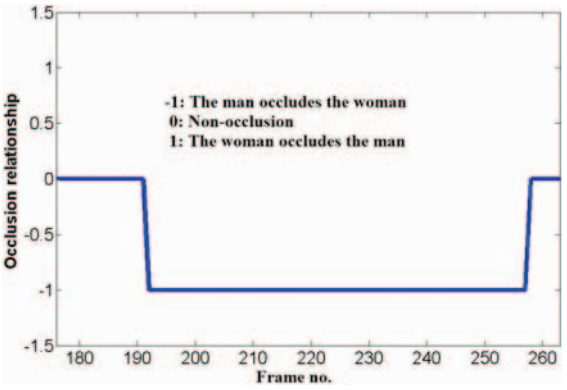


Fig. 26. Recovered occlusion relations for Example 9.

Jepson's algorithm also does not consider the correlations between pixel values. As a result, it is not robust enough to adapt to large appearance changes.

The main contribution of Yu's algorithm is the construction of a motion model: Particle filtering is replaced with an iteration method which greatly increases the speed of the algorithm. However, online updating of the appearance model is not carried out in Yu's algorithm. This makes the results of Yu's algorithm less accurate than the results of our algorithm when large appearance changes occur.

The MIL-based algorithm represents the appearance of the tracked object using a set of image patches with a fixed size, and thus emphasizes the inherent ambiguity of object localization. While the algorithm is robust to appearance changes, it may reduce tracking accuracy if the image patches do not precisely capture the object. Furthermore, the algorithm assumes that all the instances in a positive bag are positive. However, this assumption is sometimes violated in practice. This also reduces the accuracy of the tracking results.

Yang's algorithm uses color histograms to represent object appearance models. As a result, the spatial information in the object appearances is lost. Furthermore, in Yang's algorithm, the appearance models of objects are unchanged during occlusions. However, in our algorithm object appearance models can be updated even during occlusions.

Fig. 24. The results with different appearance model updating strategies: The first row is the results of our appearance model updating strategy and the second row is the results of the conventional updating strategy.

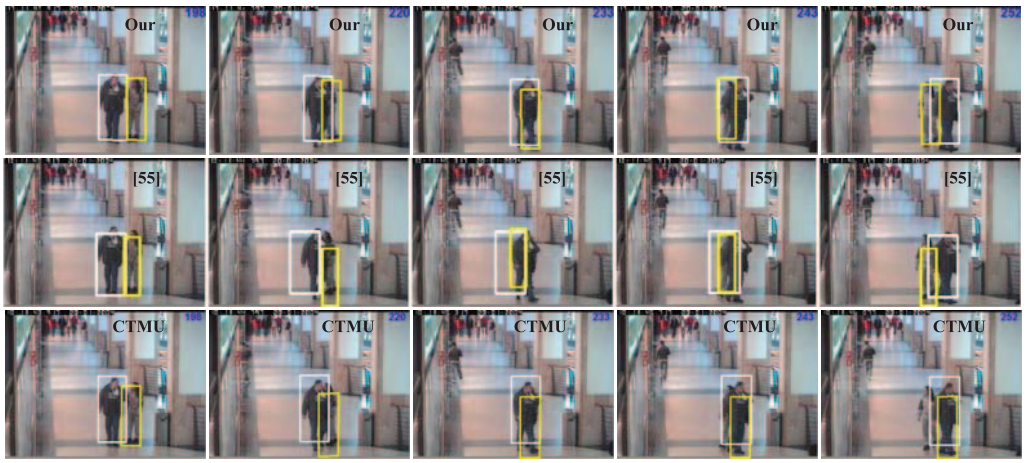


Fig. 25. Example 9: Tracking two pedestrians. From left to right, the frame numbers are 198, 220, 233, 243, and 252 respectively: The first row is the results of our algorithm, the second row is the results of Yang's algorithm [55], and the third row is the results of the algorithm based on the affine-invariant Riemannian metric.

TABLE 1
Quantitative Comparisons for Example 9

Evaluation / People	Our algorithm		Yang's algorithm		Affine-invariant Riemannian metric	
	Mean error	Number of successfully tracked frames	Mean error	Number of successfully tracked frames	Mean error	Number of successfully tracked frames
The man	4.17	89/89	14.37	78/89	7.56	89/89
The woman	4.66	87/89	12.55	86/89	31.98	57/89

Fig. 27. Example 10: Tracking three pedestrians: The first row is the results of our algorithm and the second row is the results of Yang's algorithm.

TABLE 2
Quantitative Comparisons between Our Algorithm and Yang's Algorithm for Example 10

Evaluation / People	Our algorithm		Yang's algorithm	
	Mean error	Number of successfully tracked frames	Mean error	Number of successfully tracked frames
Person A (Yellow box)	3.69	40/41	60.40	3/41
Person B (White box)	3.30	41/41	3.96	40/41
Person C (Blue box)	3.75	41/41	26.69	8/41

The block-division-based strategy in our algorithm handles occlusions more reasonably.

The runtime of our algorithm for each frame in all the examples is less than 1 second when there is no occlusion and less than 5 seconds in the case of occlusions, as measured on a P4-3.2G computer with 512M RAM. The reasons why the tracking speed is slower in the presence of occlusions than in the absence of occlusions are as follows:

- When occlusion is detected, more computation steps are required, such as computations relevant to the occlusion threshold and computations for occlusion reasoning.
- When occlusion is detected, more particles are used.

8 Conclusion

In this paper, we have proposed an incremental log-euclidean Riemannian subspace learning algorithm in which, under the log-euclidean Riemannian metric, image feature covariance matrices which directly describe spatial relations between pixel values are mapped into a vector space. The resulting linear subspace analysis is very effective in retaining the information on the covariance matrices. Furthermore, we have constructed a log-euclidean block-division appearance model which captures the local and global spatial layout information about object appearance. This appearance model ensures that our single object tracking algorithm can adapt to large appearance changes, and our algorithm for tracking multi-objects with occlusion reasoning can update the appearance models in the presence of occlusions. Experimental results have demonstrated that, compared with six state-of-art tracking algorithms, our tracking algorithm obtains more accurate tracking results when there are large variations in illumination, small objects, pose variations, occlusions, etc.

Acknowledgments

The authors thank Drs. Xue Zhou, Wei Li, Xinchu Shi, Mingliang Zhu, and Jian Chen for their valuable suggestions on the work. This work is partly supported by the NSFC (Grant No. 60825204, 60935002, 61100147), the National 863 High-Tech R&D Program of China (Grant No. 2012AA012504), the Natural Science Foundation of Beijing (Grant No. 4121003), the US National Science Foundation (IIS-0812114, CCF-1017828), the National Basic Research Program of China (2012CB316400), and the Alibaba Financial-Zhejiang University Joint Research Lab.

References

[1] X. Li, W.M. Hu, Z.F. Zhang, X.Q. Zhang, M.L. Zhu, and J. Cheng, "Visual Tracking via Incremental Log-Euclidean Riemannian Subspace Learning," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pp. 1-8, June 2008.

[2] M.J. Black and A.D. Jepson, "EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation," *Int'l J. Computer Vision*, vol. 26, no. 1, pp. 63-84, Jan. 1998.

[3] M. Isard and A. Blake, "Contour Tracking by Stochastic Propagation of Conditional Density," *Proc. European Conf. Computer Vision*, vol. 2, pp. 343-356, 1996.

[4] M.J. Black, D.J. Fleet, and Y. Yacoob, "A Framework for Modeling Appearance Change in Image Sequence," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 660-667, Jan. 1998.
[5] A.D. Jepson, D.J. Fleet, and T.F. El-Maraghi, "Robust Online Appearance Models for Visual Tracking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 415-422, 2001.
[6] S.K. Zhou, R. Chellappa, and B. Moghaddam, "Visual Tracking and Recognition Using Appearance-Adaptive Models in Particle Filters," *IEEE Trans. Image Processing,* vol. 13, no. 11, pp. 1491-1506, Nov. 2004.
[7] T. Yu and Y. Wu, "Differential Tracking Based on Spatial-Appearance Model (SAM)," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 720-727, June 2006.
[8] J. Li, S.K. Zhou, and R. Chellappa, "Appearance Modeling under Geometric Context," *Proc. IEEE Int'l Conf. Computer Vision,* vol. 2, pp. 1252-1259, 2005.
[9] K. Lee and D. Kriegman, "Online Learning of Probabilistic Appearance Manifolds for Video-Based Recognition and Tracking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 852-859, 2005.
[10] H. Lim, V. Morariu, O.I. Camps, and M. Sznaier, "Dynamic Appearance Modeling for Human Tracking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 751-757, 2006.
[11] J. Ho, K. Lee, M. Yang, and D. Kriegman, "Visual Tracking Using Learned Linear Subspaces," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 782-789, 2004.
[12] Y. Li, "On Incremental and Robust Subspace Learning," *Pattern Recognition,* vol. 37, no. 7, pp. 1509-1518, 2004.
[13] D. Skocaj and A. Leonardis, "Weighted and Robust Incremental Method for Subspace Learning," *Proc. Ninth IEEE Int'l Conf. Computer Vision,* vol. 2, pp. 1494-1501, Oct. 2003.
[14] D.A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental Learning for Robust Visual Tracking," *Int'l J. Computer Vision,* vol. 77, no. 2, pp. 125-141, May 2008.
[15] Y. Wu, T. Yu, and G. Hua, "Tracking Appearances with Occlusions," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 789-795, June 2003.
[16] A. Yilmaz, "Object Tracking by Asymmetric Kernel Mean Shift with Automatic Scale and Orientation Selection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1-6, June 2007.
[17] G. Silveira and E. Malis, "Real-Time Visual Tracking under Arbitrary Illumination Changes," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1-6, June 2007.
[18] M. Grabner, H. Grabner, and H. Bischof, "Learning Features for Tracking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1-8, June 2007.
[19] X. Zhou, W.M. Hu, Y. Chen, and W. Hu, "Markov Random Field Modeled Level Sets Method for Object Tracking with Moving Cameras," *Proc. Asian Conf. Computer Vision,* pp. 832-842, 2007.
[20] S. Ilic and P. Fua, "Non-Linear Beam Model for Tracking Large Deformations," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 1-8, June 2007.
[21] S. Tran and L. Davis, "Robust Object Tracking with Regional Affine Invariant Features," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 1-8, 2007.
[22] Q. Zhao, S. Brennan, and H. Tao, "Differential EMD Tracking," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 1-8, Oct. 2007.
[23] H. Wang, D. Suter, K. Schindler, and C. Shen, "Adaptive Object Tracking Based on an Effective Appearance Filter," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 9, pp. 1661-1667, Sept. 2007.
[24] F. Porikli, O. Tuzel, and P. Meer, "Covariance Tracking Using Model Update Based on Lie Algebra," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 728-735, 2006.
[25] O. Tuzel, F. Porikli, and P. Meer, "Human Detection via Classification on Riemannian Manifolds," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1-8, June 2007.
[26] P.T. Fletcher and S. Joshi, "Principal Geodesic Analysis on Symmetric Spaces: Statistics of Diffusion Tensors," *Proc. Computer Vision and Math. Methods in Medical and Biomedical Image Analysis,* pp. 87-98, 2004.
[27] T. Lin and H. Zha, "Riemannian Manifold Learning," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 5, pp. 796-809, May 2008.
[28] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, "Geometric Means in a Novel Vector Space Structure on Symmetric Positive-Definite Matrices," *SIAM J. Matrix Analysis and Applications,* vol. 29, no. 1, pp. 328-347, Feb. 2007.
[29] O. Tuzel, F. Porikli, and P. Meer, "Region Covariance: A Fast Descriptor for Detection and Classification," *Proc. European Conf. Computer Vision,* vol. 2, pp. 589-600, 2006.
[30] X. Pennec, P. Fillard, and N. Ayache, "A Riemannian Framework for Tensor Computing," *Int'l J. Computer Vision,* vol. 66, no. 1, pp. 41-66, Jan. 2006.
[31] W. Rossmann, *Lie Groups: An Introduction through Linear Group.* Oxford Univ. Press, 2002.
[32] 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, Aug. 2000.
[33] A. Mittal and L.S. Davis, "M2Tracker: A Multi-View Approach to Segmenting and Tracking People in a Cluttered Scene," *Int'l J. Computer Vision,* vol. 51, no. 3, pp. 189-203, Feb./Mar. 2003.
[34] C. Stauffer and W.E.L. Grimson, "Adaptive Background Mixture Models for Real-Time Tracking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 246-252, 1999.
[35] S. Yan, S. Shan, X. Chen, W. Gao, and J. Chen, "Matrix-Structural Learning (MSL) of Cascaded Classifier from Enormous Training Set," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1-7, June 2007.
[36] B. Babenko, M.-H. Yang, and S. Belongie, "Visual Tracking with Online Multiple Instance Learning," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 983-990, June 2009.
[37] G. Hager and P. Belhumeur, "Real-Time Tracking of Image Regions with Changes in Geometry and Illumination," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 403-410, June 1996.
[38] B. Wu and R. Nevatia, "Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian Combination of Edgelet Based Part Detectors," *Int'l J. Computer Vision,* vol. 75, no. 2, pp. 247-266, Nov. 2007.
[39] B. Wu and R. Nevatia, "Tracking of Multiple, Partially Occluded Humans Based on Static Body Part Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 951-958, June 2006.
[40] H. Wang and D. Suter, "Tracking and Segmenting People with Occlusions by a Sample Consensus-Based Method," *Proc. IEEE Int'l Conf. Image Processing,* vol. 2, pp. 410-413, Sept. 2005.
[41] S. Khan and M. Shah, "Tracking People in Presence of Occlusion," *Proc. Asian Conf. Computer Vision,* pp. 1132-1137, Jan. 2000.
[42] T. Zhao and R. Nevatia, "Tracking Multiple Humans in Crowded Environment," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 406-413, June-July 2004.
[43] N. Joshi, S. Avidan, W. Matusik, and D. Kriegman, "Synthetic Aperture Tracking: Tracking through Occlusions," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 1-8, Oct. 2007.
[44] M. Yang, Z. Fan, J. Fan, and Y. Wu, "Tracking Nonstationary Visual Appearances by Data-Driven Adaptation," *IEEE Trans. Image Processing,* vol. 18, no. 7, pp. 1633-1644, July 2009.
[45] J. Kwon and K.M. Lee, "Tracking of a Non-Rigid Object via Patch-Based Dynamic Appearance Modeling and Adaptive Basin Hopping Monte Carlo Sampling," *Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops,* pp. 1208-1215, June 2009.
[46] K. Ishiguro, T. Yamada, and N. Ueda, "Simultaneous Clustering and Tracking Unknown Number of Objects," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1-8, June 2008.
[47] X. Song, J. Cui, H. Zha, and H. Zhao, "Vision-Based Multiple Interacting Targets Tracking via On-Line Supervised Learning," *Proc. 10th European Conf. Computer Vision,* vol. 3, pp. 642-655, 2008.
[48] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua, "Multicamera People Tracking with a Probabilistic Occupancy Map," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 2, pp. 267-282, Feb. 2008.
[49] S. Khan and M. Shah, "Tracking Multiple Occluding People by Localizing on Multiple Scene Planes," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 31, no. 3, pp. 505-519, Mar. 2009.
[50] X. Mei and H.B. Ling, "Robust Visual Tracking Using L1 Minimization," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 1436-1443, 2009.

[51] D. Liang, Q. Huang, H. Yao, S. Jiang, R. Ji, and W. Gao, "Novel Observation Model for Probabilistic Object Tracking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1387-1394, June 2010.

[52] W. He, T. Yamashita, H. Lu, and S. Lao, "Surf Tracking," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 1586-1592, 2009.

[53] D.-N. Ta, W.-C. Chen, N. Gelfand, and K. Pulli, "SURFTrac: Efficient Tracking and Continuous Object Recognition Using Local Feature Descriptors," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 2937-2944, June 2009.

[54] W. Qu, D. Schonfeld, and M. Mohamed, "Real-Time Distributed Multi-Object Tracking Using Multiple Interactive Trackers and a Magnetic-Inertia Potential Model," *IEEE Trans. Multimedia,* vol. 9, no. 3, pp. 511-519, Apr. 2007.

[55] M. Yang, T. Yu, and Y. Wu, "Game-Theoretic Multiple Target Tracking," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 1-8, 2007.

[56] Y. Jin and F. Mokhtarian, "Variational Particle Filter for Multi-Object Tracking," *Proc. Int'l Conf. Computer Vision,* pp. 1-8, 2007.

[57] L. Zhang, Y. Li, and R. Nevatia, "Global Data Association for Multi-Object Tracking Using Network Flows," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1-8, June 2008.

[58] A. Ess, B. Leibe, K. Schindler, and L.V. Gool, "A Mobile Vision System for Robust Multi-Person Tracking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1-8, June 2008.

[59] C. Huang, B. Wu, and R. Nevatia, "Robust Object Tracking by Hierarchical Association of Detection Responses," *Proc. 10th European Conf. Computer Vision,* vol. 2, pp. 788-801, 2008.

[60] D. Mitzel, E. Horbert, A. Ess, and B. Leibe, "Multi-Person Tracking with Sparse Detection and Continuous Segmentation," *Proc. European Conf. Computer Vision,* pp. 397-410, Sept. 2010.

[61] J. Kwon, K.M. Lee, and F.C. Park, "Visual Tracking via Geometric Particle Filtering on the Affine Group with Optimal Importance Functions," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 991-998, June 2009.

[62] F. Porikli and O. Tuzel, "Learning on Lie Groups for Invariant Detection via Tracking," *Proc. Int'l Workshop Object Recognition,* invited, 2008.

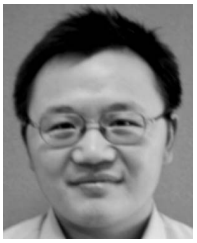
[63] E.B. Sudderth, M.I. Mandel, W.T. Freeman, and A.S. Willsky, "Distributed Occlusion Reasoning for Tracking with Nonparametric Belief Propagation," *Proc. Ann. Conf. Neural Information Processing Systems,* pp. 1369-1376, 2004.

[64] L. Zhang, Y. Li, and R. Nevatia, "Global Data Association for Multi-Object Tracking Using Network Flows," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1-8, 2008.

[65] C. Wang, M.L. Gorce, and N. Paragios, "Segmentation, Ordering, and Multi-Object Tracking Using Graphical Models," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 747-754, 2009.

[66] E. Herbst, S. Seitz, and S. Baker, "Occlusion Reasoning for Temporal Interpolation Using Optical Flow," technical report, Microsoft Research, Aug. 2009.

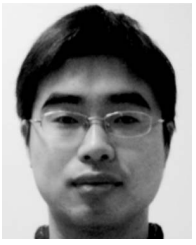
[67] V. Gay-Bellile, A. Bartoli, and P. Sayd, "Direct Estimation of Nonrigid Registrations with Image-Based Self-Occlusion Reasoning," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 32, no. 1, pp. 87-104, Jan. 2010.



Weiming Hu received the PhD degree from the Department of Computer Science and Engineering, Zhejiang University, in 1998. From April 1998 to March 2000, he was a postdoctoral research fellow with the Institute of Computer Science and Technology, Peking University. Now he is a professor in the Institute of Automation, Chinese Academy of Sciences. His research interests include visual surveillance and filtering of Internet objectionable information.

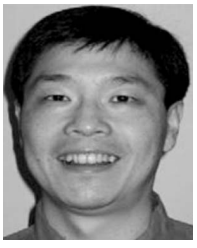
Xi Li received the doctoral degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2009. He is currently a senior research associate at the University of Adelaide, Australia. From 2009 to 2010, he worked as a postdoctoral researcher at CNRS Telecomd ParisTech, France.

Wenhan Luo received the BSc degree in automation from Huazhong University of Science and Technology, China, in 2009. Currently, he is working toward the MSc degree in the Institute of Automation, Chinese Academy of Sciences, China. His research interests include computer vision and pattern recognition.



Xiaoqin Zhang received the BSc degree in electronic information science and technology from Central South University, China, in 2005 and the PhD degree from the Institute of Automation, Chinese Academy of Sciences, China, in 2010. He is currently a lecturer at Wenzhou University, China. His research interests include visual tracking, motion analysis, and action recognition.

Stephen Maybank received the BA degree in mathematics from King's College Cambridge in 1976 and the PhD degree in computer science from Birkbeck College, University of London in 1988. Now he is a professor in the School of Computer Science and Information Systems, Birkbeck College. His research interests include the geometry of multiple images, camera calibration, visual surveillance, etc.



Zhongfei Zhang received the BS degree in electronics engineering, the MS degree in information science, both from Zhejiang University, China, and the PhD degree in computer science from the University of Massachusetts at Amherst. He is a professor of computer science at the State University of New York at Binghamton. His research interests include computer vision and multimedia processing, etc.