

AN EFFICIENT MULTI OBJECT TRACKING WITH WIEMENN SUBSPACE MODEL

D.Neelima¹, K.M. Prathyusha Kalathoti²

¹ Assistant Professor, Dept of CSE, Jawaharlal Nehru Technological University, Kakinada, India.

² M.Tech student, Dept of CSE, Jawaharlal Nehru Technological University, Kakinada, India.

Abstract

Object appearance modeling is crucial for tracking objects especially in videos captured by non-stationary cameras and for reasoning about occlusions between multiple moving objects. Based on the log-Euclidean Wiemann metric on symmetric positive definite matrices, we propose an incremental log-Euclidean Wiemann subspace learning algorithm in which covariance matrices of image features are mapped into a vector space with the log-Euclidean Wiemann 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.

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 learnt subspaces [14], etc. Among these appearance models, subspace-based ones have attracted much attention, because of their robustness

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 good subspace learning-based tracking algorithm. A pre-trained, 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 nice visual tracking algorithm based on linear subspace learning. In each subspace update, the subspace is re-computed 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 good weighted incremental PCA algorithm for subspace learning. Its limitation is that each update includes only one new sample, rather than multi-samples, and as a result it is necessary to update the subspace at every frame. Li [12] proposes a good incremental PCA algorithm for subspace learning. It can deal with multi-samples 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 and Lim *et al.* [14] propose a robust 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 correlations between values of pixels in the tracked image region. Lee and Kriegman [9] present a nice online algorithm to incrementally learn a generic appearance model for video-based recognition and

tracking. Lim *et al.* [10] present an interesting 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, 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.

Proposed Work

Wiemenn metrics :

A covariance matrix descriptor [24, 29], which is obtained based on the features of intensity derivatives etc, 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 etc. 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 Wiemenn metric. Researchers have applied Wiemenn metrics to model object appearances. Porikli *et al.* propose a Wiemenn metric-based object tracking method in which object appearances are represented using the covariance matrix of image features. Tuzel *et al.* propose an algorithm for detecting people by classification on Wiemenn manifolds. Wiemenn 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 2-D affine group for visual tracking. Porikli and Tuzel [62] propose a novel Lie group learning-based motion model for tracking combined with object detection. The algorithms in [24, 25] represent object appearances by points on a Wiemenn manifold and utilize an affine-invariant Wiemenn metric to calculate a Wiemenn mean for the data. There is no closed form solution for the Wiemenn mean. It is computed using an iterative numerical procedure. Arsigny *et al.* [28] propose the log-Euclidean Wiemenn metric for statistics on the manifold of symmetric positive definite matrices. This metric is simpler than the affine-invariant

Wiemenn metric. In particular, the computation of a sample's Wiemenn mean is more efficient than in the affine invariant case. Kwon *et al.* [61] propose a closed form approximation to the Wiemenn mean of a set of particle offsets. In this paper, we apply the log-Euclidean Wiemenn metric to represent object appearances and construct a new subspace-based appearance model for object tracking.

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 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 over-sensitive 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,]. 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 non-rigid appearance variations efficiently. Yang *et al.* use negative data constraints and bottom-up pair-wise data constraints to dynamically adapt to the changes in the object appearance. Kwon and Lee use a local patch-based appearance model to maintain relations between local patches by online updating. Mei and Ling 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.* track objects using the relations between local invariant feature motions and the object global motion. Ta *et al.* 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.

Multi-object tracking

There has been much work on tracking multi-objects using object appearance models in videos captured by stationary or non-stationary 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. Khan and Shah 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.* 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 depend on background subtraction. Zhao and Nevatia adopt a 3D shape model as well as camera models to track people and handle occlusions. Mittal and Davis use appearance models to detect people and an occlusion likelihood is applied to reason about occlusion relations between people. Joshi *et al.* track a 3D object through significant occlusions by combining video sequences from multiple nearby cameras. The algorithms in [33, 42, 43] depend on a costly camera calibration. Fleuret *et al.* use multi-cameras to model positions of multi-objects during tracking. Their algorithm depends on a discrete occupancy grid, besides camera calibration. Khan and Shah track multiple occluding people by localizing them on multiple scene planes. The algorithm depends on the planar homography occupancy constraint between multi-cameras.

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

Multi-object tracking with non-stationary cameras

For non-stationary cameras, background subtraction, calibration, and homography constraints cannot be used. As a result, multi-object tracking with non-stationary 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 multi-view templates. Qu *et al.* use a magnetic-inertia potential model to carry out the multi-object labeling. Yang *et al.* track multi-objects by finding the Nash equilibrium of a game. Jin and Mokhtarian use the variational particle filter to track multi-objects. One limitation in current algorithms for tracking multi-objects in videos taken by non-stationary 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 multi-pedestrians. These methods firstly 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 the hypothesis selection the Hungarian algorithm and continuous segmentation. The performance of these detection-based tracking methods greatly depends on the accuracy of pedestrian detection.

Incremental Log-Euclidean Wiemann Subspace

Learning

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

3.1. Covariance matrix descriptor

Let $i f$ be a d -dimensional feature vector of pixel i in an image region. The vector $i f$ is defined by $(x, y, (E_j) j \square 1, \dots, \square)$ where (x, y) are the pixel coordinates, \square is the number of color channels in the image, and

$$E_j = \left(I^j, |I_x^j|, |I_y^j|, \sqrt{(I_x^j)^2 + (I_y^j)^2}, |I_{xx}^j|, |I_{yy}^j|, \arctan \frac{|I_x^j|}{|I_y^j|} \right) \quad (1)$$

intensity derivatives in the j th color channel, and the last term is the first-order gradient orientation. For a grayscale image, if is a 9-dimensional feature vector (i.e. $\mathcal{U} = 1$ and $d = 9$). For a color image with three channels, if is a 23-dimensional vector (i.e. $\mathcal{U} = 3$ 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 $\{f_i\}_{i=1, \dots, L}$. The image region R is represented using a $d * d$ covariance matrix R_C [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.

Wiemenn 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 Wiemenn geometry for symmetric positive definite matrices is available for calculating statistics of covariance matrices. The Wiemenn geometry depends on the Wiemenn metric which describes the distance relations between samples in the Wiemenn space and determines the computation of the Wiemenn mean.

In the space of $d * d$ 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 ($A = U e U^T$) produces the orthogonal matrix U , and the diagonal matrix

$$\Sigma = \text{Diag}(\lambda_1, \lambda_2, \dots, \lambda_d) \tag{3}$$

where $\{\lambda_i\}_{i=1,2,\dots,d}$

are the eigenvalues of A . Then, the matrix exponential of A is defined by:

$$\exp(A) = \sum_{k=0}^{\infty} \frac{A^k}{k!} = U \cdot \text{Diag}(\exp(\lambda_1), \exp(\lambda_2), \dots, \exp(\lambda_d)) \cdot U^T$$

The matrix logarithm of A is defined by

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

where $d I$ is the $d * d$ identity matrix.

The log-Euclidean Wiemenn metric was proposed [28, 61]. The symmetric positive definite matrices are a subset of a Lie group. Under the log-Euclidean Wiemenn 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 \square of matrix logarithms obtained using the matrix logarithmic operation in (4) is simply their arithmetic mean. Given N symmetric positive definite matrices $\{X_i\}_{i=1, \dots, N}$, the mean \square in the Lie algebra is explicitly computed by

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

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

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

Moreover, under the log-Euclidean Wiemenn 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 Wiemenn 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 Wiemenn metric is used to calculate statistics of covariance matrices of image features

Incremental log-Euclidean Wiemenn

Subspace learning

As the log-Euclidean Wiemenn 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 Wiemenn 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 $\{C^d \in \mathbb{R}^{d \times d}\}_{d=1,2,\dots,N}$ which constitute a covariance matrix sequence $A \in \mathbb{R}^{d \times d \times N}$. In order to ensure that C^d

is not a singular matrix, we replace C^t with $C^t + \epsilon I_d$, where ϵ is a very small positive constant and I 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: $\alpha = (\log(c^1), \dots, \log(c^t), \dots, \log(c^N))$. We unfold the matrix $\log(C^t)$ into a d^2 -dimensional column vector $V^t (1 \leq t \leq N)$ 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 $\Gamma = (v^1 v^2 \dots v^t \dots v^N) \in \mathbb{R}^{d^2 \times N}$ (the t -th 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 Γ . This subspace is incrementally updated when new data arrive. The mean vector μ is obtained by taking the mean of the column vectors in Γ . We construct a matrix X whose columns are obtained by subtracting μ from each column vector in Γ . The SVD for X is carried out: $X = UDV^T$, producing a $d \times d$ 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 ($k \leq N$) 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 Wiemann subspace is represented by $\{\mu, U^k, D^k\}$. The incremental SVD technique in [14, 32] is applied to incrementally update the log-Euclidean Wiemann subspace. Let $\{\mu_{t-1}, U_{t-1}^k, D_{t-1}^k\}$ be the previous log-Euclidean Wiemann subspace at stage $t-1$. At stage t , a new covariance matrix sequence $A^* \in \mathbb{R}^{d \times d \times N^*}$ 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 $\Gamma^* \in \mathbb{R}^{d^2 \times N^*}$. 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 Γ^* . This incremental updating process is outlined as follows:

Step 1: Update the mean vector:

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

Where μ^* is the mean column vector of Γ^* , and r 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 Γ^* have the zero mean: $\Gamma^* \leftarrow \Gamma^* - \mu^*$.

Step 3: Construct the combined matrix Υ :

$$\Upsilon = \left(\Gamma U_{t-1} D_{t-1} | \Upsilon^* | \sqrt{\frac{N N^*}{N + N^*}} (\mu_{t-1} - \mu^*) \right), \tag{8}$$

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_i and singular values D_i by:

$$U_i = QU, D_i = D \cdot \sqrt{N / (N^* + \Gamma \cdot N)}. \tag{9}$$

Step 7: The k largest singular values in D_i are selected to form the diagonal matrix D_i^k , and the k columns corresponding to the elements in D_i^k are chosen from U_i to form U_i^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 Wiemann space 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 $C^t \in \mathbb{R}^{d \times d}$ 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 Wiemann subspace $\{\mu, U, D\}$, the square \mathbb{Z} of the Euclidean vector distance between v^t and $\{\mu, U, D\}$ is calculated as the subspace reconstruction error:

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

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

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

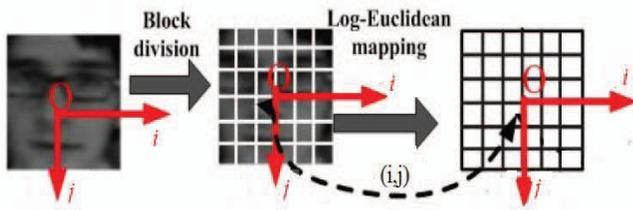
The smaller the \mathbb{Z} , the larger the likelihood.

4. Log-Euclidean Block-Division Appearance Model

We divide the object appearance region into non-overlapping blocks whose log-Euclidean Wiemann 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 $\{F^t\}_{t=1,2,\dots,N}$, 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) ($1 \leq i \leq m, 1 \leq j \leq n$), the covariance matrix feature $C_{ij}^t \in R^{d \times d}$ is extracted using Equations (1) and (2). Covariance matrices $\{C_{ij}^t\}_{t=1,2,\dots,N}$ 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 $Y_{ij} \in R^{d^2 \times N}$. 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 Y_{ij} is learned using our incremental log-Euclidean Wiemann subspace learning algorithm.



The square 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 $M_l = (P_{ij}^l)_{m \times n} \in R^{m \times n}$ 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 be 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 to the center of the appearance region, the more weight it is given.

4.4. Observation likelihood

The overall likelihood p of a candidate object appearance region given the learned block-division appearance model positively relates to the product of all the corresponding block-specific likelihoods after the local and global spatial filtering:

$$p_{overall} \propto \prod_{i=1}^m \prod_{j=1}^n p_{ij}^{\epsilon} \quad (14)$$

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

1 1

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

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 relations between the image regions corresponding to the block in the image sequence are reflected in the log-Euclidean Wiemann 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 non-stationary cameras is usually modeled by affine warping which is defined by parameters $(x_t, y_t, \eta_t, s_t, \beta_t, \phi_t)$, and x_t, y_t, η_t, s_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 $p(O_t | X_t)$ and a dynamic model X_t 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 $t X$ 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 Equations (14) and (15). A Gaussian distribution [14] with a diagonal covariance matrix with diagonal elements

$$\sigma_x^2, \sigma_y^2, \sigma_\eta^2, \sigma_s^2, \sigma_\beta^2 \text{ and } \sigma_\phi^2$$

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 non-stationary cameras, to localize multiple moving objects even when there are occlusions between them, and to explicitly determine their 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 blocks with large appearance changes do not take part in the likelihood evaluation, and the appearance subspaces of these blocks are unchanged, while those for the remaining blocks are updated in the current frame. Then, the dynamical model and the proposal density function in our multi-object tracking algorithm are the same as those used in single object tracking. The observation model and the appearance updating model for multi-object tracking are designed to handle occlusions, and are of course different from those used in the single object tracking algorithm. In the following, we describe occlusion detection, observation likelihood evaluation 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. Observation likelihood 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 (a) shows an exemplar video frame in which the bottom part of a girl's face is occluded by a man's face, and (b) shows the reconstruction errors of blocks of the girl's face. As shown in (b), blocks corresponding to the occluded part of the girl's face have much larger reconstruction errors than the un-occluded blocks. Only blocks with reconstruction errors less than a given threshold $\mathcal{Z}_{threshold}$ are used to evaluate the likelihood. Equation (15) is replaced by:

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

where Ω is the set of blocks with reconstruction errors less than $\mathcal{Z}_{threshold}$ 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 learnt 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 $\mathcal{Z}_{threshold}$. 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 non-rigid 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 (their previous occlusion relations depend on their previous states). 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 \bar{z}_{oca} and \bar{z}_{ocb} be, respectively, the average reconstruction errors of such overlapped blocks in objects a and b . Let \bar{z}_{oca} and \bar{z}_{ocb} 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 :

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

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

$$\phi_{(a,b)}^t = \begin{cases} -1 & \text{if } \bar{z}_{oca} - \bar{z}_a < \bar{z}_{ocb} - \bar{z}_b \\ 1 & \text{if } \bar{z}_{oca} - \bar{z}_a > \bar{z}_{ocb} - \bar{z}_b \\ \phi_{(a,b)}^{t-1} & \text{otherwise} \end{cases} \quad (18)$$

6.5. Appearance and disappearance

We handle the appearance and disappearance of objects for both stationary and non-stationary 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 can be 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 non-stationary 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 have not only 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 fit closely 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 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 non-stationary 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 un-tracked moving objects or scene elements, e.g. static objects: block-

wise appearance outliers of the object are monitored and the subspaces of the un-occluded 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 Wiemann 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 non-stationary 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 non-stationary 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_i' defined in (9) to the sum of all the singular values in D_i' . The number k is the least number such that the reconstruction quality is above 98%. 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 $(\sigma_x^2, \sigma_y^2, \sigma_\eta^2, \sigma_s^2, \sigma_\beta^2, \sigma_\phi^2)$ in the dynamic model were given the values of 52, 52, 0.032, 0.032, 0.0052 and 0.0012, respectively. The forgetting factor r in (7), (8), and (9) was set to 0.99. The factor σ_ξ in (12) was set to 8. The factor

σ_{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 un-occluded 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 Wiemann 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 Wiemann 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

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 the 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 Wiemann 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 onwards. 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 onwards, 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 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. 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.

Fig. 5 shows the results obtained by omitting non-overlapping blocks, or local and global filtering from our algorithm. Fig. 6 shows tracking error curves with and without non-overlapping blocks or local and global filtering. The mean errors without non-overlapping blocks, with non-overlapping blocks but without local and global filtering, and with non-overlapping blocks and local and global filtering, are 15.25, 7.47, and 5.29, respectively. It is apparent that the tracking results without non-overlapping blocks are much less accurate than the results with non-overlapping blocks but without local and global

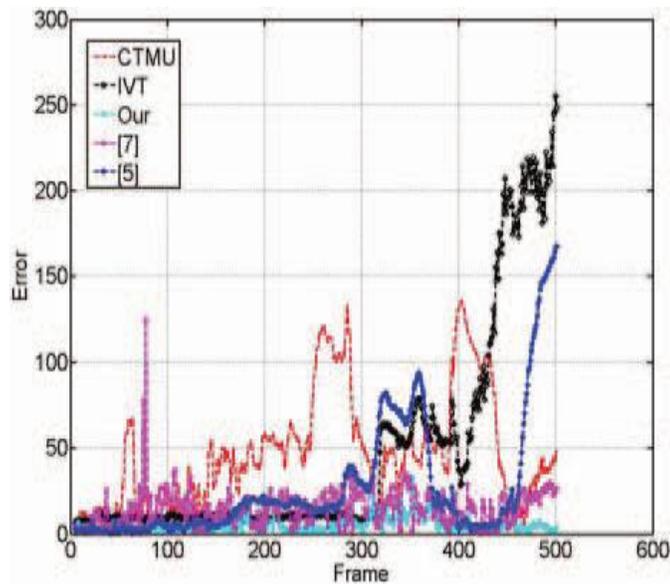


Fig. 4. The quantitative comparison between our algorithm and the competing algorithms for Example 1: Our algorithm corresponds to the cyan curve, the algorithm based on the affine-invariant Wiemann metric the red curve, the vector subspace-based algorithm the black curve, Jepson’s algorithm the blue curve, and Yu’s algorithm the magenta curve. filtering, which are overall less accurate than the results with non-overlapping blocks and local and global filtering. So, the division of the object appearance into blocks is more important than the local and global filtering.



Fig. 5. The results for Example 1 without non-overlapping blocks, or local and global filtering: The first row shows the results without non-overlapping blocks; The second row shows the results with non-overlapping blocks but without local and global filtering.

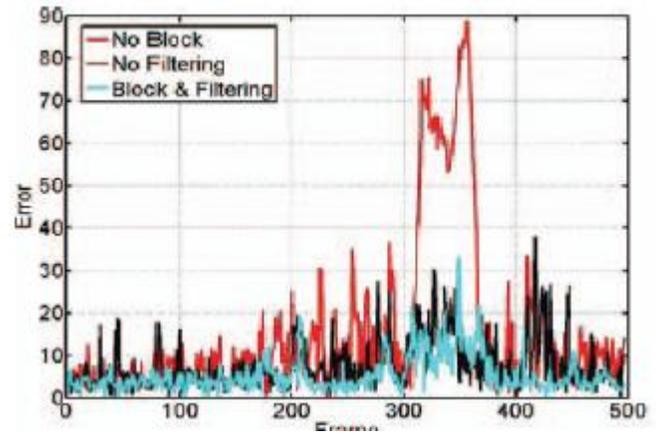


Fig. 6. The quantitative comparison results for Example 1 with and without non-overlapping blocks, or local and global filtering: the red, black and blue curves correspond, respectively, to the results without non-overlapping blocks, the results with non-overlapping blocks but without local and global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

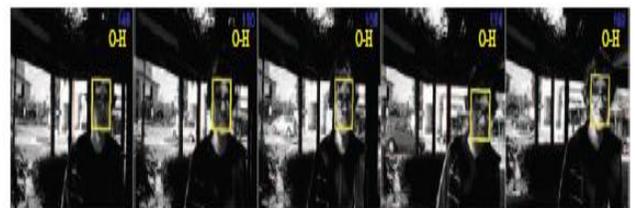


Fig. 7. Example 1: Tracking the face of the boy using occlusion handling.

As stated in Section 6.6, our block-wise 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 handling. 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.

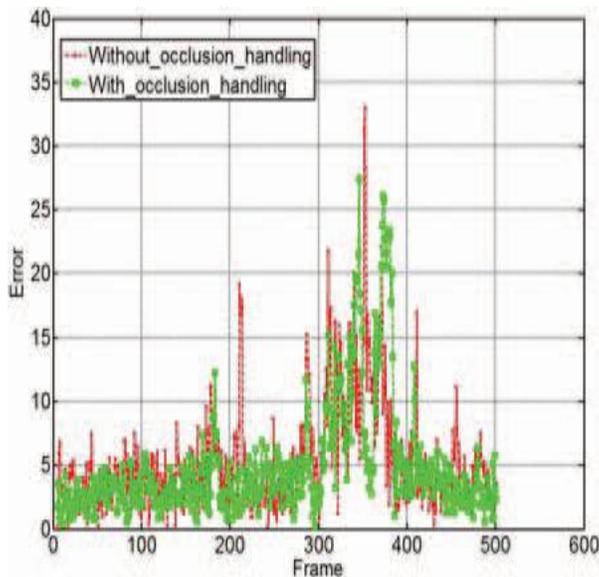


Fig. 8. The quantitative comparison between the results with and without occlusion handling for Example 1: the green and red curves correspond to the results with and without occlusion handling, respectively

CONCLUSION

In this paper, we have proposed an incremental log-Euclidean Wiemann subspace learning algorithm in which under the log-Euclidean Wiemann 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, and occlusions etc.

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 Wiemann Subspace Learning," in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition*, Anchorage Alaska. June 23-28, pp. 1-8, 2008.

2. M.J. Black and A.D. Jepson, "EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation," *International Journal of 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," in *Proc. of European Conference on 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," in *Proc. of IEEE International Conference on 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," in *Proc. of IEEE Conference on 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. on 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)," in *Proc. of IEEE Conference on 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," in *Proc. of IEEE International Conference on 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," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 852-859, 2005.

10. H. Lim, V. Morariu, O.I. Camps, and M. Sznaiar, "Dynamic Appearance Modeling for Human Tracking," in *Proc. of IEEE Conference on 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," in *Proc. of IEEE Conference on 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," in *Proc. of IEEE*

International Conference on 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," *International Journal of Computer Vision*, vol. 77, no. 2, pp. 125-141, May 2008.

15. Y. Wu, T. Yu, and G. Hua, "Tracking Appearances with Occlusions," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Madison, vol. 1, pp. 789-795, June 2003.

16. A. Yilmaz, "Object Tracking by Asymmetric Kernel Mean Shift with Automatic Scale and Orientation Selection," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-6, June 2007.

17. G. Silveira and E. Malis, "Real-Time Visual Tracking under Arbitrary Illumination Changes," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-6, June 2007.

18. M. Grabner, H. Grabner, and H. Bischof, "Learning Features for Tracking," in *Proc. of IEEE Conference on 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," in *Proc. of Asian Conference on Computer Vision*, Part I, pp. 832-842, 2007.

20. S. Ilic and P. Fua, "Non-Linear Beam Model for Tracking Large Deformations," in *Proc. of IEEE International Conference on Computer Vision*, pp. 1-8, June 2007.

21. S. Tran and L. Davis, "Robust Object Tracking with Regional Affine Invariant Features," in *Proc. of IEEE International Conference on Computer Vision*, pp. 1-8, 2007.

22. Q. Zhao, S. Brennan, and H. Tao, "Differential EMD Tracking," in *Proc. of IEEE International Conference on 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. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 9, pp. 1661-1667, Sep. 2007.

24. F. Porikli, O. Tuzel, and P. Meer, "Covariance Tracking Using Model Update Based on Lie Algebra," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 728-735, 2006.

25. O. Tuzel, F. Porikli, and P. Meer, "Human Detection via Classification on Wiemann Manifolds," in *Proc. of IEEE Conference on 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," in *Proc. of Computer Vision and Mathematical Methods in Medical and Biomedical Image Analysis*, pp. 87-98, 2004.

27. T. Lin and H. Zha, "Wiemann Manifold Learning," *IEEE Trans. on 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 Journal on 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," in *Proc. of European Conference on Computer Vision*, vol. 2, pp. 589-600, 2006.

30. X. Pennec, P. Fillard, and N. Ayache, "A Wiemann Framework for Tensor Computing," *International Journal of Computer Vision*, vol. 66, no. 1, pp. 41-66, Jan. 2006.

31. W. Rossmann, "Lie Groups: an Introduction through Linear Group," Oxford Press, 2002.

32. A. Levy and M. Lindenbaum, "Sequential Karhunen-Loeve Basis Extraction and Its Application to Images," *IEEE Trans. on 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," *International Journal of Computer Vision*, vol. 51, no. 3, pp. 189-203, Feb./Mar. 2003. **IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

34. C. Stauffer and W.E.L. Grimson "Adaptive Background Mixture Models for Real-Time Tracking," in *Proc. of IEEE Conference on 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,” in *Proc. of IEEE Conference on 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,” in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 983-990, June 2009

BIOGRAPHIES



Mrs. D. Neelima is B.Tech(CSE), M.Tech(CSE) from JNTU Hyderabad and currently pursuing P.hd in JNTU Kakinada , Andhra Pradesh, India. She is working as Assistant professor in Computer Science & Engineering department in Jawaharlal Nehru Technological University Kakinada , Andhra Pradesh, India. She has 4 years of experience in teaching Computer

Science and Engineering related subjects . She is a research scholar and her area of interest and research includes Video Image Processing. She has published several Research papers out of which 2 are international Journals and 2 are international conferences. She has guided more than 60 students of Bachelor degree, 20 Students of Master degree in Computer Science and Engineering in their major projects.



I am **K.M.Prathyusha Kalathoti** doing M.Tech in Jawaharlal Nehru Technological University, Kakinada ,A.P.,and interesting research area is Video Image Processing.