Object Tracking with Bayesian Estimation of Dynamic Layer Representations

Hai Tao, *Member*, *IEEE Computer Society*, Harpreet S. Sawhney, *Member*, *IEEE Computer Society*, and Rakesh Kumar, *Member*, *IEEE Computer Society*

Abstract—Decomposing video frames into coherent two-dimensional motion layers is a powerful method for representing videos. Such a representation provides an intermediate description that enables applications such as object tracking, video summarization and visualization, video insertion, and sprite-based video compression. Previous work on motion layer analysis has largely concentrated on two-frame or multiframe batch formulations. The temporal coherency of motion layers and the domain constraints on shapes have not been exploited. This paper introduces a complete dynamic motion layer representation in which spatial and temporal constraints on shape, motion, and layer appearance are modeled and estimated in a maximum a posteriori (MAP) framework using the generalized expectation-maximization (EM) algorithm. In order to limit the computational complexity of tracking arbitrarily shaped layer ownership, we propose a shape prior that parameterizes the representation of shape and prevents motion layers from evolving into arbitrary shapes. In this work, a Gaussian shape prior is chosen to specifically develop a near real-time tracker for vehicle tracking in aerial videos. However, the general idea of using a parametric shape representation as part of the state of a tracker is a powerful one that can be extended to other domains as well. Based on the dynamic layer representation, an iterative algorithm is developed for continuous object tracking over time. The proposed method has been successfully applied in an airborne vehicle tracking system. Its performance is compared with that of a correlation-based tracker and a motion change-based tracker to demonstrate the advantages of the new method. Examples of tracking when the backgrounds are cluttered and the vehicles undergo various rigid motions and complex interactions such as passing, turning, and stop-and-go demonstrate the strength of the complete dynamic layer representation.

Index Terms—Motion analysis, dynamic layer representation, tracking, aerial video surveillance.

1 Introduction

Over the past several years, layer representations and their associated algorithms have emerged as powerful motion analysis tools. Motion layers represent regions of homogeneous motion in an image sequence. The motion models and their layers of support together constitute a compact representation of the significant scene structures. Algorithms have been designed based on such representations to precisely estimate and segment the motions of multiple independent components in dynamic scenes. Some applications enabled by these algorithms are video insertion, sprite-based video compression, and video summarization. The key idea of layer-based motion analysis is to estimate both the motions and the support of independent moving objects simultaneously based on the motion coherency across images. Each layer possesses a coherent two-dimensional motion that is usually modeled as rigid, affine, or projective. Starting from an initial solution, the motion and the segmentation are iteratively estimated: from the estimated segmentation, the motion is refined; from the estimated motion, better segmentation is computed. Such an iterative process is equivalent to the expectation-maximization (EM) algorithm for unsupervised data clustering where each motion layer is a cluster. The bulk of existing work has largely concentrated on two-frame or multiframe batch formulations in which various motion models and local constraints on the layer segmentation are employed to regularize the solution.

In this paper, we consider a more general problem of estimating motion layers in extended image sequences. This requires a mechanism for maintaining the coherency of the motion, the appearance, and the shape of each layer over time. We solve this problem by formulating a complete dynamic motion layer representation in which the spatial and temporal constraints on shape, motion, and layer appearance are modeled. This representation is continuously estimated over time in a maximum a posteriori (MAP) framework using the generalized EM algorithm. More specifically, the main contributions of this paper are:

1. Use of a new global shape constraint to incorporate the domain knowledge of the object shapes into the estimation process. The shape constraint is a parametric prior function in the Bayesian formulation. Its main purpose is to prevent motion layers from evolving into arbitrary shapes and to limit the computational complexity of tracking layer ownership.
2. Temporal tracking of the complete layer representation that consists of appearance, motion, segmentation, and shape.
3. A generalized EM algorithm to continuously estimate the proposed dynamic layer representation

- *H. Tao is with the Department of Computer Engineering, University of California at Santa Cruz, Santa Cruz, CA 95064. E-mail: tao@soe.ucsc.edu.*
- *H.S. Sawhney and R. Kumar are with Sarnoff Corporation, 201 Washington Rd., Princeton, NJ 08543. E-mail: {hsawhney,rkumar}@sarnoff.com.*

Manuscript received 22 May 2000; revised 9 May 2001; accepted 25 July 2001.
Recommended for acceptance by M. Irani.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 111746.

TABLE 1
Three Categories of Motion Models, Shape Constraints, and Appearance Constraints

	Local Spatial	Global Spatial	Dynamic
Motion Models	**Smooth dense flow:** Weiss97	**2D projective:** Torr99 **2D affine:** Darrell91, Wang93, Hsu94, Ayer95 Weiss96, Vasconcelos97 **2D rigid:** This paper – Section 2.2	**Constant velocity:** This paper – Section 2.1
Shape Constraints	**MRF segmentation prior:** Weiss96, Vasconcelos97	**Background+Gaussian segmentation prior:** This paper – Section 2.2	**Constant segmentation prior:** This paper – Section 2.2
Appearance Constraints			**Constant appearance:** This paper – Section 2.3

over time. This algorithm has been successfully applied to an airborne vehicle tracking system.

In the following paragraphs, we will briefly examine existing models and constraints and motivate the key ideas in the new approach. In Table 1, three categories of motion models, shape constraints, and appearance constraints are presented. Each category is further divided into three subclasses: local spatial constraints, global spatial constraints, and dynamic constraints. Related works in these categories are listed. It should be noted that the references are by no means exhaustive.

1.1 Motion Models

Motion layers undergo coherent two-dimensional displacements that are modeled as global parametric motions in most existing works. Two-dimensional affine motion [1], [2], [4], [5], [6], [8], and 2D projective motion [9] models have been extensively investigated. Local motion models with more degrees of freedom have also been proposed [7] to describe more complex motions. The idea is to model each motion group as a linear combination of basis functions. Among the models that correctly describe the motions of the scenes, the ones with fewer parameters are generally preferred. In this work, a 2D rigid motion model that has only 2D translation and rotation components is investigated.

1.2 Segmentation Constraints

The process of decomposing an image into motion layers is called motion segmentation. Individual pixels are assigned to motion layers according to the color consistency measure induced by different motions. More specifically, in a two-frame motion layer formulation, each pixel in the reference view is assigned to the motion layer that best predicts its image intensity in the other image. Segmentations derived using this method are noisy due to image noise appearance changes, and matching ambiguities in the scene. These problems can be partially solved by imposing segmentation constraints to regularize the shapes of layers. Local smoothness models, such as the first order Markov random fields (MRF), have been previously investigated [6], [8]. The assumption behind the MRF model is that pixels spatially close to each other tend to be in the same layer. With this constraint, layers with regular boundary shapes are preferred and holes in the segmentation are suppressed. MRF-like constraints are not suitable for dynamics object tracking since estimation based on MRF constraints is computationally expensive and formulating a complete recursive tracking scheme is cumbersome. In order to reduce computational complexity but allow flexibility, in this paper, we propose a Gaussian prior function to handle objects with compact shapes. Such a model imposes a strong assumption about the overall shape of the object, but allows arbitrary variations within that assumption. The model is used only as a prior function for object shape. The actual segmentation of the object layer is the posterior function. This property will be extensively discussed throughout this paper.

1.3 Dynamic Motion, Segmentation, and Appearance Model of Layers

Most existing layer methods are limited to two-frame or multiframe batch formulations. When temporally related image frames are considered in a recursive formulation, additional dynamic constraints on layer motion, segmentation, and appearance are available. In this paper, we describe a new tracking formulation in which the MAP solution of the layer representation at the current time instant is estimated based on the previous time instants. A Markovian assumption simplifies the formulation by assuming that the parameters at the current time instant depend only on those at the previous time instant. Dynamic models and constraints on layer motion, segmentation, and appearance are proposed. We are not aware of any existing works on layer-based motion analysis incorporating these constraints into a single estimation formulation.

In our formulation, a dynamic motion model describes the temporal behaviors of the objects in a scene. For the rigid motion model, dynamic models such as the constant position model, the constant velocity model, and the constant acceleration model have been extensively investigated. For the applications described in this paper, we adopt a constant velocity model that will be further discussed in detail in Section 2.1.

The temporal constraints on layer segmentation, on the other hand, represent the dynamics of the shape changes over time. For the applications in which we are interested, it is known that the rough shapes of objects do not change dramatically. Therefore, a constant shape model is employed.

When multiple images are considered, constraints on the layer appearance need to be considered. A reasonable assumption that the appearances of objects remain unchanged or change slowly over time can be quantitatively described using a constant appearance model. A noise term is added to the model to allow for gradual changes in appearance in real scenarios.

1.4 Dynamic Layer Respresentation and Tracking

We define a dynamic layer representation at any time instant t as $\Lambda_t = (\Phi_t, \Theta_t, A_t)$, where Φ_t is the shape prior, Θ_t is the motion model, and A_t is the layer appearance. This representation is continuously estimated based on its value Λ_{t-1} at the previous time instant and the current image observation I_t. More specifically, the dynamic layer estimation problem is formulated as finding the maximum posterior probability

$$\max_{\Lambda_t} \arg P(\Lambda_t | I_t, \ldots, I_0, \Lambda_{t-1}, \ldots, \Lambda_0). \tag{1}$$

Using the Markovian assumption and Bayes' rule, this can be simplified as

$$\begin{aligned} &\max_{\Lambda_t} \arg P(\Lambda_t | I_t, \ldots, I_0, \Lambda_{t-1}, \ldots, \Lambda_0) \\ &= \max_{\Lambda_t} \arg P(\Lambda_t | I_t, I_{t-1}, \Lambda_{t-1}) \\ &= \max_{\Lambda_t} \arg P(I_t | \Lambda_t, I_{t-1}, \Lambda_{t-1}) P(\Lambda_t | I_{t-1}, \Lambda_{t-1}), \end{aligned} \tag{2}$$

where $P(I_t|\Lambda_t, I_{t-1}, \Lambda_{t-1})$ is the likelihood function and $P(\Lambda_t|I_{t-1}, \Lambda_{t-1})$ is the dynamic model of the state Λ_t. A solution can be obtained using the EM algorithm. Details will be discussed in Section 3.

Tracking with such a complete state representation is important for applications that utilize the appearance information of objects (video indexing and object recognition, for example). For applications requiring only position and geometric information, it produces more robust results than trackers that use partial representations only. For example, change-based trackers ignore the appearance information and thus have difficulty dealing with close-by or stationary objects. Template trackers typically update only motion parameters and, hence, can drift off or get attached to other objects of similar appearance [12]. Some template trackers use parametric transformations (affine, similarity, etc.) to update both the motion and the shape of the template [11]. However, since there is no explicit updating of template ownership, drift may still occur. The Transformed Hidden Markov Model (THMM) algorithm [15] includes both motion and appearance in its state representation and formulates the tracking problem as the MAP estimation of the whole temporal state sequence, whereas most existing trackers formulate the problem as an incremental one-step-at-a-time estimation problem. However, this advantage comes with the expense that the state (appearance and motion) has to be discrete and the number of possible states cannot be too large. THMM in its current form does not explicitly model segmentation and does not address the problem of tracking multiple objects. Multiple-hypothesis tracking methods [13], [14] solve the MAP problem in a batch mode. The computational complexity of these algorithms limits their state representations to simple motion information only, e.g., x and y positions of feature points and, also, precludes any (near) real-time implementations.

The rest of the paper is organized as follows: The details of the dynamic layer representation are presented in Section 2. Section 3 describes the MAP estimation of this representation. Some implementation issues and experimental results are shown in Section 4, which is followed by discussions and conclusions in Section 5.

2 Dynamic Layer Representation

In many practical situations, scenes as observed in image sequences can be completely described using the three components of a dynamic layer representation: motion, segmentation, and appearance. This is particularly true for airborne surveillance videos and ground-based videos with pan-tilt cameras. We will show that, under such conditions, the complete layer description can be analytically formulated and dynamically estimated.

2.1 Motion Model

The motion model describes the coherent motion of a layer in an image. Affine and projective motion models have been extensively investigated in the existing methods. An affine motion model has six parameters whereas a projective model has eight parameters. These correspond to the image transformations induced by physical planes or the motion of a pan-tilt camera and imaging conditions in which the scene is far away from the camera. The choice of a motion model depends on the application at hand. For object tracking in aerial videos, the displacement of the ground plane motion is modeled as a projective motion. With the background motion compensated, the motion of the foreground layer j at time instant t can be approximated by a 2D rigid motion which is described using a 2D translation vector $\dot{\mu}_{t,j}$ and a rotation $\dot{\omega}_{t,j}$. Such a motion model is a special case of the more general affine or projective models and is compactly specified by three parameters. The motion parameters for the layer j are then denoted as $\Theta_{t,j} = [\dot{\mu}_{t,j}^T, \dot{\omega}_{t,j}]^T$. Since vehicles move at relatively constant speeds, a commonly used 2D constant velocity model is adopted for modeling the dynamic behaviors of the layers over time. More specifically, given the motion $\Theta_{t-1,j}$ in the previous time instant, the current motion is described by a Gaussian distribution

$$P(\Theta_{t,j}|\Theta_{t-1,j}) = N(\Theta_{t,j} : \Theta_{t-1,j}, diag[\sigma_\mu^2, \sigma_\mu^2, \sigma_\omega^2]), \tag{3}$$

where $N(x : \mu, \sigma^2)$ denotes a normal distribution for a random variable x with mean μ and variance σ^2. σ_μ^2 and σ_ω^2 in the covariance matrix represent the model uncertainty in translation and rotation.

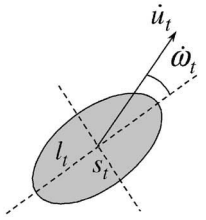

Fig. 1. The motion of a foreground object is described by a translation and a rotation. Its shape prior is modeled as a Gaussian distribution.

2.2 Dynamic Segmentation Prior

Segmentation of a scene into motion layers is typically achieved by assigning pixels to motion models that lead to the best image alignment for the corresponding layers. However, existing methods are limited in their ability to track layers over time. First, the resultant segmentation can be noisy due to motion ambiguities and image noise. Motion ambiguities occur when multiple motions give good predictions of the image intensities. This problem is frequently observed in textureless regions. Second, since motion segmentation is computed independently at each instant of time, motion layers may drift and eventually evolve into arbitrary shapes in the presence of clutter, occlusions, and ambiguous backgrounds. Third, domain knowledge regarding the shapes of layers is not considered in the model. Researchers have employed Markov random fields to address the first problem [6], [8] by imposing smoothness priors on the segmentations. However, we are not aware of any previous work that considers the other two problems.

We propose a dynamic Gaussian segmentation prior that encodes the domain knowledge that the foreground objects have compact shapes. We also model the dynamics of the segmentation prior so that gradual changes over time are allowed. The motivation for employing such a global parametric shape prior is twofold. First, the prior imposes a preference on the shape of a foreground layer and prevents the layer from evolving into an arbitrary shape in the course of tracking. As a result, it assists in tracking when ambiguous or cluttered measurements occur. Second, only the compact parametric form of the prior function needs to be estimated, which makes the estimation process computationally efficient. It is to be emphasized that the parametric representation of segmentation is used only as a compact way to represent a shape in motion. At each time instant, data association for each pixel in a new image is determined using both a motion alignment measure (as in traditional layer estimators) and the additional dynamic shape prior.

In the context of vehicle tracking from airborne platforms, the dominant image region is the ground. Its displacements can be accurately modeled as a projective motion. The prior function for each pixel belonging to the ground layer is a constant value β. Moving vehicles are the foreground layers. Their segmentation prior functions are modeled as Gaussian distributions. More specifically, the prior for each foreground layer j is $\gamma + \exp[-(x_i - \mu_{t,j})^T \Sigma_{t,j}^{-1}(x_i - \mu_{t,j})/2]$, where $\mu_{t,j}$ is the center of the distribution and $\Sigma_{t,j}$ is the covariance matrix that defines the span of the distribution. $x_i, i = 0, \ldots n-1$ is the image coordinates of the ith pixel. In Fig. 2, a cross-section of the prior functions for the background and a single foreground layer are illustrated. One of the consequences of this model is that pixels with larger distances from any foreground layer center will have a higher prior of belonging to the ground layer. This prior is combined with the image likelihood to produce the final segmentation. The constant γ is a small positive value. It allows pixels to belong to a foreground layer even if they are relatively far away from the layer center as long as their likelihood values are high. Therefore, γ represents the uncertainty of the layer shape. Including this uncertainty in the prior is important because the shapes of vehicles are not exactly elliptical and they change constantly over time.

In summary, suppose there are g motion layers and the layer 0 is the ground layer, then the prior function for a pixel x_i belonging to a layer j is defined as

$$\mathrm{L}_{t,j}(x_i) = \begin{cases} \gamma + \exp\left[-(x_i - \mu_{t,j})^T \Sigma_{t,j}^{-1}(x_i - \mu_{t,j})/2\right] & j = 1, \ldots, g-1 \\ \beta & j = 0. \end{cases} \tag{4}$$

The covariance matrix $\Sigma_{t,j}$ is defined as

$$\Sigma_{t,j} = R^T(-\omega_{t,j}) Diag\left[l_{t,j}^2, s_{t,j}^2\right] R(-\omega_{t,j}), \tag{5}$$

where $l_{t,j}$ and $s_{t,j}$ are proportional to the lengths of the major and the minor axes of the iso-probability contours and, thus, describe the shape of each foreground layer, as shown in Fig. 1. The translation $\mu_{t,j}$ and the rotation angle $\omega_{t,j}$ are motion parameters and will be discussed in the next section.

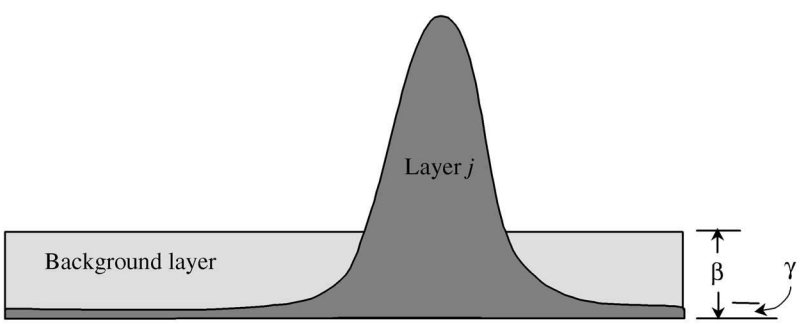

Fig. 2. A background+Gaussian segmentation prior function $L_{t,j}(x_i)$.

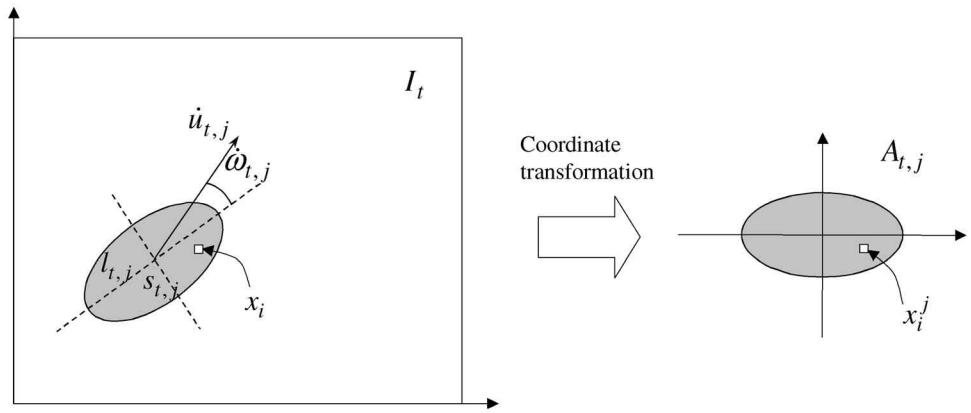

Fig. 3. The appearance image is defined in a local coordinate system determined by the motion parameters of layers.

$\Phi_{t,j} = [l_{t,j}, s_{t,j}]$ denotes the shape prior parameter of the layer j at time instant t.

The normalized prior distribution is computed as:

$$S_{t,j}(x_i) = L_{t,j}(x_i) \Big/ \sum_{j=0}^{g-1} L_{t,j}(x_i). \tag{6}$$

With the domain information that the airborne platform changes its altitude slowly and there is only a small amount of camera zoom, constancy of shape is used to describe the dynamic behavior of object shapes. The constancy of shape over time is modeled using a Gaussian distribution

$$P(\Phi_{t,j}|\Phi_{t-1,j}) = \mathrm{N}\Big(\Phi_{t,j} : \Phi_{t-1,j}, diag\big[\sigma_{ls}^2, \sigma_{ls}^2\big]\Big), \tag{7}$$

where the variance σ_{ls}^2 represents the uncertainty of the model.

It should be emphasized that the segmentation prior only imposes preference for certain shapes. The final segmentation is computed by combining both the likelihood function and the prior function. As a result, in this formulation, only the parameters of the shape prior need to be carried over time, instead of the propagation of arbitrary shape meshes.

2.3 Image Observation Model and Dynamic Layer Appearance Model

The appearance of layer j is denoted by $A_{t,j}$. It is in a local coordinate system that is defined by the center and the axes of the Gaussian segmentation prior. The coordinate transformation from the original image to this local coordinate system is $x_i^j = R(-\omega_j)(x_i - \mu_j)$. It is determined by the motion parameters of layer j (see Fig. 3). For any pixel x_i in the original image, the observation model for layer j is

$$P\Big(I_t(x_i)|A_{t,j}(x_i^j)\Big) = N\Big(I_t(x_i) : A_{t,j}(x_i^j), \sigma_I^2\Big), \tag{8}$$

where the variance σ_I^2 accounts for the noise in image intensity.

Appearances of the foreground objects and the ground layer change gradually over time. This domain information is encoded in the dynamic layer appearance model. In this model, the intensity value of a pixel in the layer j is a Gaussian distribution

$$P\Big(A_{t,j}(x_i^j)|A_{t-1,j}(x_i^j)\Big) = \mathrm{N}\Big(A_{t,j}(x_i^j) : A_{t-1,j}(x_i^j), \sigma_A^2\Big), \tag{9}$$

where σ_A^2 is the variance that represents the uncertainty of the model and accounts for the temporal changes in layer appearance.

3 EM Algorithm and the Layer Tracker

3.1 EM Algorithm

Our goal is to estimate the state of layers Λ_t at time t that maximizes the posterior probability

$$P(I_t|\Lambda_t, \Lambda_{t-1}, I_{t-1})P(\Lambda_t|\Lambda_{t-1}, I_{t-1})$$

(2). At every time instant t, we need to estimate a new segmentation and also update the layer parameters. There are two key problems that need to be solved: 1) the problem of data association that establishes the correspondences between pixels and layers and 2) the computation of the optimal layer parameters. The EM algorithm [16] can be used to solve both problems through explicitly computing hidden variables—the actual layer segmentation. According to the generalized EM algorithm, a local optimal solution can be achieved by iteratively optimizing or improving the following function Q with respect to Λ_t (see Appendix A for a proof).

$$\begin{aligned} Q = {} & E[\log P(I_t, z_t|\Lambda_t, \Lambda_{t-1}, I_{t-1})|I_t, \Lambda_t', \Lambda_{t-1}, I_{t-1}]+ \\ & \log P(\Lambda_t|\Lambda_{t-1}, I_{t-1}), \end{aligned} \tag{10}$$

where z_t is a hidden variable that indicates the association of each pixel to each layer and Λ_t' is the result of the previous iteration. As shown in Appendix B, this is equivalent to the iterative optimization or improvement of the function

$$\begin{aligned} & \sum_{i=0}^{n-1}\sum_{j=0}^{g-1} h_{i,j}\Big\{\log S_{t,j}(x_i^j) + \log P\Big(I_t(x_i)|A_{t,j}(x_i^j)\Big)\Big\}+ \\ & \sum_{j=1}^{g-1}\Big\{\log \mathrm{N}\big(\Phi_{t,j} : \Phi_{t-1,j}, diag\big[\sigma_{ls}^2, \sigma_{ls}^2\big]\big)+ \\ & \log \mathrm{N}\Big(\Theta_{t,j} : \Theta_{t-1,j}, diag\Big[\sigma_\mu^2, \sigma_\mu^2, \sigma_\omega^2\Big]\Big)+ \\ & \sum_{i=0}^{n-1} \log\Big(\mathrm{N}\big(A_{t,j}(x_i^j : A_{t-1,j}(x_i^j), \sigma_A^I\big)\Big)\Big\}, \end{aligned} \tag{11}$$

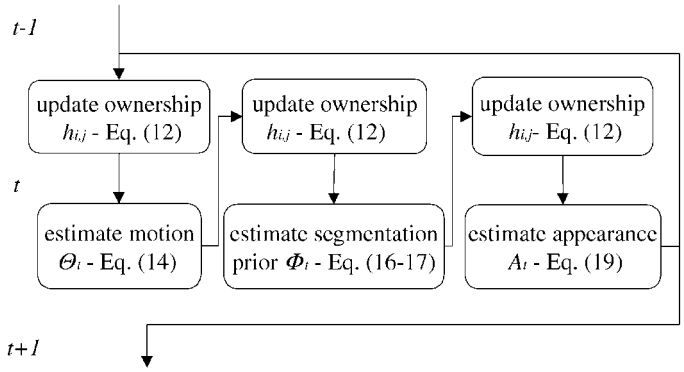

Fig. 4. The dynamic layer tracking algorithm.

where $h_{i,j}$ is the layer ownership—the posterior probability of the pixel x_i belonging to the layer j conditioned on Λ'_t. Though not used in the computation, the intermediate layer segmentation can be derived by choosing, for each pixel, the layer with the maximum ownership value.

3.2 Optimization

Since it is difficult to optimize Φ_t, Θ_t, and A_t simultaneously in (11), we adopt the strategy of improving each of them in turn with the other two fixed. This is the generalized EM algorithm and it can be proven that it converges to a local optimal solution. Fig. 4 summarizes the optimization process. As shown in the figure, motion parameters of the layers are computed first in each iteration. Then, the segmentation prior and the appearance are reestimated. The layer ownership $h_{i,j}$ needs to be updated whenever Φ_t, Θ_t, or A_t is reestimated. Multiple iterations are executed before proceeding to the next time instant. Individual steps are elaborated in the following sections.

3.2.1 Updating the Layer Ownership

The layer ownership $h_{i,j}$ is computed as

$$\begin{aligned} h_{i,j} &= P(z_t(x_i) = j | I_t, \Lambda'_t, \Lambda_{t-1}, I_{t-1}) \\ &= \frac{P(I_t | z_t(x_i) = j, \Lambda'_t, \Lambda_{t-1}, I_{t-1}) P(z_t(x_i) = j | \Lambda'_t, \Lambda_{t-1}, I_{t-1})}{P(I_t | \Lambda'_t, \Lambda_{t-1}, I_{t-1})} \\ &= P(I_t(x_i) | A'_{t,j}(x_i^j)) S_{t,j}(x_i)/Z. \end{aligned} \tag{12}$$

The first two terms are the likelihood function and the prior function defined in (8) and (6), respectively. The first term is the likelihood function that measures how well the image matches the appearance template; the second term is the prior function that describes the prior probability of pixel i belonging to layer j. Z normalizes $h_{i,j}$ so that $\sum_{j=0}^{g-1} h_{i,j} = 1$. The layer ownership $h_{i,j}$ is the posterior probability of the pixel i belonging to the layer j. Again, this equation illustrates that the actual segmentation $h_{i,j}$ is influenced both by the shape prior function and the image match measure.

3.2.2 Motion Estimation

If we assume that the shape prior Φ_t and the appearance A_t are known, the motion estimation step finds the motion Θ_t that improves

$$\begin{aligned} &\sum_{j=1}^{g-1} \log N\Big(\Theta_{t,j} : \Theta_{t-1,j}, diag\Big[\sigma_\mu^2, \sigma_\mu^2, \sigma_\omega^2\Big]\Big) + \\ &\sum_{i=0}^{n-1} \sum_{j=1}^{g-1} h_{i,j} \Big\{ \log S_{t,j}(x_i) + \log P\Big(I_t(x_i) | A_{t,j}(x_i^j)\Big)\Big\}. \end{aligned} \tag{13}$$

The motion of each individual foreground layer is estimated sequentially according to

$$\begin{aligned} &\min_{\Theta_{t,j}} \arg |\dot{\mu}_{t,j} - \dot{\mu}_{t-1,j}|/\sigma_\mu^2 + |\dot{\omega}_{t,j} - \dot{\omega}_{t-1,j}|/\sigma_\omega^2 - \\ &\sum_{i=0}^{n-1} 2h_{i,j} \log S_{t,j}(x_i) + \sum_{i=0}^{n-1} h_{i,j} \Big(I_t(x_i) - A_{t,j}(x_i^j)\Big)^2 / \sigma_I^2. \end{aligned} \tag{14}$$

The first term is the logarithm of the motion prior. The second term is the correlation between the layer ownership and the logarithm of the segmentation prior. The third term is the weighted sum of the squared differences between the image and the appearance of the layer j under motion $\Theta_{t,j}$. The solution is obtained by searching in the space of translation and rotation parameters. For the ground layer, the motion can be computed using a direct method like the one described in [10].

3.2.3 Shape Estimation

The shape Φ_t is estimated as

$$\begin{aligned} \max_{\Phi_t} \arg\ f = &\sum_{j-0}^{g-1} \log N\big(\Phi_{t,j} : \Phi_{t-1,j}, diag\big[\sigma_{ls}^2, \sigma_{ls}^2\big]\big) + \\ &\sum_{i=0}^{n-1} \sum_{j=0}^{g-1} h_{i,j} \log S_{t,j}(x_i). \end{aligned} \tag{15}$$

Gradient descent is used to optimize this function. As shown in Appendix C,

$$\begin{aligned} \frac{\partial f}{\partial l_{t,j}} = &\sum_{i=0}^{n-1} \frac{h_{i,j}(D(x_i) - L_{t,j}(x_i))}{L_{t,j}(x_i) D(x_i)} (L_{t,j}(x_i) - \gamma) y_{i,j.x}^2 / l_{t,j}^3 \\ &- (l_{t,j} - l_{t-1,j})/\sigma_{ls}^2 \end{aligned} \tag{16}$$

and similarly,

$$\begin{aligned} \frac{\partial f}{\partial s_{t,j}} = &\sum_{i=0}^{n-1} \frac{h_{i,j}\big(D(x_i) - L_{t,j}(x_i)\big)}{L_{t,j}(x_i) D(x_i)} (L_{t,j}(x_i) - \gamma) y_{i,j.y}^2 / s_{t,j}^3 \\ &- (s_{t,j} - s_{t-1,j})/\sigma_{ls}^2, \end{aligned} \tag{17}$$

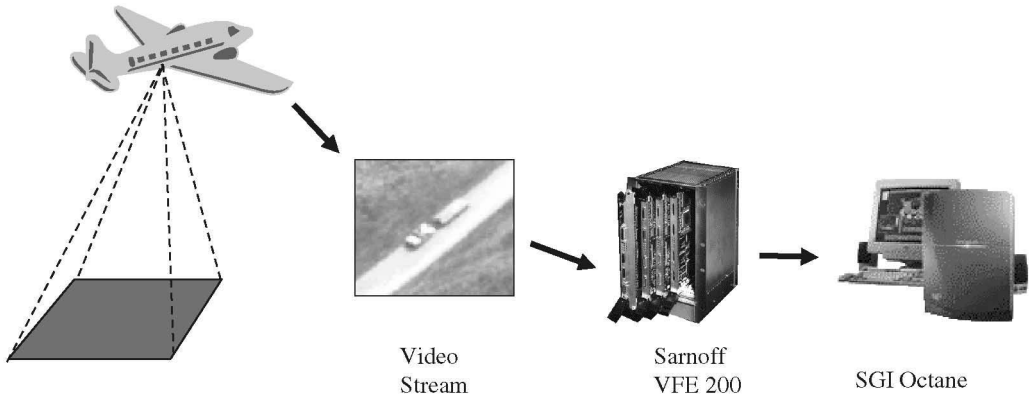

Fig. 5. The aerial video surveillance system.

where $D(x_i) = \sum_{j=0}^{g-1} L_{t,j}(x_i)$ and

$$[y_{i,j,x}, y_{i,j,y}]^T = R(-\omega)(x_i - \mu_j).$$

3.2.4 Appearance Estimation

The next step is to update the appearance model of each layer with Θ_t and Φ_t fixed according to

$$\max_{A_{t,j}} \arg \sum_{i=0}^{n-1} \left\{ \log\Big(N\big(A_{t,j}(x_i^j) : A_{t-1,j}(x_i^j), \sigma_A^2\big)\Big) + h_{i,j} \log P\Big(I_t(x_i) | A_{t,j}(x_i^j)\Big) \right\}. \tag{18}$$

From Appendix D, $A_{t,j}(x_i^j)$ is directly computed as

$$A_{t,j}(x_i^j) = \frac{A_{t,j}(x_i^j)/\sigma_A^2 + h_{i,j} I_t(x_i)/\sigma_I^2}{(1/\sigma_A^2 + h_{i,j}/\sigma_I^2)}. \tag{19}$$

This is the weighted average of the previous template and the current image. The weight is determined based on the ownership $h_{i,j}$ and the appearance variance σ_A^2. The update equation can be understood as follows: The larger $h_{i,j}$, the more certain that pixel i belongs to layer j. Therefore, the pixel contributes more to the appearance update of the layer j. In addition, the larger is the appearance variance σ_A^2, the less certain is the constant appearance model. Therefore, more weight is carried by the observation term $I_t(x_i)$.

4 Implementation and Experimental Results

The dynamic layer representation was initially developed for a real-time aerial video surveillance system. With a slight modification of the high-level control module and the fine-tuning of some parameters, it was later used for a ground-based video surveillance system where the primary task is tracking people and vehicles from a distance. In this paper, we will mostly concentrate on the vehicle tracking system. We call the core tracking component of the system the *layer tracker*. The performance of this tracker is compared with a correlation-based tracker and a change-based tracker. The intention of this comparison is to demonstrate the characteristics of the dynamic layer representation through real examples and qualitatively illustrate the advantages of employing such a complete representation in motion analysis. In addition, the results on tracking people from a distance are shown briefly to demonstrate the generality of the proposed dynamic layer representation.

4.1 Aerial Video Surveillance System (AVS)

We have developed a real-time aerial video surveillance system using the proposed dynamic layer tracker. The purpose of the system is to detect and track vehicles on the ground in real-time from moving airborne cameras. The overall system is illustrated in Fig. 5. Videos are taken from a camera mounted on an airplane or an unmanned aerial vehicle (UAV). The video stream is sent to a ground station through a wireless transmission channel. The videos then pass through the Sarnoff Video Front End (VFE) processor, which is a real-time system for video processing. The task of ground plane registration is performed in this system. The original video stream and the registration parameters are then fed into the layer tracker that resides on a workstation. A typical video frame from an AVS video is shown in Fig. 6a. The resolution of the video images is 320×240 pixels. The camera is moving and the sizes of the vehicles range from 10×10 to 40×40 pixels.

4.2 Initialization and Status Determination

Besides the core tracking algorithm described in Fig. 4, other issues that need to addressed are: 1) initialization of the layers, 2) deletion and addition of foreground layers, 3) determination of the status of an object as stationary, occluded, or disappeared. These tasks are handled in a separate module. The inputs to this module include the change blob images (Fig. 6b) and the estimates of the current layer representation. The change blob image is computed by aligning consecutive frames based on the background motion and computing the image intensity difference between them. At the center of this module is a state machine. As shown in Fig. 7, there are five different states that denote the state of objects at any given time instant. The states are: new object appearance, an object disappearance, a moving object, a stationary object, and an occluded object. They are linked by directed edges that represent the state transitions. The conditions for these transitions are marked along the edges.

New objects. A new object is initialized if a change blob is detected far away from any existing objects. When a new layer (vehicle) is added, an initialization step estimates the three components of a new motion layer from the change blob and the image. More specifically, the position of the

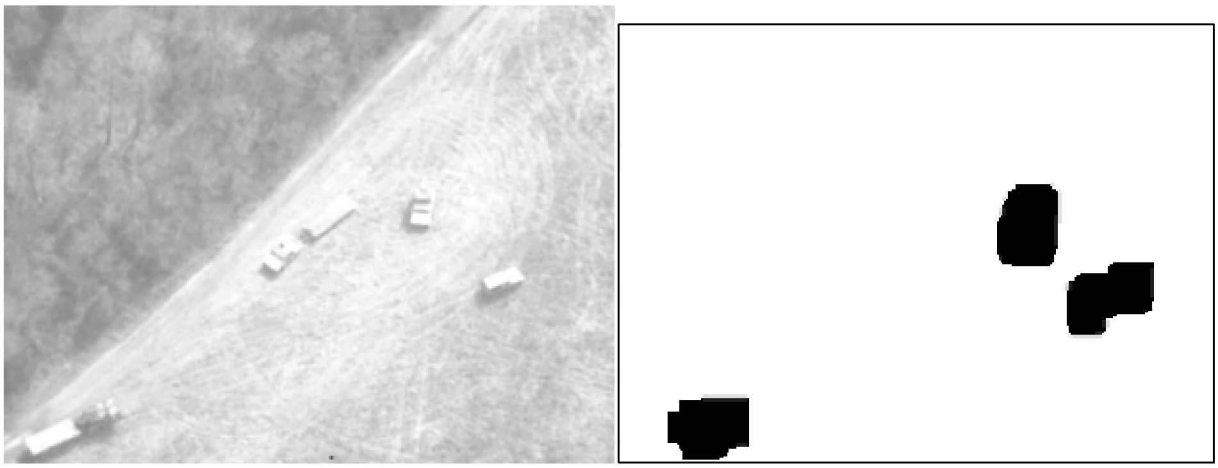

Fig. 6. (a) A typical frame from an aerial surveillance video and (b) its change blob image. Only three vehicles are moving.

object is located at the center of the blob. A zero velocity is assigned. The segmentation prior is estimated from the second order moments of the blob. The appearance is obtained from the original image.

Moving objects. In the course of tracking, objects stay in this state most of the time. The state of an object is transferred to moving if: 1) For a new object its associated motion blobs are continuously present, and the object is inside the image boundaries and 2) for a stationary or an occluded object, motion blobs reappear and the template matching score is high.

Object disappearance. An object is deleted if the following conditions are satisfied: 1) for a moving object, if it moves out of the image; 2) for a stationary object, if it moves out of the image (the camera moves) or the template matches poorly; 3) for an occluded object, if it moves out of the image or no motion blob is detected around it for a certain period of time.

Stationary objects. A moving object becomes stationary if no motion blob is detected around it, the template matching score is good, and the estimated motion is close to zero.

Occluded objects. A moving object becomes occluded if no motion blob is detected around it and the template matching score is poor.

4.3 A Real-Time Tracking System

The computational bottleneck in the real-time implementation of the proposed algorithm is the motion estimation step,

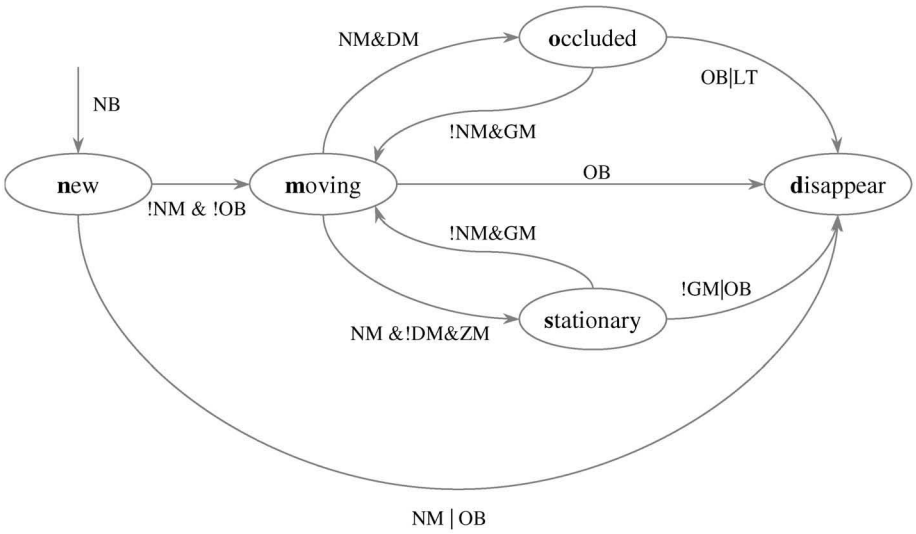

Fig. 7. State transition diagram for the dynamic layer tracker.

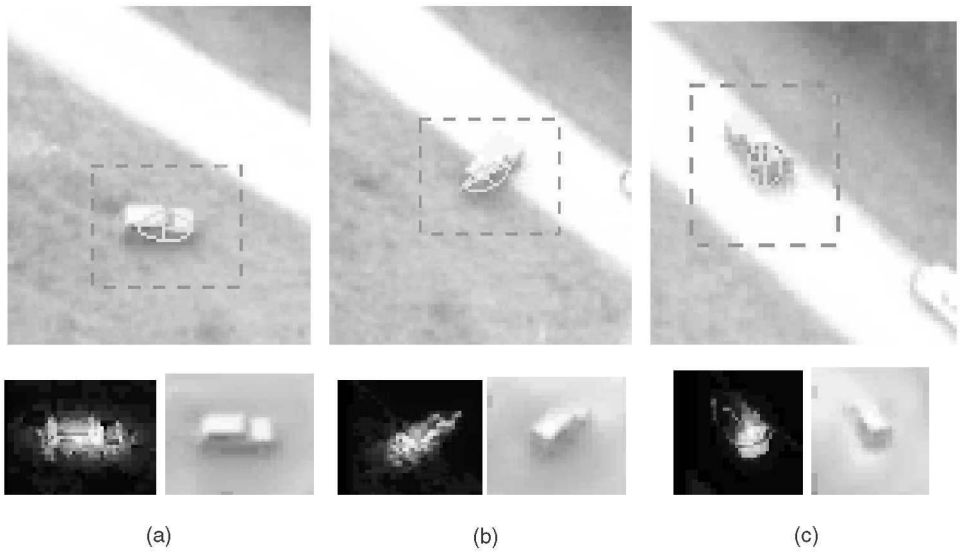

Fig. 8. Vehicle turning example using the layer tracker. The first row shows the cutouts of the original video frames and the Gaussian shape priors. The next row shows the segmentation and the appearance (warped to the image coordinates). (a) Frame 145, (b) frame 180, and (c) frame 210.

which accounts for more than 95 percent of the computation. In our implementation, the dominant background motion parameters are estimated at video rate using a VFE implementation of a direct method [10]. This information, together with the video frames, is then fed to a tracking system that runs on an SGI Octane workstation, where the foreground motion is estimated using a coarse-to-fine template matching method. A low resolution change blob image is also computed on the workstation. Though multiple iterations of the EM algorithm may be performed in each frame, we found that a single iteration is sufficient in practice. The current system can handle two moving objects at 10 Hz or four moving objects at 5 Hz.

4.4 Robust Tracking of Multiple Vehicles

A tracking system is designed to handle various motions and complex interactions such as passing and stopping (video clips of the experimental results presented in this section are available online at [17]). The design of the layer tracker is actually motivated by the fact that two of our existing trackers, a correlation-based tracker and a change-based tracker, failed to handle such difficult tracking tasks. The correlation-based tracker computes motion of foreground objects by correlating their appearance templates with the images. Once the motion is computed, the template is modified by linearly combining the old template and the new image evidence. The template is a rectangular window initialized manually in the first frame. The difference between the correlation-based tracker and the layer tracker is that the correlation-based tracker does not take into account the ownership of individual pixels in the correlation stage and the template update stage. Every pixel in the template window, whether it is background or foreground, is considered on an equal footing. Consequently, it is easily confused by background clutter or nearby foreground objects. The change-based tracker employs information contained in change blobs only. When a new change blob is detected, an object is initialized. The dynamic models of the blobs that include velocities and accelerations are estimated using a Kalman filter. One obvious problem with this type of tracker is that it cannot track an object when it becomes stationary. When motion blobs disappear, the tracker cannot determine if the object becomes stationary or disappears. Another problem is that, when objects are close to each other, their change blobs merge. When they split into multiple blobs later, motion is the only cue to infer their identities. This can be unreliable if the merge lasts an extended period of time. The layer tracker, on the other hand, handles these situations by considering appearance information during the tracking. Tracking results of the layer tracker will be demonstrated in this section along with those of the other two trackers. The results demonstrate the superiority of the layer tracker. We emphasize that the comparison with correlation and change trackers is for illustrative and demonstrative purposes only. We have not performed an exhaustive and quantitative comparison. Furthermore, although we did not use correlation and change trackers from other sources, we expect their performance to be similar to the ones we used for comparison.

In Fig. 8, the layer tracker results on a video clip with a turning vehicle are demonstrated. In this example, a vehicle in the scene turns 180 degrees within 6 seconds. Its appearance, shape, and motion change dramatically during this period of time. The layer tracker estimates them correctly and maintains the track. The estimated layer segmentation and appearance in three frames are shown. It can be observed that the appearance of the vehicle is adaptively updated over time. The correlation-based tracker (see Fig. 9), on the other hand, is distracted by the strong background texture and fails in frame 210.

In Fig. 10, the layer tracker results on vehicles passing from opposite directions are demonstrated. The passing is of a short duration, lasting less than one second. Since the two foreground layers have significantly different motion parameters, the segmentation task is relatively easy. The correlation-based tracker fails when the passing occurs (Figs. 11b and 11c)) because the nearby vehicle is included in the matching window and distracts the tracker. The

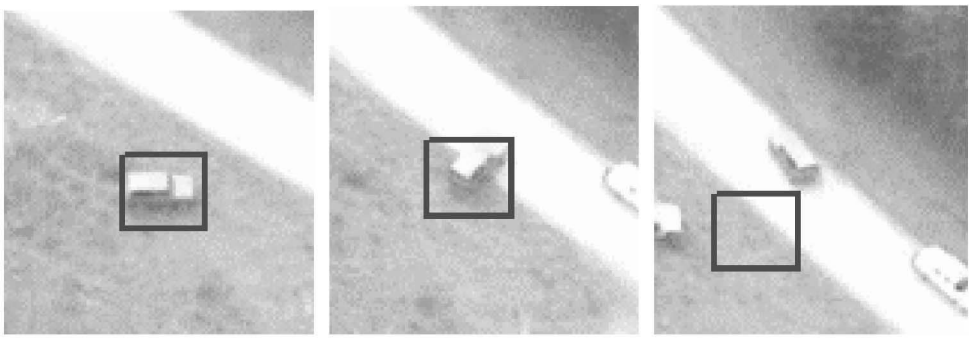

Fig. 9. Vehicle turning example with the correlation-based tracker. The tracker fails because of the fast appearance change of the vehicle and the cluttered background. (a) Frame 145, (b) frame 180, and (c) frame 210.

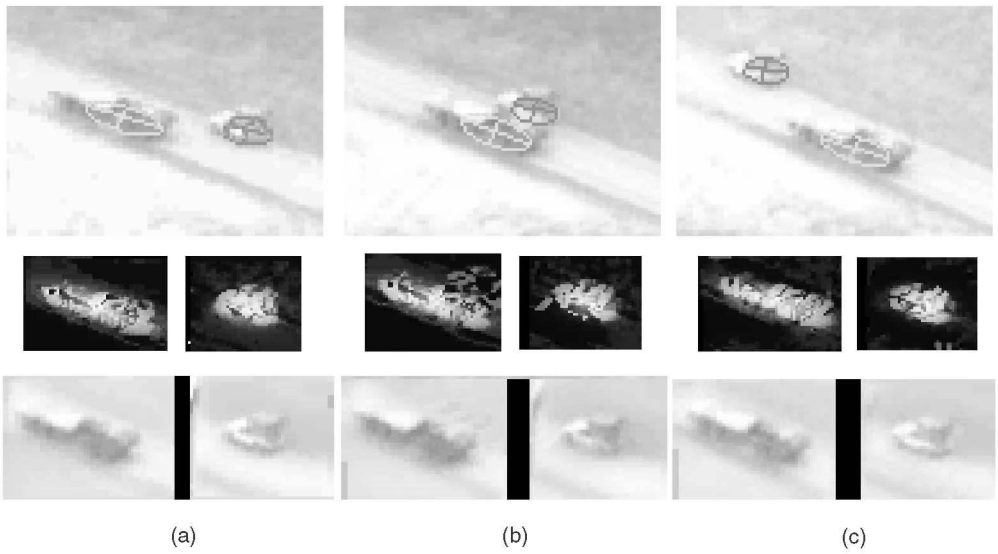

Fig. 10. Passing (opposite directions)—layer tracker. The first row shows the cutouts of the original video frames and the Gaussian shape priors. The next two rows show the segmentation and the appearance (warped to the image coordinates). (a) Frame 36, (b) frame 41, and (c) frame 49.

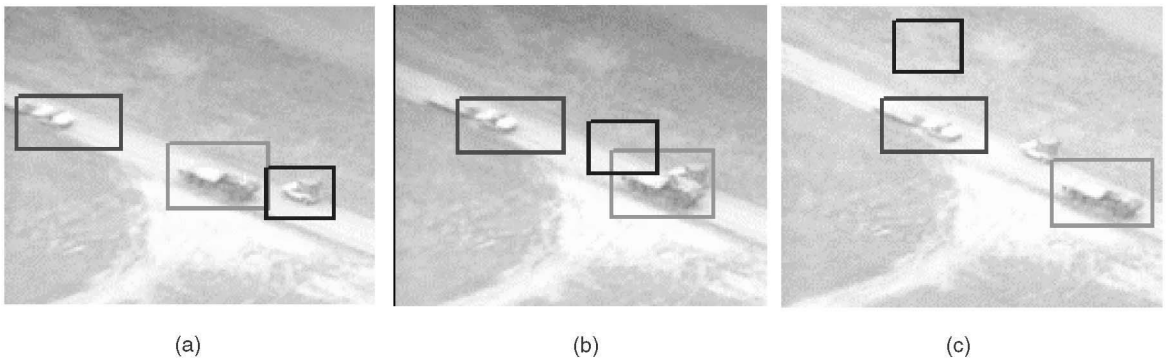

Fig. 11. Passing (opposite directions)—correlation-based tracker. The tracker fails when a nearby vehicle enters the matching window. (a) Frame 36, (b) frame 41, and (c) frame 49.

above two examples clearly demonstrate that both the appearance and the support of motion layers need to be estimated during tracking.

In Figs. 12 and 13, the tracking results on vehicles passing in the same direction are shown. The passing lasts about seven seconds. Compared to the previous passing sequence, this is more challenging because the vehicles remain close to each other longer and they have similar motions. In the layer tracker (Fig. 12), the appearance and the shape prior help the two layers maintain their shapes during the passing. This example demonstrates the importance of the global shape prior function. In the extreme case, if the passing lasts for an indefinitely long period of time, layer ownership cannot be determined solely by motion because both layers have the same motion. A change-based tracker works well when objects are far away from each other (Fig. 13a). When objects

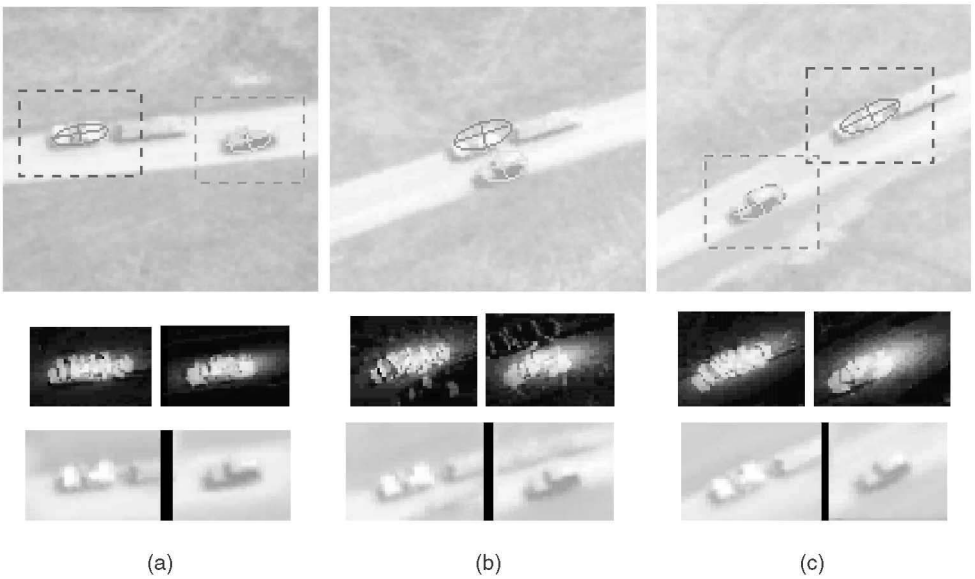

Fig. 12. Passing (same direction)—layer tracker. The first row shows the cutouts of the original video frames and the Gaussian shape priors. The next two rows show the segmentation and the appearance (warped to the image coordinates). (a) Frame 178, (b) frame 220, and (c) frame 253.

merge (Fig. 13b), their change blobs merge also. The motion information estimated from the merged blob is inaccurate. When the two vehicles split (Fig. 13c), the predicted location of one vehicle is far away from its actual position and the tracker fails. Layer tracker handles this example correctly because it accurately estimates the layer motion using the segmentation and the appearance information.

In Fig. 14, three vehicles are tracked. One of them eventually becomes stationary. This sequence demonstrates the importance of the layer appearance. A change-based tracker, which does not employ the layer appearance information, cannot handle this scenario because a stationary object does not create change blobs.

4.5 A Ground-Based Surveillance System

The proposed layer tracker is also being integrated into a ground-based surveillance system. The primary goal of the system is to monitor activities in an area covered by a ground-based stationary pan-tilt-zoom camera. Tracking moving objects, mainly people and automobiles, is a key element of the system. Our goal is to reliably track all the moving objects, such as people and vehicles, in the scene. The resolution of the video images is 320×240 pixels. The background is static because, for the examples we used, the camera is stationary. The size of people in images ranges from 5×5 to 40×40 pixels.

A potential problem in directly applying the proposed layer tracker is that the articulated motions of people walking and their changing 3D viewing angles violate the 2D rigid motion models. However, we found that in practice, when the object is at distance, the tracker still works reasonably well. There are several reasons that explain this phenomenon. First, most parts of a walking person, such as the torso and the head, undergo rigid motions. They account for a larger portion of the whole human body. Second, walking people rotate their bodies slowly compared to the video rate. When the object is at a distance, the pixelwise intensity change is gradual. The relatively slow appearance changes caused by such a transformation are captured in the appearance update step, which is controlled by the appearance uncertainty parameter σ_A^2. Third, the shapes of walking people are compact. Therefore, the Gaussian shape prior still applies.

However, parameters in the layer tracker need to be retuned for the ground-based tracking system. To compensate for the appearance change caused by rotations, the appearance variance σ_A^2, which represents the uncertainty of the constant appearance model, should be increased. The

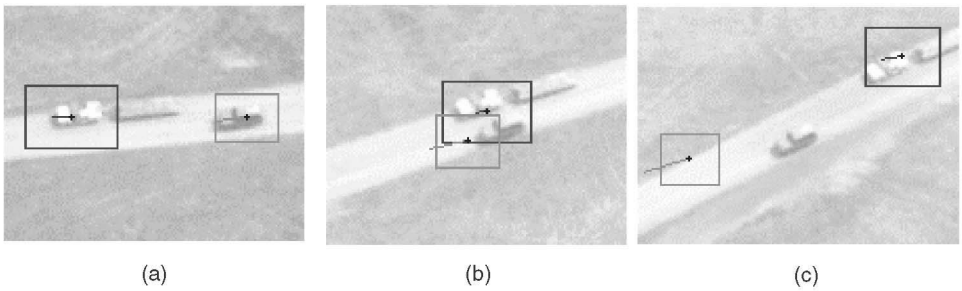

Fig. 13. Passing (same direction)—change-based tracker. The tracker fails during the passing because the motion cannot be accurately estimated. (a) Frame 178, (b) frame 220, and (c) frame 253.

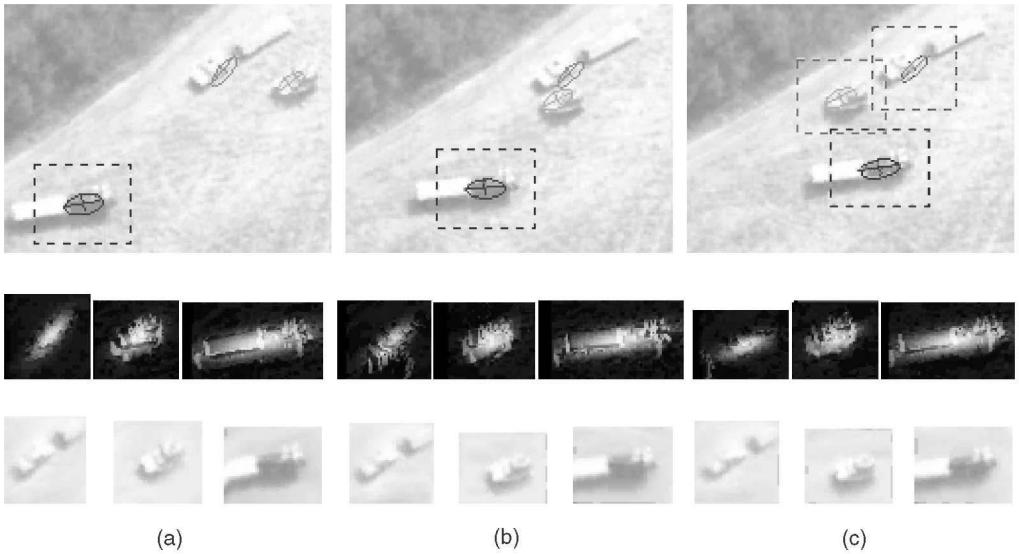

Fig. 14. Example of vehicle passing and stationary vehicles. The first row shows the cutouts of original video frames and the Gaussian shape priors. The next two rows show the segmentation and the appearance (warped to the image coordinates). (a) Frame 273, (b) frame 301, and (c) frame 321.

consequence is that, in the update stage, the image observations carry a larger weight and have a larger influence on the appearance template. Since, in the ground-based views, the ground is highly oblique, depending on the distance of the objects from the camera, the object sizes change more significantly than those in the aerial system. Therefore, a larger shape variance is needed to accommodate such size changes. The state transition machine needs to be tuned also. However, the performance of the tracker is less sensitive to those changes. Due to the page limits, we will not discuss the details further. Some results of the ground-based surveillance system are shown in Fig. 15.

5 Discussions and Conclusions

A dynamic layer representation and the associated estimation algorithm have been proposed in this paper. Compared to the traditional layer formulation, new extensions include the appearance model, the global segmentation prior, and the complete temporal consistency constraints (Table 1). In a sense, the new representation captures a complete representation of each layer in terms of motion, appearance, and shape. An estimation algorithm is proposed for this new representation using the EM algorithm in a MAP estimation framework. It provides a principled solution for the tracking problem.

One advantage of the proposed algorithm over many other trackers is that the ground layer and the objects compete with each other in the layer estimation using motion cues. This improves the robustness of the tracker against the background clutter and makes the tracking more resilient to distraction from other close-by objects.

The difference between the Gaussian segmentation prior and a Gaussian model in a model-based approach is that, in the latter, the actual pixelwise segmentation is

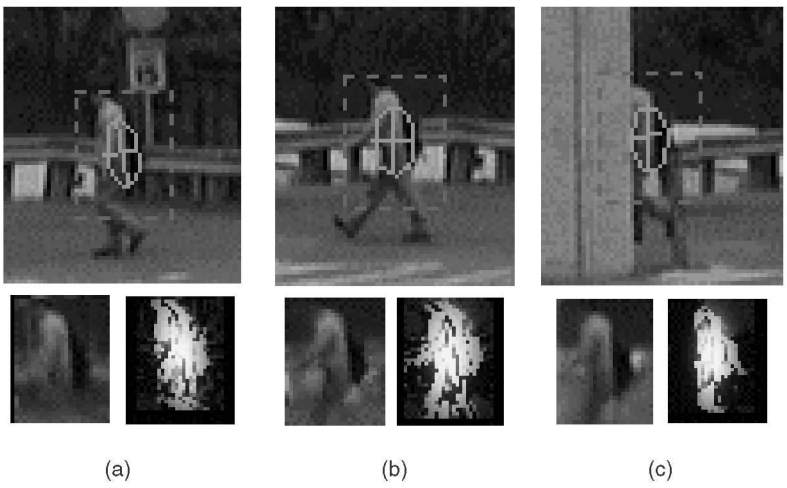

Fig. 15. Tracking people at distance using the layer tracker. The first row shows the cutouts of original video frames and the Gaussian shape priors (69×60 pixels). The next row shows the corresponding estimation of the appearance and the segmentation in (a) frame 16, (b) frame 28, and (c) frame 48.

not computed and, if the shape of the object is not similar to an ellipse, it will erroneously use the background pixel for motion estimation. In the proposed method, the global shape constraint acts as a segmentation prior and is a weaker constraint. The actual segmentation is still computed. Both the data-driven property of the layer approach and the efficiency of the model-based approach are preserved. An interesting question is how to incorporate more complicated segmentation priors for objects such as human forms into this framework.

APPENDIX A

Suppose x is a variable and its observation is y. If the distribution of x is governed by a parameter θ, then θ can be estimated by maximizing the posterior probability $P(\theta|y)$. In general, it is difficult to find a globally optimal solution to this problem. The generalized EM algorithm finds a local maximum by iteratively improving θ. The formulation and the sketch of the proof are described below (details can be found in [16]).

Suppose some initial estimation θ' is already available. We take the expectation of $\log P(\theta, y)$ with respect to the distribution $P(x|\theta', y)$. The result is still $\log P(\theta, y)$ since $\log P(\theta, y)$ is independent of variable x. In other words,

$$\log P(\theta, y) = E[\log P(\theta, y)|\theta', y]. \tag{I}$$

By applying the identity

$$\log P(y|\theta) = \log P(x, y|\theta) - \log P(x|\theta, y)$$

to (I), the right side is expanded as

$$\begin{aligned} &E[\log P(\theta, y)|\theta', y] \\ &= E[\log P(y|\theta)|\theta', y] + E[\log P(\theta)|\theta', y] \\ &= E[\log P(x, y|\theta)|\theta', y] - E[\log P(x|\theta, y)|\theta', y] + \\ &\quad E[\log P(\theta)|\theta', y]. \end{aligned} \tag{II}$$

Our goal is to find a new $\theta = \theta''$ to improve this quantity. We note without proof that the second term in (II) is minimized when $\theta = \theta'$, so any value θ'' will not decrease the second term. If θ'' also increases the other two terms, that is,

$$\begin{aligned} &E[\log P(x, y|\theta'')|\theta', y] + E[\log P(\theta'')|\theta', y] > \\ &\quad E[\log P(x, y|\theta')|\theta', y] + E[\log P(\theta')|\theta', y] \Leftrightarrow \\ &E[\log P(x, y|\theta'')|\theta', y] + \log P(\theta'') > \\ &\quad E[\log P(x, y|\theta')|\theta', y] + \log P(\theta'), \end{aligned} \tag{III}$$

then replacing θ' with θ'' improves $E[\log P(\theta, y)|\theta', y]$ or $P(\theta'', y) > P(\theta', y)$. Dividing both sides by $P(y)$, we get $P(\theta''|y) > P(\theta'|y)$. Therefore, any θ'' that satisfies (III) is an improved solution.

APPENDIX B

We assume that the segmentation prior of each pixel is independent of each other conditioned on the shape parameters, i.e.,

$$\log P(z_t|\Lambda_t, \Lambda_{t-1}, I_{t-1}) = \sum_{i=0}^{n-1} \log P(z_t(x_i)|\Lambda_t, \Lambda_{t-1}, I_{t-1}),$$

and the likelihood of each pixel belonging to a certain layer is independent of each other too, i.e.,

$$\log P(I_t|z_t, \Lambda_t, \Lambda_{t-1}, I_{t-1}) = \sum_{i=0}^{n-1} \log P(I_t(x_i)|\Lambda_t, \Lambda_{t-1}, I_{t-1}).$$

Then, the function Q in (10) can be expanded by explicitly computing the expectation

$$\begin{aligned} Q = &\sum_{i=0}^{n-1}\sum_{j=0}^{g-1} P(z_t(x_i) = j|I_t, \Lambda'_t, \Lambda_{t-1}, I_{t-1})\{ \\ &\log P(z_t(x_i) = j|\Lambda_t, \Lambda_{t-1}, I_{t-1}) + \\ &\log P(I_t(x_i)|z_t(x_i) = j, \Lambda_t, \Lambda_{t-1}, I_{t-1})\} + \log P(\Lambda_t|\Lambda_{t-1}). \end{aligned}$$

We denote $h_{i,j} = P(z_t(x_i) = j|I_t, \Lambda'_t, \Lambda_{t-1}, I_{t-1})$ as the conditional probability of pixel x belonging to layer j. It is the distribution over which the expectation is taken.

As the segmentation prior $P(z_t(x_i) = j|\Lambda_t, \Lambda_{t-1}, I_{t-1})$ equals $S_{t,j}(x_i)$ defined in (6),

$$\begin{aligned} Q = &\sum_{i=0}^{n-1}\sum_{j=0}^{g-1} h_{i,j}\Big\{\log S_{t,j}(x_i) + \\ &\log P\Big(I_t(x_i)|z_t(x_i) = j, \Lambda_t, \Lambda_{t-1}, I_{t-1}\Big)\Big\} + \log P(\Lambda_t|\Lambda_{t-1}) \\ = &\sum_{i=0}^{n-1}\sum_{j=0}^{g-1} h_{i,j}\Big\{\log S_{t,j}(x_i) + \log P\Big(I_t(x_i)|A_{t,j}(x_i^j)\Big)\Big\} + \\ &\log P(\Lambda_t|\Lambda_{t-1}). \end{aligned}$$

By substituting the shape, motion, and appearance priors from (3), (7), and (9), we obtain

$$\begin{aligned} &\log P(\Lambda_t|\Lambda_{t-1}) = \log P(\Phi_t, \Theta_t, A_t|\Phi_{t-1}, \Theta_{t-1}, A_{t-1}) \\ &= \sum_{\mathrm{j}=0}^{\mathrm{g}-1}\Big\{\log\Big(\big(\Phi_{t,j} : \Phi_{t-1,j}, \mathrm{diag}\big[\sigma_{ls}^2, \sigma_{ls}^2\big]\big) + \\ &\quad \log(N\Big(\Theta_{t,j} : \Theta_{t-1,j}, \mathrm{diag}\Big[\sigma_\mu^2, \sigma_\mu^2, \sigma_\omega^2\Big]\Big) + \\ &\quad \sum_{\mathrm{i}=0}^{\mathrm{n}-1} \log(N\big(A_{t,j}(x_i^j) : A_{t-1,j}(x_i^j), \sigma_A^2\big)\Big)\Big\}. \end{aligned}$$

Substitution of the above expression in the equation for Q results in the following:

$$\begin{aligned} Q = &\sum_{i=0}^{n-1}\sum_{j=0}^{g-1} h_{i,j}\Big\{\log S_{t,j}(x_i) + \log P\big(I_t(x_i)|A_{t,j}(x_i^j)\big)\Big\} + \\ &\sum_{j=0}^{g-1}\Big\{\log N\big(\Phi_t : \Phi_{t-1}, diag\big[\sigma_{ls}^2, \sigma_{ls}^2\big]\big) + \\ &\log N\Big(\Theta_{t,j} : \Theta_{t-1,j}, diag\Big[\sigma_\mu^2, \sigma_\mu^2, \sigma_\omega^2\Big]\Big) + \\ &\sum_{i=0}^{n-1} \log\Big(N\big(A_{t,j}(x_i^j) : A_{t-1,j}(x_i^j), \sigma_A^2\big)\Big)\Big\}. \end{aligned}$$

APPENDIX C

Taking the derivative of the objective function in (15), we have

$$\begin{aligned}
&\frac{\partial f}{\partial l_{t,j}} = \frac{-(l_{t,j}-l_{t-1,j})^2/2\sigma_{ls}^2}{\partial l_{t,j}} + \sum_{i=0}^{n-1}\frac{h_{i,j}}{S_{t,j}(x_i)}\frac{\partial S_{t,j}(x_i)}{\partial l_{t,j}} \\
&= -(l_{t,j}-l_{t-1,j})/\sigma_{ls}^2 - \\
&\quad 1/2\sum_{i=0}^{n-1}\frac{h_{i,j}D(x_i)}{L_{t,j}(x_i)}\frac{D(x_i)-L_{t,j}(x_i)}{D^2(x_i)}(L_{t,j}(x_i)-\gamma)\cdot \\
&\quad \frac{\partial (x_i-\mu_j)^T R^T(-\omega)Diag[1/l_{t,j}^2, 1/s_{t,j}^2]R(-\omega)(x_i-\mu_j)}{\partial l_{t,j}} \\
&= -(l_{t,j}-l_{t-1,j})/\sigma_{ls}^2 - \\
&\quad 1/2\sum_{i=0}^{n-1}\frac{h_{i,j}(D(x_i)-L_{t,j}(x_i))}{L_{t,j}(x_i)D(x_i)}(L_{t,j}(x_i)-\gamma) \\
&\quad \frac{\partial y_{i,j}^T Diag[1/l_{t,j}^2, 1/s_{t,j}^2]y_{i,j}}{\partial l_{t,j}} \\
&= -(l_{t,j}-l_{t-1,j})/\sigma_{ls}^2 + \sum_{i=0}^{n-1}\frac{h_{i,j}(D(x_i)-L_{t,j}(x_i))}{L_{t,j}(x_i)D(x_i)} \\
&\quad (L_{t,j}(x_i)-\gamma)y_{i,j,x}^2/l_{t,j}^3,
\end{aligned}$$

where $D(x_i) = \sum_{j=0}^{g-1} L_{t,j}(x_i)$ and

$$[y_{i,j,x}, y_{i,j,y}]^T = R(-\omega)(x_i-\mu_j).$$

APPENDIX D

Taking the derivative of the objective function in (18) with respect to the brightness value of each template pixel and setting the gradient equal to 0, we have

$$\begin{aligned}
&\frac{\partial}{\partial A_{t,j}(x_i^j)}\Big\{-(A_{t,j}(x_i^j)-A_{t-1,j}(x_i^j))^2/2\sigma_A^2 \\
&\qquad -h_{i,j}(I_t(x_i)-A_{t,j}(x_i^j))^2/2\sigma_I^2\Big\} \\
&= -\big(A_{t,j}(x_i^j)-A_{t-1,j}(x_i^j)\big)/\sigma_A^2 - h_{i,j}\big(A_{t,j}(x_i^j)-I_t(x_i)\big)/\sigma_I^2 \\
&= -(1/\sigma_A^2+h_{i,j}/\sigma_I^2)A_{t,j}(x_i^j) + A_{t-1,j}(x_i^j)/\sigma_A^2 + h_{i,j}I_t(x_i)/\sigma_I^2 \\
&= 0 \Leftrightarrow \\
&A_{t,j}(x_i^j) = \frac{A_{t-1,j}(x_i^j)/\sigma_A^2 + h_{i,j}I_t(x_i)/\sigma_I^2}{(1/\sigma_A^2+h_{i,j}/\sigma_I^2)}.
\end{aligned}$$

ACKNOWLEDGMENTS

This work was partly supported by US Defense Advanced Research Projects Agency grant DAAB07-98-C-J023. The authors would like to thank Dave Hirvonen, Supun Samarasekera, and Mike Hansen for their support in the development of this algorithm. This work was performed while Hai Tao was employed by the Sarnoff Corporation.

REFERENCES

[1] T. Darrell and A. Pentland, "Robust Estimation of Multi-Layered Motion Representation," *Proc. IEEE Workshop Visual Motion*, pp. 173-178, 1991.

[2] J.Y.A. Wang and E.H. Adelson, "Layered Representation for Motion Analysis," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 361-366, 1993.

[3] M. Irani and S. Peleg, "Motion Analysis for Image Enhancement: Resolution, Occlusion, and Transparency," *J. Visual Comm. and Image Representation*, vol. 4, no. 4 pp. 324-335, Dec. 1993.

[4] S. Hsu, P. Anandan, and S. Peleg, "Accurate Computation of Optical Flow by Using Layered Motion Representations," *Proc. Int'l Conf. Pattern Recognition*, 1994.

[5] S. Ayer and H.S. Sawhney, "Layered Representation of Motion Video Using Robust Maximum-Likelihood Estimation of Mixture Models and MDL Encoding," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 777-784, 1995.

[6] Y. Weiss and E.H. Adelson, "A Unified Mixture Framework for Motion Segmentation: Incorporating Spatial Coherence and Estimating the Number of Models," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 321-326, 1996.

[7] Y. Weiss, "Smoothness in Layers: Motion Segmentation Using Nonparametric Mixture Estimation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 520-526, 1997.

[8] N. Vasconcelos, "Emprical Bayesian EM-Based Motion Segmentation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 527-532, 1997.

[9] P.H.S. Torr, R. Szeliski, and P. Anandan, "An Integrated Bayesian Approach to Layer Extraction from Image Sequences," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 983-990, 1999.

[10] J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani, "Hiearchical Model-Based Motion Estimation," *Proc. Second European Conf. Computer Vision*, pp. 237-252, 1992.

[11] M.J. Black and Y. Yacoob, "Tracking and Recognizing Rigid and Non-Rigid Facial Motions Using Local Parametric Models of Image Motion," *Proc. Fifth Int'l Conf. Computer Vision, ICCV '95*, pp. 374-381, 1995.

[12] 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, 1996.

[13] D.B. Reid, "An Algorithm for Tracking Multiple Targets," *IEEE Trans. Automatic Control*, vol. 24, no. 6, pp. 843-854, Dec. 1979.

[14] I.J. Cox and S.L. Hingorani, "An Efficient Implementation of Reid's Multiple Hypothesis Tracking Algorithm and TTS Evaluation for the Purpose of Visual Tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 2, pp. 138-150, Feb. 1996.

[15] N. Jojic, N. Petrovic, B. Frey, and T.S. Huang, "Transformed Hidden Markov Models: Estimating Mixture Models of Images and Inferring Spatial Transformations in Video Sequences," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 26-33, 2000.

[16] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statistical Soc. B*, vol. 39, pp. 1-38, 1977.

[17] http://www.soe.ucsc.edu/~tao/LAYER/index.html.

Hai Tao received the BS and MS degrees in automation from Tsinghua University in 1991 and 1993, respectively. He received the MS degree in electrical engineering from Mississippi State University in 1995. He received the PhD degree in electrical engineering from the University of Illinois at Urbana-Champaign in January 1999. From 1999 to 2001, he was a member of the technical staff in the Vision Technology Laboratory at Sarnoff Corporation, New Jersey. Since July 2001, he has been with the Department of Computer Engineering at the University of California at Santa Cruz, where he is now an assistant professor. Dr. Tao's research interests include image and video processing, computer vision, vision-based computer graphics, and human-computer interaction. He has published more than 30 technical papers and two book chapters. He holds two US patents. He is a member of the IEEE Computer Society.

Harpreet S. Sawhney graduated from the Indian Institute of Technology, Kanpur with a Btech degree in electrical engineering in 1979 and an MTech degree in data communications in 1981. He received the PhD degree in computer science in 1992 from the University of Massachusetts, Amherst, focusing on computer vision. He is a senior member of the technical staff in the Vision Technologies Laboratory at the Sarnoff Corporation, where he has led R&D in image-based 3D modeling and manipulation for immersive tele-presence and enhanced visualization, video enhancement and indexing, and video mosaicing under a number of commercial and government programs since 1995. He led R&D in video annotation and indexing at the IBM Almaden Research Center from 1992 to 1995. He worked in hardware design with Hindustan Computers Ltd. (HCL), New Delhi, and in data communications with the Indian Space Research Organization's (ISRO) Satellite Center (ISAC), Bangalore, from 1981 to 1985. Dr. Sawhney has authored more than 40 technical publications, holds five patents, and has a number of patent applications pending. He is a member of the IEEE Computer Society.

Rakesh Kumar received the BTech degree from the Indian Institute of Technology, Kanpur, in 1983, the MS degree from the State University of New York, Buffalo, in 1985, and the PhD degree in computer science from the University of Massachusetts at Amherst in 1992 He is currently the head of the Media Vision Group at Sarnoff Corporation, Princeton, New Jersey. At Sarnoff, he has been directing commercial and government research and development projects in computer vision with a focus in the areas of immersive tele-presence and 3D modeling from images, image registration, video manipulation and exploitation. He is an associate editor for the *IEEE Transactions on Pattern Analysis and Machine Intelligence*. He is an author/coauthor of more than 25 technical publications and is a co-inventor of five patents. He is a member of the IEEE Computer Society.