MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Visual Tracking and Recognition Using Appearance-Adaptive Models in Particle Filters

Shaohua Zhou, Rama Chellappa, Baback Moghaddam

TR2004-028 December 2004

Abstract

We propose an approach that incorporates appearance-based models in a particle filter to realize robust visual tracking and recognition algorithms. In conventional tracking algorithms, the appearance model is either fixed or rapidly changing, and the motion model is simply a random walk with fixed noise variance. Also, the number of particles is typically fixed. All these factors make the visual tracker unstable. To stabilize the tracker, we propose the following features: an observation model arising from an adaptive appearance model, an adaptive velocity motion model with adaptive noise variance, and an adaptive number of particles. The adaptive-velocity model is derived using a first-order linear predictor based on the appearance difference between the incoming observation and the previous particle configuration. Occlusion analysis is implemented using robust statistics. Experimental results on tracking visual objects in long outdoor and indoor video sequences demonstrate the effectiveness and robustness of our tracking algorithm. We then perform simultaneous tracking and recognition by embedding them in one particle filter. For recognition purposes, we model the appearance changes between frames and gallery images by constructing the intra- and extra-personal spaces. Accurate recognition is achieved when confronted by pose and view variations.

IEEE Transactions on Image Processing, 13:11, pps. 1491-1506, 2004

Visual tracking and recognition using appearance-adaptive models in particle filters

Shaohua Kevin Zhou[1], Rama Chellappa[1], and Baback Moghaddam[2]

[1] Center for Automation Research (CfAR) and

Department of Electrical and Computer Engineering

University of Maryland, College Park, MD 20740

Email: {shaohua, rama}@cfar.umd.edu

[2] Mitsubishi Electric Research Laboratories (MERL)

201 Broadway, Cambridge, MA 02139

Email: {baback}@merl.com

Abstract

We present an approach that incorporates appearance-adaptive models in a particle filter to realize robust visual tracking and recognition algorithms. Tracking needs modeling inter-frame motion and appearance changes whereas recognition needs modeling appearance changes between frames and gallery images. In conventional tracking algorithms, the appearance model is either fixed or rapidly changing, and the motion model is simply a random walk with fixed noise variance. Also, the number of particles is typically fixed. All these factors make the visual tracker unstable. To stabilize the tracker, we propose the following modifications: an observation model arising from an adaptive appearance model, an adaptive velocity motion model with adaptive noise variance, and an adaptive number of particles. The adaptive-velocity model is derived using a first-order linear predictor based on the appearance difference between the incoming observation and the previous particle configuration. Occlusion analysis is implemented using robust statistics. Experimental results on tracking visual objects in long outdoor and indoor video sequences demonstrate the effectiveness and robustness of our tracking algorithm. We then perform simultaneous tracking and recognition by embedding them in a particle filter. For recognition purposes, we model the appearance changes between frames and gallery images by constructing the intra- and extra-personal spaces. Accurate recognition is achieved when confronted by pose and view variations.

Index Terms

Visual tracking, visual recognition, particle filtering, appearance-adaptive model, occlusion.

I. INTRODUCTION

Particle filtering [1] is an inference technique for estimating the unknown motion state, θ_t, from a noisy collection of observations, $Y_{1:t} = \{Y_1, ..., Y_t\}$ arriving in a sequential fashion. A state space model is often employed to accommodate such a time series. Two important components of this approach are state transition and observation models whose most general forms can be defined as follows:

$$\textit{State transition model:} \quad \theta_t = F_t(\theta_{t-1}, U_t), \tag{1}$$

$$\textit{Observation model:} \quad Y_t = G_t(\theta_t, V_t), \tag{2}$$

where U_t is the system noise, $F_t(.,.)$ characterizes the kinematics, V_t is the observation noise, and $G_t(.,.)$ models the observer. The particle filter approximates the posterior distribution $p(\theta_t|Y_{1:t})$ by a set of weighted particles $\{\theta_t^{(j)}, w_t^{(j)}\}_{j=1}^J$. Then, the state estimate $\hat{\theta}_t$ can either be the minimum mean square error (MMSE) estimate,

$$\hat{\theta}_t = \theta_t^{mmse} = E[\theta_t|Y_{1:t}] \approx J^{-1} \sum_{j=1}^{J} w_t^{(j)} \theta_t^{(j)}, \tag{3}$$

or the maximum a posteriori (MAP) estimate,

$$\hat{\theta}_t = \theta_t^{map} = \arg\max_{\theta_t} p(\theta_t|Y_{1:t}) \approx \arg\max_{\theta_t} w_t^{(j)}, \tag{4}$$

or other forms based on $p(\theta_t|Y_{1:t})$.

The state transition model characterizes the motion change between frames. In a visual tracking problem, it is ideal to have an exact motion model governing the kinematics of the object. In practice, however, approximate models are used. There are two types of approximations commonly found in the literature. (i) One is to learn a motion model directly from a training video [2], [3]. However such a model may overfit the training data and may not necessarily succeed when presented with testing videos containing objects arbitrarily moving at different times and places. Also one cannot always rely on the availability of training data. (ii) Secondly, a fixed constant-velocity model with fixed noise variance is fitted as in [4], [5], [6], [7].

$$\theta_t = \theta_{t-1} + U_t, \tag{5}$$

where U_t has a fixed noise variance of the form $U_t = R_0 * U_0$ with R_0 a fixed constant measuring the extent of noise and U_0 a 'standardized' random variable/vector [1]. If R_0 is small, it is very hard to model rapid movements; if R_0 is large, it is computationally inefficient since many more particles are needed to accommodate the large noise variance. All these factors make use of such a model ineffective. In this paper, we overcome this by introducing an adaptive-velocity model.

While contour is the visual cue used in many tracking algorithms [2], another class of tracking approaches [8], [9], [7] exploit an appearance model A_t. In its simplest form, we have the following observation equation[2],

$$Z_t = \mathcal{T}\{Y_t; \theta_t\} = A_t + V_t, \tag{6}$$

where Z_t is the image patch of interest in the video frame Y_t, parameterized by θ_t. In [8], a fixed template, $A_t = A_0$, is matched with observations to minimize a cost function in the form of sum of squared distance (SSD). This is equivalent to assuming that the noise V_t is a normal random vector with zero mean and a diagonal (isotropic) covariance matrix. At the other extreme, one could use a rapidly changing model [9], say, $A_t = \hat{Z}_{t-1}$, i.e., the 'best' patch of interest in the previous frame. However, a fixed template cannot handle appearance changes in the video, while a rapidly changing model is susceptible to drift. Thus, it is necessary to have a model which is a compromise between these two cases. In [10], Jepson *et. al.* proposed an online appearance model (OAM) for a robust visual tracker, which is a mixture of three components. Two EM algorithms are used, one for updating the appearance model and the other for deriving the tracking parameters.

Our approach to visual tracking is to make both observation and state transition models adaptive in the framework of a particle filter, with provisions for handling occlusion. The main features of our tracking approach are as follows:

- Appearance-based. The only visual cue used in our tracker is the 2-D appearance; i.e., we employ only image intensities, though in general features derived from image intensities, such as the phase information of the filter responses [10] or the Gabor feature graph

[1]Consider the scalar case for example. If U_t is distributed as $\mathsf{N}(0, \sigma^2)$, we can write $U_t = \sigma U_0$ where U_0 is standard normal $\mathsf{N}(0, 1)$. This also applies to multivariate cases.

[2]For the sake of simplicity, we denote: $Z_t = \mathcal{T}\{Y_t; \theta_t\}$, $Z_t^{(j)} = \mathcal{T}\{Y_t; \theta_t^{(j)}\}$, $\hat{Z}_t = \mathcal{T}\{Y_t; \hat{\theta}_t\}$. Also, we can always vectorize the 2-D image by a lexicographical scanning of all pixels and denote the number of pixels by d.

presentation [11], are also applicable. No prior object models are invoked. In addition, we only use gray scale images.

- Adaptive observation model. We adopt an appearance-based approach. The original online appearance model (OAM) is modified and then embedded in our particle filter. Therefore, the observation model is adaptive as the appearance A_t involved in Eq. (6) is adaptive.
- Adaptive state transition model. Instead of using a fixed model, we use an adaptive-velocity model, where the adaptive motion velocity is predicted using a first-order linear approximation based on the appearance difference between the incoming observation and the previous particle configuration. We also use an adaptive noise component, i.e, $U_t = R_t * U_0$, whose magnitude R_t is a function of the prediction error. It is natural to vary the number of particles based on the degree of uncertainty R_t in the noise component.
- Handling occlusion. Occlusion is handled using robust statistics [12], [8], [13]. We robustify the likelihood measurement and the adaptive velocity estimate by downweighting the 'outlier' pixels. If occlusion is declared, we stop updating the appearance model and estimating the motion velocity.

Video-based recognition needs to handle uncertainties in both tracking and recognition. While conventional methods [14] resolve these uncertainties separately, i.e. tracking followed by recognition, we have proposed in [7] a framework to model both uncertainties in a unified way to realize simultaneous tracking and recognition. As evidenced by the empirical results (on a relatively modest databases) in [7], this algorithm improves its recognition rate over the conventional ones without sacrificing accuracy in tracking.

We focus on face recognition in this paper. Though the time series formulation allows very general models, our earlier efforts invoked rather simple ones, which may yield unsatisfactory results in both tracking and recognition when confronted by severe pose and illumination variations. We improve our approach in the following three aspects: (i) Modeling the inter-frame motion and appearance changes within the video sequence; (ii) Modeling the appearance changes between the video frames and gallery images by constructing intra- and extra-personal spaces which can be treated as a 'generalized' version of discriminative analysis [15]; and (iii) Utilizing the fact that the gallery images are in frontal views. By embedding these in a particle filter, we are able to achieve a stabilized tracker and an accurate recognizer to handle pose and illumination variations.

This paper is organized as follows. We briefly review the related literature on visual tracking and particle filters in Section II. We examine the details of an adaptive observation model in Section III, with a special focus on the adaptive appearance model, and of an adaptive state transition model in Section IV with a special focus on how to calculate the motion velocity. Handling occlusion is discussed in Section V, and experimental results on tracking vehicles and human faces in Section VI. Simultaneous tracking and recognition is discussed in Section VII, with conclusions presented in Section VIII.

II. Related work on visual tracking and particle filters

A. Visual tracking

Roughly speaking, previous work on visual tracking can be divided into two groups: deterministic tracking and stochastic tracking. Our approach combines the merits of both stochastic and deterministic tracking approaches in a unified framework using a particle filter. We give below a brief review of both approaches.

Deterministic approaches usually reduce to an optimization problem, e.g., minimizing an appropriate cost function. The definition of the cost function is a key issue. A common choice in the literature is the SSD used in many optical flow approaches [8].[3] A gradient descent algorithm is most commonly used to find the minimum. Very often, only a local minimum can be reached. In [8], the cost function is defined as the SSD between the observation and a fixed template, and the motion is parameterized as affine. Hence the task is to find the affine parameter minimizing the cost function. Using a Taylor series expansion and keeping only the first-order terms, a linear prediction equation is obtained. It has been shown that for the affine case, the system matrix can be computed efficiently since a fixed template is used. Mean shift [16] is an alternative deterministic approach to visual tracking, where the cost function is derived from the color histogram.

Stochastic tracking approaches often reduce to an estimation problem, e.g., estimating the state for a time series state space model. Early works [17], [18] used the Kalman filter or its variants [19] to provide solutions. However, this restricts the type of model that can be used. Recently

[3]We note that using SSD is equivalent to using a model where the noise obeys an iid Gaussian distribution; therefore this case can also be viewed as stochastic tracking.

sequential Monte Carlo (SMC) algorithms [1], [20], [21], [22], which can model nonlinear/non-Gaussian cases, have gained prevalence in the tracking literature due in part to the CONDENSATION algorithm [2]. Stochastic tracking improves robustness over its deterministic counterpart by its capability for escaping local minimum since the search directions are for the most part random even though they are governed by a deterministic state transition model. Toyama and Blake [23] proposed a probabilistic paradigm for tracking with the following properties: Exemplars are learned from the raw training data and embedded in a mixture density; The kinematics is also learned; The likelihood measurement is constructed on a metric space. Other approaches are also discussed in Section II-B. However, as far as the computational load is concerned, stochastic algorithms in general are more intense. Note that the stochastic approaches often lead to optimization problems too.

B. Particle Filter

General algorithm: Given the state transition model in (1) characterized by the state transition probability $p(\theta_t|\theta_{t-1})$ and the observation model in (2) characterized by the likelihood function $p(Y_t|\theta_t)$, the problem is reduced to computing the posterior probability $p(\theta_t|Y_{1:t})$ The nonlinearity/nonnormality in (1) and (2) result in Kalman filter [19] being ineffective. The particle filter is a means to approximate the posterior distribution $p(\theta_t|Y_{1:t})$ by a set of weighted particles $\mathcal{S}_t = \{\theta_t^{(j)}, w_t^{(j)}\}_{j=1}^J$ with $\sum_{j=1}^J w_t^{(j)} = 1$. It can be shown [20] that $\mathcal{S}_t$ is *properly weighted* with respect to $p(\theta_t|Y_{1:t})$ in the sense that, for every bounded function $h(.)$,

$$\lim_{J\to\infty} \sum_{j=1}^{J} w_t^{(j)} h(\theta_t^{(j)}) = E_p[h(\theta_t)]. \tag{7}$$

Given $\mathcal{S}_{t-1} = \{\theta_{t-1}^{(j)}, w_{t-1}^{(j)}\}_{j=1}^J$ which is properly weighted with respect to $p(\theta_{t-1}|Y_{1:t-1})$, we first resample $\mathcal{S}_{t-1}$ to reach a new set of samples with equal weights $\{\theta_{t-1}^{'(j)}, 1\}_{j=1}^J$. We then draw samples $\{U_t^{(j)}\}_{j=1}^J$ for U_t and propagate $\theta_{t-1}^{'(j)}$ to $\theta_t^{'(j)}$ by Eq. (1). The new weight is updated as

$$w_t \propto p(Y_t|\theta_t) \tag{8}$$

The complete algorithm is summarized in Fig. 1.

Variations of Particle Filters: Sequential Importance Sampling (SIS) [20], [24] draws particles from a *proposal distribution* $g(\theta_t|\theta_{t-1}, Y_{1:t})$ and then for each particle a proper weight is assigned as follows:

$$w_t \propto p(Y_t|\theta_t)p(\theta_t|\theta_{t-1})/g(\theta_t|\theta_{t-1}, Y_{1:t}). \tag{9}$$

Initialize *a sample set* $\mathcal{S}_0 = \{\theta_0^{(j)}, 1)\}_{j=1}^{J}$ *according to prior distribution* $p(\theta_0)$.
For $t = 1, 2, \ldots$
  For $j = 1, 2, \ldots, J$
    Resample $\mathcal{S}_{t-1} = \{\theta_{t-1}^{(j)}, w_{t-1}^{(j)}\}$ *to obtain a new sample* $(\theta_{t-1}^{'(j)}, 1)$.
    Predict *the sample by drawing* $U_t^{(j)}$ *for* U_t *and computing* $\theta_t^{(j)} = F_t(\theta_{t-1}^{'(j)}, U_t^{(j)})$.
    Compute *the transformed image* $Z_t^{(j)}$.
    Update *the weight using* $w_t^{(j)} = p(Y_t|\theta_t^{(j)}) = p(Z_t^{(j)}|\theta_t^{(j)})$.
  End
  Normalize *the weight using* $w_t^{(j)} = w_t^{(j)} / \sum_{j=1}^{J} w_t^{(j)}$.
End

Fig. 1. The general particle filter algorithm.

Selection of the proposal distribution $g(\theta_t|\theta_{t-1}, Y_{1:t})$ is usually dependent on the application. For example, in the ICONDENSATION algorithm [25] which fuses low-level and high-level visual cues in the conventional CONDENSATION algorithm [2], the proposal distribution, a fixed Gaussian distribution for low-level color cue, is used to predict the particle configurations, then the posterior distribution of the high-level shape cue is approximated using SIS. It is interesting to note that two different cues can be even combined together into one state vector to yield a robust tracker, using the co-inference algorithm [6] and the approach proposed in [26]. We also use a prediction scheme but our prediction is based on the same visual cue i.e. the appearance in the image, and it is directly used in the state transition model rather than used as a proposal distribution. Additional visual cues are not used.

III. ADAPTIVE OBSERVATION MODEL

The adaptive observation model arises from the adaptive appearance model A_t. We use a modified version of OAM as developed in [10]. The differences between our appearance model and the original OAM are highlighted below.

A. Mixture appearance model

The original OAM assumes that the observations are explained by different causes, thereby indicating the use of a mixture density of components. In the original OAM presented in [10], three components are used, namely the W-component characterizing the two-frame variations,

the S-component depicting the stable structure within all past observations (though it is slowly-varying), and the L-component accounting for outliers such as occluded pixels.

We modify the OAM to accommodate our appearance analysis in the following aspects. (i) We directly use the image intensities while they use phase information derived from the image intensities. Direct use of the image intensities is computationally more efficient than using the phase information that requires filtering and visually more interpretable. (ii) As an option, in order to further stabilize the tracker one could use an F-component which is a fixed template that one is expecting to observe most often. For example, in face tracking this could be just the facial image as seen from a frontal view. In the sequel, we derive the equations as if there is an F-component. However, the effect of this component can be ignored by setting its initial mixing probability to zero. (iii) We embed the appearance model in a particle filter to perform tracking while they use the EM algorithm. (iv) In our implementation, we do not incorporate the L-component because we model the occlusion in a different manner (using robust statistics) as discussed in Sec. V.

We now describe the mixture appearance model. The appearance model at time t, $A_t = \{W_t, S_t, F_t\}$, is a time-varying one that models the appearances present in all observations up to time $t-1$. It obeys a mixture of Gaussians, with W_t, S_t, F_t as mixture centers $\{\mu_{i,t};\ i = w, s, f\}$ and their corresponding variances $\{\sigma^2_{i,t};\ i = w, s, f\}$ and mixing probabilities $\{m_{i,t};\ i = w, s, f\}$. Notice that $\{m_{i,t}, \mu_{i,t}, \sigma^2_{i,t};\ i = w, s, f\}$ are 'images' consisting of d pixels that are assumed to be independent of each other.

In summary, the observation likelihood is written as

$$p(Y_t|\theta_t) = p(Z_t|\theta_t) = \prod_{j=1}^{d}\{\sum_{i=w,s,f} m_{i,t}(j)\mathsf{N}(Z_t(j); \mu_{i,t}(j), \sigma^2_{i,t}(j))\}, \tag{10}$$

where $\mathsf{N}(x; \mu, \sigma^2)$ is a normal density

$$\mathsf{N}(x; \mu, \sigma^2) = (2\pi\sigma^2)^{-1/2}\exp\{-\rho(\frac{x-\mu}{\sigma})\}, \quad \rho(x) = \frac{1}{2}x^2. \tag{11}$$

B. Model update

To keep our paper self-contained, we show how to update the current appearance model A_t to A_{t+1} after $\hat{Z}_t$ becomes available, i.e., we want to compute the new mixing probabilities, mixture centers, and variances for time $t+1$, $\{m_{i,t+1}, \mu_{i,t+1}, \sigma^2_{i,t+1};\ i = w, s, f\}$.

It is assumed that the past observations are exponentially 'forgotten' with respect to their contributions to the current appearance model. Denote the exponential envelop by $\mathcal{E}_t(k) = \alpha \exp(-\tau^{-1}(t-k))$ for $k \leq t$, where $\tau = n_h / \log 2$, n_h is the half-life of the envelope in frames, and $\alpha = 1 - \exp(-\tau^{-1})$ to guarantee that the area under the envelope is 1. We just sketch the updating equations as follows and refer the interested readers to [10] for technical details and justifications.

The EM algorithm [27] is invoked. Since we assume that the pixels are independent of each other, we can deal with each pixel separately. The following computation is valid for $j = 1, 2, \ldots, d$ where d is the number of pixels in the appearance model.

Firstly, the posterior responsibility probabilities are computed as

$$o_{i,t}(j) \propto m_{i,t}(j)\mathsf{N}(\hat{Z}_t(j); \mu_{i,t}(j), \sigma^2_{i,t}(j)); \quad i = w, s, f, \quad \& \sum_{i=w,s,f} o_{i,t}(j) = 1. \tag{12}$$

Then, the mixing probabilities are updated as

$$m_{i,t+1}(j) = \alpha \; o_{i,t}(j) + (1-\alpha) \; m_{i,t}(j); \quad i = w, s, f, \tag{13}$$

and the first- and second-moment images $\{M_{p,t+1};\ p = 1, 2\}$ are evaluated as

$$M_{p,t+1}(j) = \alpha \; \hat{Z}^p_t(j) o_{s,t}(j) + (1-\alpha) \; M_{p,t}(j); \quad p = 1, 2. \tag{14}$$

Finally, the mixture centers and the variances are updated as:

$$S_{t+1}(j) = \mu_{s,t+1}(j) = \frac{M_{1,t+1}(j)}{m_{s,t+1}(j)}, \quad \sigma^2_{s,t+1}(j) = \frac{M_{2,t+1}(j)}{m_{s,t+1}(j)} - \mu^2_{s,t+1}(j). \tag{15}$$

$$W_{t+1}(j) = \mu_{w,t+1}(j) = \hat{Z}_t(j), \quad \sigma^2_{w,t+1}(j) = \sigma^2_{w,1}(j), \tag{16}$$

$$F_{t+1}(j) = \mu_{f,t+1}(j) = F_1(j), \quad \sigma^2_{f,t+1}(j) = \sigma^2_{f,1}(j). \tag{17}$$

C. Model initialization

To initialize A_1, we set $W_1 = S_1 = F_1 = T_0$ (with T_0 supplied by a detection algorithm or manually), $\{m_{i,1}, \sigma^2_{i,1};\ i = w, s, f\}$, and $M_{1,1} = m_{s,1} T_0$ and $M_{2,1} = m_{s,1}\sigma^2_{s,1} + T_0^2$.

IV. Adaptive state transition model

The state transition model we use incoporates a term for modeling adaptive velocity. The adaptive velocity is calculated using a first-order linear prediction method based on the appearance difference between two successive frames. The previous particle configuration is incorporated in the prediction scheme.

Construction of the particle configuration involves the costly computation of image warping (in the experiments reported here, it usually accounts for about half of the computations). In a conventional particle filtering algorithm, the particle configuration is used only to update the weight, i.e., computing weight for each particle by comparing the warped image with the online appearance model using the observation equation. But, our approach in addition uses the particle configuration in the state transition equation. In some sense, we 'maximally' utilize the information contained in the particles (without wasting the costly computation of image warping) since we use it in both state and observation models.

In [28], random samples are guided by deterministic search. Momentum for each particle is computed as the sum of absolute difference between two frames. If the momentum is below a threshold, a deterministic search is first performed using a gradient descent method and a small number of offsprings is then generated by stochastic diffusion; otherwise, stochastic diffusion is performed to generate a large number of offsprings. The stochastic diffusion is based on a second-order autoregressive process. But, the gradient descent method does not utilize the previous particle configuration in its entirety. Also, the generated particle configuration could severely deviate from the second-order autoregressive model, which clearly implies the need for an adaptive model.

A. Adaptive velocity

With the availability of the sample set $\Theta_{t-1} = \{\theta_{t-1}^{(j)}\}_{j=1}^{J}$ and the image patches of interest $\mathcal{Z}_{t-1} = \{Z_{t-1}^{(j)}\}_{j=1}^{J}$, for a new observation Y_t, we can predict the shift in the motion vector (or adaptive velocity) $\nu_t = \theta_t - \hat{\theta}_{t-1}$ using a first-order linear approximation [8], [29], [30], [31], which essentially comes from the constant brightness constraint, i.e., there exists a θ_t such that

$$\mathcal{T}\{Y_t; \theta_t\} \simeq \hat{Z}_{t-1}. \tag{18}$$

Approximating $\mathcal{T}\{Y_t;\theta_t\}$ using a first-order Taylor series expansion around $\tilde{\theta}_t$ (we set $\tilde{\theta}_t = \hat{\theta}_{t-1}$) yields

$$\mathcal{T}\{Y_t;\theta_t\} \simeq \mathcal{T}\{Y_t;\tilde{\theta}_t\} + C_t(\theta_t - \tilde{\theta}_t) = \mathcal{T}\{Y_t;\tilde{\theta}_t\} + C_t\nu_t, \tag{19}$$

where C_t is the Jacobian matrix.

Combining (18) and (19) gives

$$\hat{Z}_{t-1} \simeq \mathcal{T}\{Y_t;\tilde{\theta}_t\} + C_t\nu_t, \tag{20}$$

i.e.,

$$\nu_t = \theta_t - \tilde{\theta}_t \simeq -B_t(\mathcal{T}\{Y_t;\tilde{\theta}_t\} - \hat{Z}_{t-1}), \tag{21}$$

where B_t is the pseudo-inverse of the C_t matrix, which can be efficiently estimated from the available data Θ_{t-1} and $\mathcal{Z}_{t-1}$.

Specifically, to estimate B_t we stack into matrices the differences in motion vectors and image patches, using $\hat{\theta}_{t-1}$ and $\hat{Z}_{t-1}$ as pivotal points:

$$\Theta^{\delta}_{t-1} = [\theta^{(1)}_{t-1} - \hat{\theta}_{t-1},\ \ldots,\ \theta^{(J)}_{t-1} - \hat{\theta}_{t-1}], \tag{22}$$

$$\mathcal{Z}^{\delta}_{t-1} = [Z^{(1)}_{t-1} - \hat{Z}_{t-1},\ \ldots,\ Z^{(J)}_{t-1} - \hat{Z}_{t-1}]. \tag{23}$$

The least square (LS) solution for B_t is

$$B_t = (\Theta^{\delta}_{t-1}\mathcal{Z}^{\delta\,\mathsf{T}}_{t-1})(\mathcal{Z}^{\delta}_{t-1}\mathcal{Z}^{\delta\,\mathsf{T}}_{t-1})^{-1}, \tag{24}$$

where $(.)^{\mathsf{T}}$ means matrix transposition. However, it turns out that the matrix $\mathcal{Z}^{\delta}_{t-1}\mathcal{Z}^{\delta\,\mathsf{T}}_{t-1}$ is very often rank-deficient due to the high dimensionality of the data (unless the number of the particles at least exceeds the data dimension). To overcome this, we use the singular value decomposition (SVD).

$$\mathcal{Z}^{\delta}_{t-1} = USV^{\mathsf{T}} \tag{25}$$

It can be easily shown that

$$B_t = \Theta^{\delta}_{t-1}VS^{-1}U^{\mathsf{T}}. \tag{26}$$

To gain some computational efficiency, we can further approximate

$$B_t = \Theta^{\delta}_{t-1}V_qS^{-1}_qU^{\mathsf{T}}_q, \tag{27}$$

by retaining the top q components. Notice that if only a fixed template is used [29], the B matrix is fixed and pre-computable. But, in our case, the appearance is changing so that we have to compute the B_t matrix in each time step.

In practice, one may run several iterations till $\tilde{Z}_t = \mathcal{T}\{Y_t; \tilde{\theta}_t + \nu_t\}$ stabilizes, i.e., the error ϵ_t defined below is small enough.

$$\epsilon_t = \phi(\tilde{Z}_t, A_t) = \frac{2}{d}\sum_{j=1}^{d}\{\sum_{i=w,s,f} m_{i,t}(j)\rho(\frac{\tilde{Z}_t(j) - \mu_{i,t}(j)}{\sigma_{i,t}(j)})\}. \tag{28}$$

In (28), ϵ_t measures the distance between $\mathcal{T}\{Y_t; \tilde{\theta}_t + \nu_t\}$ and the updated appearance model A_t. The iterations proceed as follows: We initially set $\tilde{\theta}_t^1 = \hat{\theta}_{t-1}$. For the first iteration, we compute ν_t^1 as usual. For the k^{th} iteration, we use the predicted $\tilde{\theta}_t^k = \tilde{\theta}_t^{k-1} + \nu_t^{k-1}$ as a pivotal point for the Taylor expansion in (19) and the rest of the calculation then follows. It is rather beneficial to run several iterations especially when the object moves very fast in two successive frames since $\hat{\theta}_{t-1}$ might cover the target in Y_t in a small portion. After one iteration, the computed ν_t might be not accurate, but indicates a good minimization direction. Using several iterations helps to find ν_t (compared to $\hat{\theta}_{t-1}$) more accurately.

We use the following adaptive state transition model

$$\theta_t = \hat{\theta}_{t-1} + \nu_t + U_t, \tag{29}$$

where ν_t is the predicted shift in the motion vector. The choice of U_t is discussed below. One should note that we are not using (29) as a proposal function to draw particles, which requires using (9) to compute the particle weight. Instead we directly use it as the state transition model and hence use (8) to compute the particle weight. Our model can be easily interpreted as a time-varying state model.

It is interesting to note that the approach proposed in [26] also uses motion cues as well as color parameter adaptation. Our approach is different from [26] in that: (i) We use the motion cue in the state transition model while they use it as part of observations; (ii) We only use the gray images without using the color cue which is used in [26]; and (iii) We use an adaptive appearance models which is updated by the EM algorithm while they use an adaptive color model which is updated by a stochastic version of the EM algorithm.

B. Adaptive noise

The value of ϵ_t determines the quality of prediction. Therefore, if ϵ_t is small, which implies a good prediction, we only need noise with small variance to absorb the residual motion; if ϵ_t is large, which implies a poor prediction, we then need noise with large variance to cover potentially large jumps in the motion state.

To this end, we use U_t of the form $U_t = R_t * U_0$, where R_t is a function of ϵ_t. Since ϵ_t defined in (28) is a 'variance'-type measure, we use

$$R_t = \max(\min(R_0\sqrt{\epsilon_t}, R_{max}), R_{min}), \tag{30}$$

where R_{min} is the lower bound to maintain a reasonable sample coverage and R_{max} is the upper bound to constrain the computational load.

C. Adaptive number of particles

If the noise variance R_t is large, we need more particles, while conversely, fewer particles are needed for noise with small variance R_t. Based on the principle of asymptotic relative efficiency (ARE) [32], we should adjust the particle number J_t in a similar fashion, i.e.,

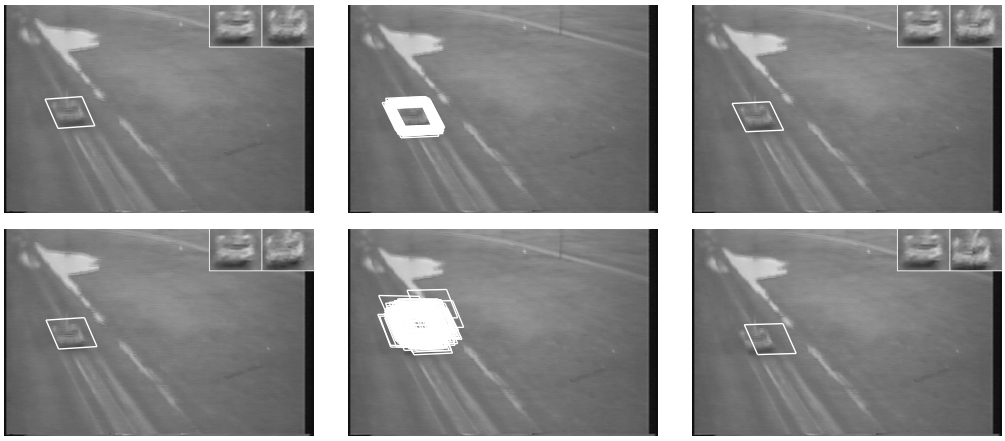
$$J_t = J_0 R_t / R_0. \tag{31}$$

Fox [33] also presents an approach to improve the efficiency of particle filters by adapting the particle numbers on-the-fly. His approach is to divide the state space into bins and approximate the posterior distribution by a multinomial distribution. A small number of particles is used if the density is focused on a small part of the state space and a large number of particles if the uncertainty in the state space is high. In this way, the error between the empirical distribution and the true distribution (approximated as a multinomial in his analysis) measured by Kullback-Leilber distance is bounded. However, in his approach, since the state space (only 2D) is exhaustively divided, the number of particles is at least several thousand, while our approach uses at most a few hundred. Our attempt is not to explore the state space (6-D affine space) exhaustively, but only the regions that have high potential for the object to be present.

D. Comparison between the adaptive velocity model and the zero velocity model

We demonstrate the necessity of the adaptive velocity model by comparing it with the zero velocity model. Fig. 2 shows the particle configurations created from the adaptive velocity model

(with $J_t < J_0$ and $R_t < R_0$ computed as above) and the zero velocity model (with $J_t = J_0$ and $R_t = R_0$). Clearly, the adaptive-velocity model generates particles very efficiently, i.e, they are tightly centered around the object of interest so that we can easily track the object at time t; while the zero-velocity model generates more particles widely spread to explore larger regions, leading to unsuccessful tracking as widespread particles often lead to a local minimum.



Tracking result at $t-1$ Particle configuration at t Tracking result at t

Fig. 2. Particle configurations from (top row) the adaptive velocity model and (bottom row) the zero-velocity model.

V. OCCLUSION HANDLING

Occlusion is usually handled in two ways. One way is to use joint probabilistic data associative filter (JPDAF) [34], [35]; and the other one is to use robust statistics [12]. We use robust statistics here.

A. Robust statistics

We assume that occlusion produces large image differences which can be treated as 'outliers'. Outlier pixels cannot be explained by the underlying process and their influences on the estimation process should be reduced. Robust statistics provide such mechanisms.

We use the $\hat{\rho}$ function defined as follows:

$$\hat{\rho}(x) = \begin{cases} \frac{1}{2}x^2 & \text{if} \quad |x| \leq c \\ cx - \frac{1}{2}c^2 & \text{if} \quad |x| > c \end{cases}, \tag{32}$$

where x is normalized to have unit variance and the constant c controls the outlier rate. In our experiment, we take $c = 1.435$ based on experimental experience. If $|x| > c$ is satisfied, we declare the corresponding pixel an outlier.

B. Robust likelihood measure and adaptive velocity estimate

The likelihood measure defined in Eq. (10) involves a multi-dimensional normal density. Since we assume that each pixel is independent, we consider the one-dimensional normal density. To make the likelihood measure robust, we replace the one-dimensional normal density $\mathsf{N}(x; \mu, \sigma^2)$ by

$$\hat{\mathsf{N}}(x; \mu, \sigma^2) = (2\pi\sigma^2)^{-1/2} \exp(-\hat{\rho}(\frac{x-\mu}{\sigma})). \tag{33}$$

Note that this is not a density function any more, but since we are dealing with discrete approximation in the particle filter, normalization makes it a probability mass function.

Existence of outlier pixels severely violates the constant brightness constraint and hence affects our estimate of the adaptive velocity. To downweight the influence of the outlier pixels in estimating the adaptive velocity, we introduce a $d \times d$ diagonal matrix L_t with its i^{th} diagonal element being $L_t(i) = \eta(x_i)$ where x_i is the pixel intensity of the difference image $(\mathcal{T}\{Y_t; \tilde{\theta}_t\} - \hat{Z}_{t-1})$ normalized by the variance of the OAM stable component and

$$\eta(x) = \frac{1}{x}\frac{d\hat{\rho}(x)}{dx} = \begin{cases} 1 & \text{if} \quad |x| \le c \\ c/|x| & \text{if} \quad |x| > c \end{cases}, \tag{34}$$

Eq. (21) becomes

$$\nu_t \simeq -B_t L_t (\mathcal{T}\{Y_t; \hat{\theta}_{t-1}\} - \hat{Z}_{t-1}). \tag{35}$$

This is similar in principle to the weighted least square algorithm.

C. Occlusion declaration

If the number of the outlier pixels in $\hat{Z}_t$ (compared with the OAM), say d_{out}, exceeds a certain threshold, i.e., $d_{out} > \lambda d$ where $0 < \lambda < 1$ (we take $\lambda = 0.15$), we declare occlusion. Since the OAM has more than one component, we count the number of outlier pixels with respect to every component and take the maximum.

If occlusion is declared, we stop updating the appearance model and estimating the motion velocity. Instead, we (i) keep the current appearance model, i.e., $A_{t+1} = A_t$ and (ii) set the

Initialize *a sample set* $\mathcal{S}_0 = \{\theta_0^{(j)}, 1/J_0)\}_{j=1}^{J_0}$ *according to prior distribution* $p(\theta_0)$.

Initialize *the appearance model* A_1.

Set $OCC_{FLAG} = 0$ *to indicate no occlusion.*

For $t = 1, 2, \ldots$

If *(*$OCC_{FLAG} == 0$*)*

Calculate *the state estimate* $\hat{\theta}_{t-1}$ *by Eq. (3) or (4), the adaptive velocity* ν_t *by Eq. (21), the noise variance* R_t *by Eq. (30), and the particle number* J_t *by Eq. (31).*

Else

$R_t = R_{max}$, $J_t = J_{max}$, $\nu_t = 0$.

End

For $j = 1, 2, \ldots, J_t$

Draw *the sample* $U_t^{(j)}$ *for* U_t *with variance* R_t.

Construct *the sample* $\theta_t^{(j)} = \hat{\theta}_{t-1} + \nu_t + U_t^{(j)}$ *by Eq. (29).*

Compute *the transformed image* $Z_t^{(j)}$.

Update *the weight using* $w_t^{(j)} = p(Y_t|\theta_t^{(j)}) = p(Z_t^{(j)}|\theta_t^{(j)})$.

End

Normalize *the weight using* $w_t^{(j)} = w_t^{(j)} / \sum_{j=1}^{J} w_t^{(j)}$.

Set OCC_{FLAG} *according to the number of the outlier pixels in* $\hat{Z}_t$.

If *(*$OCC_{FLAG} == 0$*)*

Update *the appearance model* A_{t+1} *using* $\hat{Z}_t$.

End

End

Fig. 3. The proposed visual tracking algorithm with occlusion handling.

motion velocity to zero, i.e., $\nu_t = 0$ and use the maximum number of particles sampled from the diffusion process with largest variance, i.e., $R_t = R_{max}$, and $J_t = J_{max}$.

The adaptive particle filtering algorithm with occlusion analysis is summarized in Fig. 3.

VI. Experimental results on visual tracking

In our implementation, we used the following choices. We consider affine transformations only. Specifically, the motion is characterized by $\theta = (a_1, a_2, a_3, a_4, t_x, t_y)$ where $\{a_1, a_2, a_3, a_4\}$ are deformation parameters and $\{t_x, t_y\}$ denote the 2-D translation parameters. Even though significant pose/illumincation changes are present in the video, we believe that our adaptive appearance model can easily absorb them and therefore for our purposes the affine transformation is a reasonable approximation. Regarding photometric transformations, only a zero-mean-unit-

variance normalization is used to partially compensate for contrast variations. The complete image transformation $\mathcal{T}\{Y;\theta\}$ is implemented as follows: affine transform Y using $\{a_1, a_2, a_3, a_4\}$, crop out the region of interest at position $\{t_x, t_y\}$ with the same size as the still template in the appearance model, and perform zero-mean-unit-variance normalization.

We demonstrate our algorithm by tracking a disappearing car, a moving tank from micro air vehicle, and a moving face under occlusion. Table I summarizes some statistics about the video sequences and the appearance model size used.

Video	Car	Tank	Face
# of frames	500	300	800
Frame size	576x768	240x360	240x360
A_t size	24x30	24x30	30x26
Occlusion	No	No	Yes (twice)
'adp'	o	o	x
'fa'	o	o	x
'fm'	x	x	x
'fb'	x	x	x
'adp & occ'	o	o	o

TABLE I

COMPARISON OF TRACKING RESULTS OBTAINED BY PARTICLE FILTERS WITH DIFFERENT CONFIGURATIONS. 'A_t SIZE' MEANS PIXEL SIZE IN THE COMPONENT(S) OF THE APPEARANCE MODEL. 'O' MEANS SUCCESS IN TRACKING. 'X' MEANS FAILURE IN TRACKING.

We initialize the particle filter and the appearance model with a detector algorithm (we actually used the face detector described in [36] for the face sequence) or a manually specified image patch in the first frame. R_0 and J_0 are also manually set, depending on the sequence.

A. Car tracking

We first test our algorithm to track a vehicle with the F-component but without occlusion analysis. The result of tracking a fast moving car is shown in Fig. 4 (column 1)[4]. The tracking result is shown with a bounding box. We also show the stable and wandering components

[4]Accompanying videos are available at http://www.cfar.umd.edu/∼shaohua/research/.

Fig. 4. The car sequence. Notice the fast scale change present in the video. Column 1: the tracking results obtained with an adaptive motion model and an adaptive appearance model ('adp'). Column 2: the tracking results obtained with an adaptive motion model but a fixed appearance model ('fa'). In this case, the corner shows the tracked region. Column 3: the tracking results obtained with an adaptive appearance model but a fixed motion model ('fm').

separately (in a double-zoomed size) at the corner of each frame. The video is captured by a camera mounted on the car. In this footage the relative velocity of the car with respect to the camera platform is very large, and the target rapidly decreases in size. Our algorithm's adaptive particle filter successfully tracks this rapid change in scale. Fig. 5(a) plots the scale estimate (calculated as $\sqrt{(a_1^2 + a_2^2 + a_3^2 + a_4^2)/2}$) recovered by our algorithm. It is clear that the scale follows a decreasing trend as time proceeds. The pixels located on the car in the final frame are about 12 by 15 in size, which makes the vehicle almost invisible. In this sequence we set $J_0 = 50$ and $R_0 = 0.25$. The algorithm implemented in a standard Matlab environment processes

about 1.2 frames per second (with $J_0 = 50$) running on a PC with a PIII 650 CPU and 512M memory.

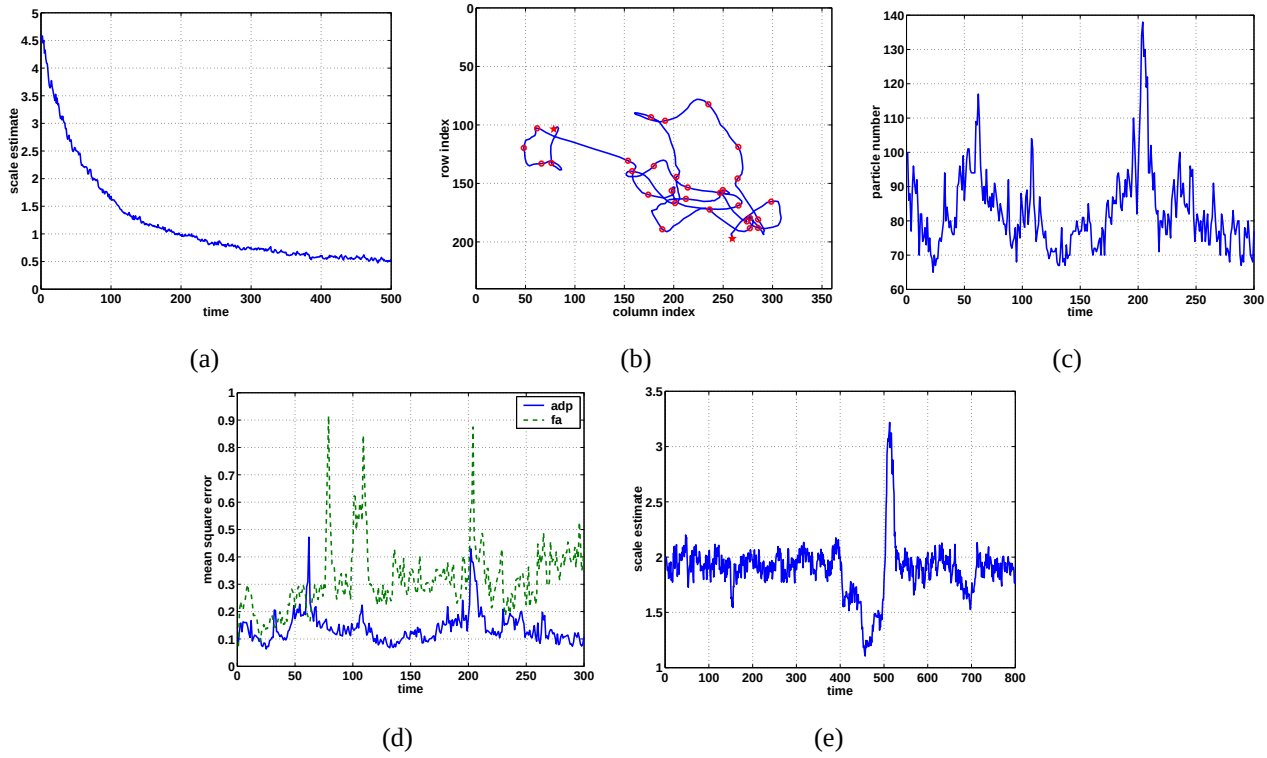


Fig. 5. (a) The scale estimate for the car. (b) The 2-D trajectory of the centroid of the tracked tank. '*' means the starting and ending points and '.' points are marked along the trajectory every 10 frames. (c) The particle number J_t *vs.* t obtained when tracking the tank. (d) The MSE invoked by the 'adp' and 'fa' algorithms. (e) The scale estimate for the face sequence.

B. Tank tracking in an aerial video

Fig. 6 shows our results on tracking a tank in an aerial video with degraded image quality due to motion blur. Also, the movement of the tank is very jerky and arbitrary because of platform motion, as evidenced in Fig. 5(b) which plots the 2-D trajectory of the centroid of the tracked tank every 10 frames, covering from the left to the right in 300 frames. Although the tank moved about 100 pixels in column index in a certain period of 10 frames, the tracking is still successful.

Fig. 5(c) displays the plot of actual number of particles J_t as a function of time t. The average number of particle is about 83, where we set J_0 to be 100, which means that in this case we actually saved about 20% in computation by using an adaptive J_t instead of a fixed number of particles.

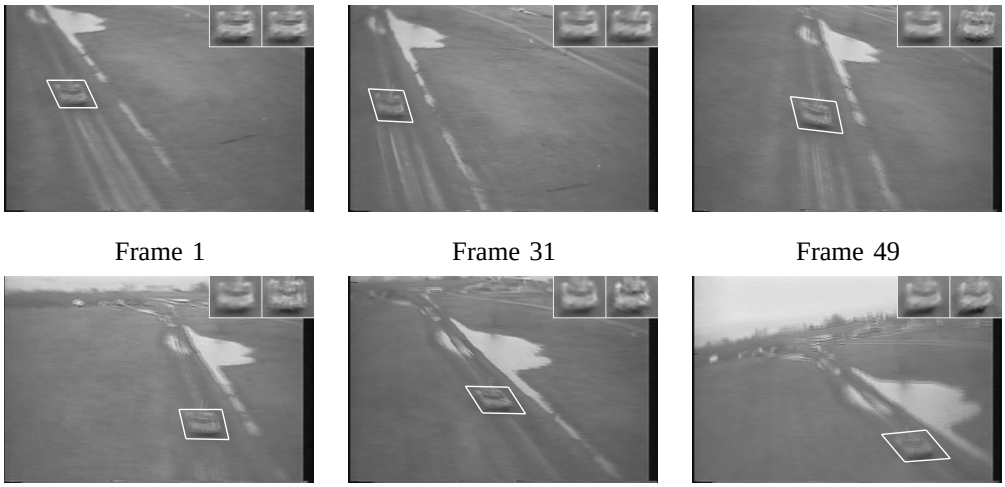


Fig. 6. Tracking a moving tank in a video acquired by an airborne camera.

To further illustrate the importance of the adaptive appearance model, we computed the mean square error (MSE) invoked by two particle filter algorithms, one (referred as 'adp' in Section VI-D) using the adaptive appearance model and the other (referred as 'fa' in Section VI-D) using a fixed appearance model. Computing the MSE for the 'fa' algorithm is straightforward, with T_0 denoting the fixed template,

$$MSE_{fa}(t) = d^{-1}\sum_{j=1}^{d}(\hat{Z}_t(j) - T_0(j))^2. \tag{36}$$

Computing the MSE for the 'adp' algorithm is as follows:

$$MSE_{adp}(t) = d^{-1}\sum_{j=1}^{d}\{\sum_{i=w,s,f} m_{i,t}(\hat{Z}_t(j) - \mu_{i,t}(j))^2\}. \tag{37}$$

Fig. 5(d) plots the functions of $MSE_{fa}(t)$ and $MSE_{adp}(t)$. Clearly, using the adaptive appearance model invokes smaller MSE for almost all 300 frames. The average MSE for the 'adp' algorithm is 0.1394 [5] while that for the 'fa' algorithm is 0.3169!

C. Face tracking

We present one example of successful tracking of a human face using a hand-held video camera in an office environment, where both camera and object motion are present.

[5]The range of MSE is very reasonable since we are using image patches after the zero-mean-unit-variance normalization not the raw image intensities.

Fig. 7 presents the tracking results on the video sequence featuring the following variations: moderate lighting variations, quick scale changes (back and forth) in the middle of the sequence, and occlusion (twice). The results are obtained by incorporating the occlusion analysis in the particle filter, but we did not use the F-component. Notice that the adaptive appearance model remains fixed during occlusion.

Fig. 8 presents the tracking results obtained using the particle filter without occlusion analysis. We have found that the predicted velocity actually accounts for the motion of the occluding hand since the outlier pixels (mainly on the hand) dominate the image difference $(\mathcal{T}\{Y_t; \tilde{\theta}_t\} - \hat{Z}_{t-1})$. Updating the appearance model deteriorates the situation.

Fig. 5(e) plots the scale estimate against time t. We clearly observe a rapid scale change (a sudden increase followed by a decrease within about 50 frames) in the middle of the sequence (though hard to display the recovered scale estimates are in perfect synchrony with the video data).

D. Comparison

We illustrate the effectiveness of our adaptive approach ('adp') by comparing the particle filter either with (a) an adaptive motion model but a fixed appearance model ('fa'), or with (b) a fixed motion model but an adaptive appearance model ('fm'); or with (c) a fixed motion model and a fixed appearance model ('fb'). Table I lists the tracking results obtained using particle filters under the above situations, where 'adp & occ' means the adaptive approach with occlusion handling. Fig. 4 also shows the tracking results on the car sequence when the 'fa' and 'fm' options are used.

Table I seems to suggest that the adaptive motion model plays a more important role than the adaptive appearance model since 'fa' always yields successful tracking while 'fm' fails, the reasons being that (i) the fixed motion model is unable to adapt to quick motion present in the video sequences, and (ii) the appearance changes in the video sequences, though significant in some cases, are still within the range of the fixed appearance model. However, as seen in the videos, 'adp' produces much smoother tracking results than 'fa', demonstrating the power of the adaptive appearance model.

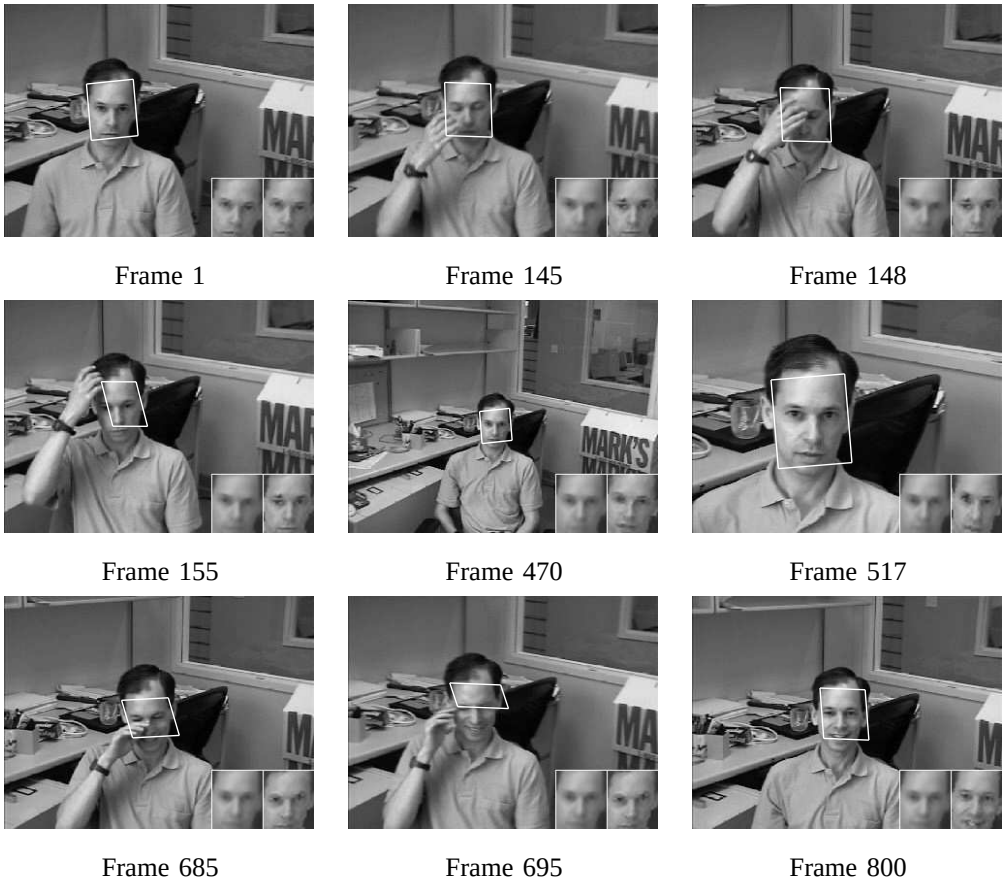
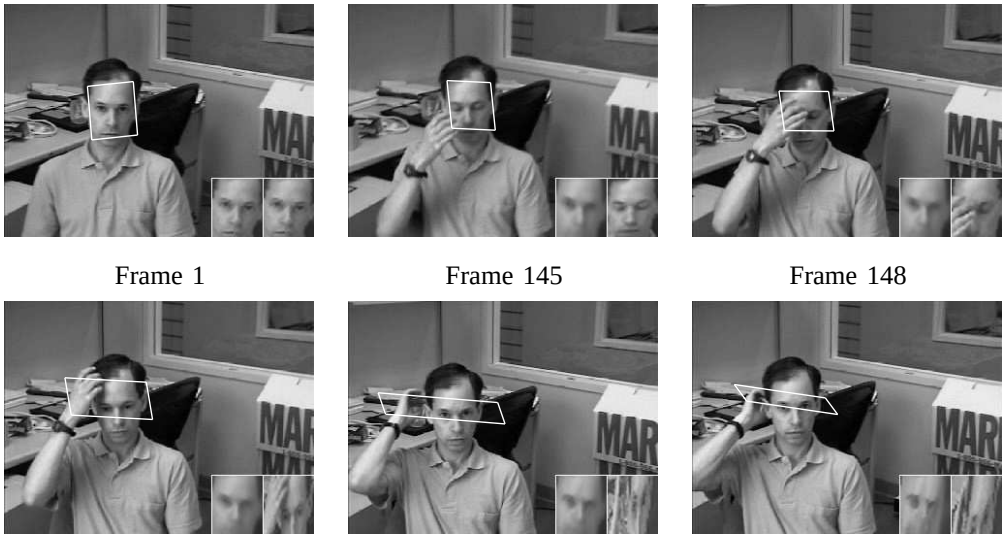


Fig. 7. The face sequence. Frames 145, 148, and 155 show the first occlusion. Frames 470 and 517 show the smallest and largest face observed. Frames 685, 690, and 710 show the second occlusion.

VII. Simultaneous tracking and recognition

Visual tracking models the inter-frame appearance differences and visual recognition models the appearance difference between video frames and gallery images. Simultaneous tracking and recognition [7] is shown to be an effective approach for handling tracking and recognition. It models appearance differences in both tracking and recognition in one framework, which actually improves both tracking and recognition accuracies over the approaches separating tracking and recognition as two tasks. The proposed framework in [7] is rather general and accommodates various model choices. The more effective the model choices are, improved performance in tracking and recognition is expected. Another important feature of [7] is the accumulation of recognition evidence in a probabilistic, recursive, and interpretable manner. In this paper, we attempt to demonstrate the effectiveness of the proposed model choices using experiments on a



Frame 1 Frame 145 Frame 148

Frame 155 Frame 170 Frame 200

Fig. 8. Tracking results on the face sequence using the adaptive particle filter without occlusion analysis.

challenging dataset that has significant pose and illumination variations.

We assume that there is a gallery set $\{I_1, ..., I_N\}$ with each individual n possessing one facial image I_n in frontal view. Here n is treated as a random variable taking value in the sample space $\mathcal{N} = \{1, 2, ..., N\}$. The essence of our framework is posterior probability computation, i.e., computing $p(n_t, \theta_t|Y_{1:t})$, whose marginal posterior probability $p(n_t|Y_{1:t})$ solves the recognition task and whose marginal posterior probability $p(\theta_t|Y_{1:t})$ solves the tracking task.

After a brief review of the time series model for recognition in Sec. VII-A, we describe in Sec. VII-B the three components yielding improvements. Experimental results and discussions are then presented in Sec. VII-C.

A. Review of recognition model

We briefly present the propagation model for recognition, consisting of the following three components, namely the motion transition equation, the identity equation, and the observation likelihood and define the recognition task as a statistical inference problem, which can be solved using particle filters.

Motion transition equation: We use the same adaptive-velocity motion model as described in Section IV.

Identity equation: Denoting the identity variable by $n_t \in \mathcal{N} = \{1, 2, ..., N\}$, indexing the gallery set $\{I_1, ..., I_N\}$, and assuming that the identity does not change as time proceeds, we have

$$n_t = n_{t-1}, \quad t \geq 1. \tag{38}$$

In practice, one may assume a small transition probability between identity variables to increase the robustness.

Observation likelihood: In [7], our empirical results show that combining contributions (or scores) from both tracking and recognition in the likelihood yields the best performance in both tracking and recognition.

To compute the tracking score $p_a(Y_t|\theta_t)$ which measures the inter-frame appearance changes, we use the appearance model introduced in Section III and the quantity defined in (10) as $p_a(Y_t|\theta_t)$.

To compute the recognition score which measures the appearance changes between probe videos and gallery images, we assume that the transformed observation is a noise-corrupted version of some still template in the gallery, i.e.,

$$Z_t = I_{n_t} + X_t, \quad t \geq 1, \tag{39}$$

where X_t is the *observation noise* at time t, whose distribution determines the recognition score $p_n(Y_t|n_t, \theta_t)$. We will physically define this quantity in Sec. VII-B.

To fully exploit the fact that all gallery images are in frontal view, we also compute in Sec. VII-B how likely the patch Z_t is in frontal view and denote this score by $p_f(Y_t|\theta_t)$. If the patch is in frontal view, we accept a recognition score; otherwise, we simply set the recognition score as equiprobable among all identities, i.e., $1/N$. The complete likelihood $p(Y_t|n_t, \theta_t)$ is now defined as

$$p(Y_t|n_t, \theta_t) \propto p_a \ \{p_f \ p_n + (1 - p_f) \ N^{-1}\}. \tag{40}$$

Particle filter for solving the model: We assume statistical independence between all noise variables and prior knowledge on the distributions $p(\theta_0)$ and $p(n_0)$ (uniform prior in fact). Given this model, our goal is to compute the posterior probability $p(n_t|Y_{1:t})$. It is in fact a probability mass function (PMF) since n_t only takes values from $\mathcal{N} = \{1, 2, ..., N\}$, as well as a marginal probability of $p(n_t, \theta_t|Y_{1:t})$, which is a mixed-type distribution. Therefore, the problem is reduced to computing the posterior probability.

Since the model is nonlinear and non-Gaussian in nature, there is no analytic solution. We invoke a particle filter to provide numerical approximations to the posterior distribution $p(n_t, \theta_t|Y_{1:t})$. Also, for this mixed-type distribution, we can greatly improve the computational load by judiciously utilizing the discrete nature of the identity variable as in [7]. We [7] also theoretically justified the evolving behavior of the recognition density $p(n_t|Y_{1:t})$ under a weak assumption.

Initialize *a sample set* $\mathcal{S}_0 = \{\theta_0^{(j)}, w_0^{(j)} = 1/J_0)\}_{j=1}^{J_0}$ *according to prior distribution* $p(\theta_0)$. *Set* $\beta_{0,l} = 1/N$. *Initialize appearance mode* A_1.

For $t = 1, 2, \ldots$

Calculate *the MAP estimate* $\hat{\theta}_{t-1}$, *the adaptive motion shift* ν_t *by Eq. (21), the noise variance* r_t *by Eq. (30), and particle number* J_t *by Eq. (44).*

For $j = 1, 2, \ldots, J_t$

Draw *the sample* $U_t^{(j)}$ *for* U_t *with variance* R_t.

Construct *the sample* $\theta_t^{(j)}$ *by Eq. (29).*

Compute *the transformed image* $Z_t^{(j)}$.

For $l = 1, 2, ..., N$

Update *the weight using* $\alpha_{t,l}^{(j)} = \beta_{t,l} p(Y_t|l, \theta_t^{(j)}) = \beta_{t,l} p(Z_t^{(j)}|l, \theta_t^{(j)})$ *by Eq. (40).*

End

End

Normalize *the weight using* $w_{t,l}^{(j)} = \alpha_{t,l}^{(j)} / \sum_{j,l} \alpha_{t,l}^{(j)}$ *and compute* $w_t^{(j)} = \sum_j w_{t,l}^{(j)}$ *and* $\beta_{t,l} = \sum_j w_{t,l}^{(j)}$.

Update *the appearance model* A_{t+1} *using* $\hat{Z}_t$.

End

Fig. 9. The visual tracking and recognition algorithm.

B. Model components in detail

As mentioned earlier, the proposed algorithm incorporates three components which improve our previous approach [7]. We will now examine each of these components in greater detail. The proposed algorithm is then summarized.

Modeling inter-frame appearance changes: Inter-frame appearance changes are related to the motion transition model and the appearance model for tracking, which were explained in Sections III and IV.

Being in frontal view: Since all gallery images are in frontal view, we simply measure the extent of being frontal by fitting a probabilistic subspace (PS) density on the top of the gallery images [37], [15], assuming that they are i.i.d. samples from the frontal face space (FFS). The method works as follows: a regular PCA is first performed (zero mean is assumed by removing the sample mean). Suppose the eigensystem for the FFS is $\{(\lambda_i, e_i)\}_{i=1}^{d}$, where d is the number of pixels and $\lambda_1 \geq ... \geq \lambda_d$. Only top s principal components corresponding to top s eigenvalues are then kept while the residual components are considered as isotropic. We refer the reader to the original paper [37] for full details. The PS density is written as follows:

$$Q(x) = \{\frac{exp(-\frac{1}{2}\sum_{i=1}^{s}\frac{q_i^2}{\lambda_i})}{(2\pi)^{s/2}\prod_{i=1}^{s}\lambda_i^{1/2}}\}\{\frac{exp(-\frac{err^2}{2\rho})}{(2\pi\xi)^{(d-s)/2}}\}, \tag{41}$$

where $q_i = e_i^T x$ for $i = 1, ..., s$ is the i^{th} principal component of x, $err^2 = \|x\|^2 - \sum_{i=1}^{s} q_i^2$ is the reconstruction error, and $\xi = (\sum_{i=s+1}^{d} \lambda_i)/(d-q)$. It is easy to write $p_f(Y_t|\theta_t)$ as follows:

$$p_f(Y_t|\theta_t) = Q_{FFS}(Z_t). \tag{42}$$

Modeling appearance changes between probe video frames and gallery images: We adopt the MAP rule developed in [15] for the recognition score $p_n(Y_t|n_t, \theta_t)$. Two subspaces are constructed to model appearance variations. The intra-personal space (IPS) is meant to cover all the variations in appearances belonging to the same person while the extra-personal space (EPS) is used to cover all the variations in appearances belonging to different people. More than one facial image per person is needed to construct the IPS. Apart from the available gallery, we crop out four images from the video ensuring no overlap with frames used in probe videos. The above PS density estimation method is applied separately to the IPS and the EPS, yielding two different eigensystems. The recognition score $p_n(Y_t|n_t, \theta_t)$ is finally computed as, assuming equal priors on the IPS and the EPS,

$$p_n(Y_t|n_t, \theta_t) = \frac{Q_{IPS}(Z_t - I_{n_t})}{Q_{IPS}(Z_t - I_{n_t}) + Q_{EPS}(Z_t - I_{n_t})}. \tag{43}$$

Proposed algorithm: We adjust the particle number J_t based on the following considerations. (i) The first issue is same as (31) based on prediction error. (ii) As proved in [7], the uncertainty in the identity variable n_t is characterized by an entropy measure H_t for $p(n_t|Y_{1:t})$ and H_t is a non-increasing function (under one weak assumption). Accordingly, we increase the number of

particles by a fixed amount J_{fix} if H_t increases; otherwise we deduct J_{fix} from J_t. Combining these two, we have

$$J_t = J_0 \frac{R_t}{R_0} + J_{fix} * (-1)^{i[H_{t-1} < H_{t-2}]}\}, \tag{44}$$

where $i[.]$ is an indication function.

The proposed particle filtering algorithm for simultaneous tracking and recognition is summarized in Fig. 9, where $w_{t,l}^{(j)}$ is the weight of the particle $(n_t = l, \theta_t = \theta_t^{(j)})$ for the posterior density $p(n_t, \theta_t | Y_{1:t})$; $w_t^{(j)}$ is the weight of the particle $\theta_t = \theta_t^{(j)}$ for the posterior density $p(\theta_t | Y_{1:t})$; and $\beta_{t,l}$ is the weight of the particle $n_t = l$ for the posterior density $p(n_t | Y_{1:t})$. Occlusion analysis can also be included in Fig. 9.

C. Experimental results on visual tracking and recognition

We have applied our algorithm for tracking and recognizing human faces captured by a hand-held video camera in office environments . There are 29 subjects in the database. Fig. 10 lists all the images in the galley set and the top 10 eigenvectors for the FFS, IPS, and EPS, respectively. Fig. 11 presents some frames (with tracking results) in the video sequence for 'Subject-2' featuring quite large pose variations, moderate illumination variations, and quick scale changes (back and forth toward the end of the sequence).

Tracking is successful for all video sequences and 100% recognition rate is achieved, while our previous approach [7] failed to track in several video sequences due to its inability to handle significant appearance changes caused by pose and illumination variations. The posterior probabilities $p(n_t | Y_{1:t})$ with $n_t = 1, 2, ...N$ obtained for the 'Subject-2' sequence are plotted in Fig. 12(a). We start from uniform prior for the identity variable, i.e., $p(n_0) = N^{-1}$ for $n_0 = 1, 2, ...N$. It is very fast, taking about less than 10 frames, to reach above 0.9 level for the posterior probability corresponding to 'Subject-2', while all other posterior probabilities corresponding to other identities approach zero. This is mainly attributed to the discriminative power of the MAP recognition score induced by IPS and EPS modeling. The previous approach [7] usually takes about 30 frames to reach 0.9 level since only intra-personal modeling is adopted. Fig. 12(b) captures the scale change in the 'Subject-2' sequence.

Fig. 10. Row 1-3: the gallery set with 29 subjects in frontal view. Rows 4, 5, and 6: the top 10 eigenvectors for the FFS, IPS, and EPS, respectively.

VIII. CONCLUSIONS

We have presented an adaptive method for visual tracking which stabilizes the tracker by embedding deterministic linear prediction into stochastic diffusion. Numerical solutions have been provided using particle filters with the adaptive observation model arising from the adaptive appearance model, adaptive state transition model, and adaptive number of particles. Occlusion analysis is also embedded in the particle filter. Our algorithm was tested on several tasks consisting of tracking visual objects such as car, tank and human faces in realistic scenarios.

We have improved our simultaneous tracking and recognition approach previously proposed in [7]. More complex models, namely adaptive appearance model, adaptive-velocity transition model, and intra- and extra-personal space models, are introduced to handle appearance changes between frames and between frames and gallery images. The fact that the gallery images are in frontal view is enforced too. Experimental results demonstrate that the tracker is stable and the recognition performance is good.

Fig. 11. Example images in 'Subject-2' probe video sequence and the tracking results.

ACKNOWLEDGEMENT

Supported in part by the Advanced Sensors Consortium sponsored by U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0008 and the Mitsubishi Electric Research Laboratories (MERL). We thank Mike Jones, MERL, for providing the face detector algorithm[36]. We also acknowledge three anonymous reviewers for their critical suggestions for improving the quality of presentation.

REFERENCES

[1] A. Doucet, N. d. Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York, 2001.

[2] M. Isard and A. Blake, "Contour tracking by stochatic propagation of conditional density," *European Conference on Computer Vision*, 1996.

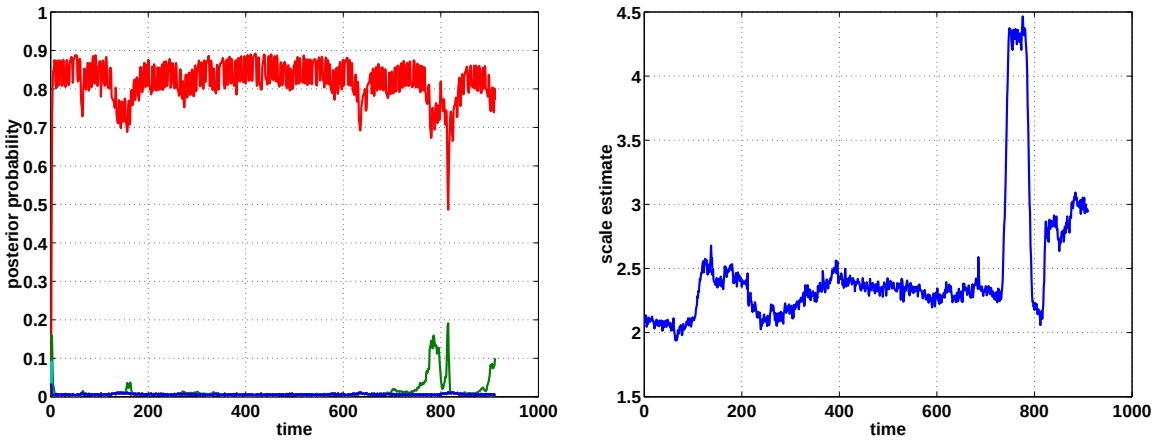
[3] B. North, A. Blake, M. Isard, and J. Rittscher, "Learning and classification of complex dynamics," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1016–1034, 2000.

[4] M. J. Black and A. D. Jepson, "A probabilistic frameworkfor matching temporal trajectories," *IEEE International Conference on Computer Vision*, pp. 176–181, 1999.

[5] M. J. Black and D. J. Fleet, "Probabilistic detection and tracking of motion discontinuities," *IEEE International Conference on Computer Vision*, vol. 2, pp. 551–558, 1999.

[6] Y. Wu and T. S. Huang, "A co-inference approach to robust visual tracking," *IEEE International Conference on Computer Vision*, vol. 2, pp. 26–33, 2001.

[7] S. Zhou, V. Krueger, and R. Chellappa, "Probabilistic recognition of human faces from video," *Computer Vision and Image Understanding*, vol. 91, pp. 214–245, 2003.



(a) (b)

Fig. 12. Results on the 'Subject-2' sequence. (a) Posterior probabilities against time t for all identities $p(n_t|Y_{1:t})$, $n_t = 1, 2, ..., N$. The line close to 1 is for the true identity. (b) Scale estimate against time t.

[8] G. D. Hager and P. N. Belhumeur, "Efficient region tracking with parametric models of geometry and illumination," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 20, no. 10, pp. 1025–1039, 1998.

[9] H. Sidenbladh, M. J. Black, and D. J. Fleet, "Stochastic tracking of 3d human figures using 2d image motion," *European Conference on Computer Vision*, vol. 2, pp. 702–718, 2002.

[10] A. D. Jepson, D. J. Fleet, and T. El-Maraghi, "Robust online appearance model for visual tracking," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 415–422, 2001.

[11] B. Li and R. Chellappa, "Face verification through tracking facial features," *Journal of Optical Society of America A*, vol. 18, no. 12, pp. 2969–2981, 2001.

[12] P. J. Huber, *Robust statistics*. Wiley, 1981.

[13] M. J. Black and A. D. Jepson, "Eigentracking: Robust matching and tracking of articulated objects using a view-based representation," *European Conference on Computer Vision*, vol. 1, pp. 329–342, 1996.

[14] T. Jebara and A. Pentland, "Parameterized structure from motion for 3D adaptive feedback tracking of faces," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 144 –150, 1997.

[15] B. Moghaddam, "Principal manifolds and probabilistic subspaces for visual recognition," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 780–788, 2002.

[16] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 142–149, 2000.

[17] T. J. Broida, S. Chandra, and R. Chellappa, "Recursive techniques for estimation of 3-d translation and rotation parameters from noisy image sequences," *IEEE Transaction on Aerospace and Electronic Systems*, vol. AES-26, pp. 639–656, 1990.

[18] A. Azarbayejani and A. Pentland, "Recursive estimation of motion, structure, and focal length," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 562–575, 1995.

[19] B. Anderson and J. Moore, *Optimal Filtering*. New Jersey: Prentice Hall, Engle-wood Cliffs, 1979.

[20] J. S. Liu and R. Chen, "Sequential monte carlo for dynamic systems," *Journal of the American Statistical Association*, vol. 93, pp. 1031–1041, 1998.

[21] G. Kitagawa, "Monte carlo filter and smoother for non-gaussian nonlinear state space models," *Journal of Computational and Graphical Statistics*, vol. 5, pp. 1–25, 1996.

[22] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," *IEE Proceedings on Radar and Signal Processing*, vol. 140, pp. 107–113, 1993.

[23] K. Toyama and A. Blake, "Probabilistic tracking in a metric space," *IEEE International Conference on Computer Vision*, pp. 50–59, 2001.

[24] A. Doucet, S. J. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–209, 2000.

[25] M. Isard and A. Blake, "ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework," *European Conference on Computer Vision*, vol. 1, pp. 767–781, 1998.

[26] J. Vermaak, P. Perez, M. Gangnet, and A. Blake, "Towards improved observation models for visual tracking: Selective adaptation," *European Conference on Computer Vision*, pp. 645–660, 2002.

[27] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm." *J. Roy. Statist. Soc. B*, 1977.

[28] J. Sullivan and J. Rittscher, "Guiding random particle by deterministic search," *International Conference on Computer Vision*, vol. 1, pp. 323 –330, 2001.

[29] F. Jurie and M. Dhome, "A simple and efficient template matching algorithm," *International Conference on Computer Vision*, vol. 2, pp. 544–549, 2001.

[30] A. Bergen, P. Anadan, K. Hanna, and R. Hingorani, "Hierarchical model-based motion estimation," *European Conference on Computer Vision*, pp. 237–252, 1992.

[31] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *International Joint Conference on Artifical Intelligence*, 1981.

[32] G. Casella and R. L. Berger, *Statistical Inference*. Duxbury, 2002.

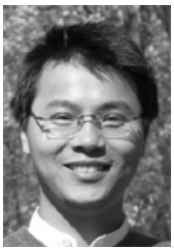
[33] D. Fox, "Kld-sampling: Adaptive particle filters and mobile robot localization," *Neural Information Processing Systems (NIPS)*, 2001.

[34] Y. Bar-Shalom and T. Fortmann, *Tracking and Data Association*. Academic Press, 1988.

[35] C. Rasmussen and G. Hager, "Probabilistic data association methods for tracking complex visual objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 560–576, 2001.

[36] P. Voila and M. Jones, "Robust real-time object detection," *Second International Workshop on Statistical and Computational Theories of Vision*, 2001.

[37] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. PAMI-19, no. 7, pp. 696–710, 1997.



Shaohua Kevin Zhou (S'01) received his B.E. degree from the University of Science and Technology of China, Hefei, China, in 1994 and M.E. degree from the National University of Singapore in 2000. He is a Ph.D. candidate in Electrical Engineering at the University of Maryland at College Park, and a graduate research assistant with the Center for Automation Research. He has general research interests in signal/image/video processing, computer vision, pattern recognition, machine learning, and statistical inference and computing. He has published papers on face recognition, motion analysis, illumination modelling, and kernel machine learning.

Rama Chellappa (S'78–M'79–SM'83–F'92) received the B.E. (Hons.) degree from the University of Madras, Madras, India, in 1975 and the M.E.(Distinction) degree from the Indian Institute of Science, Bangalore, in 1977. He received the M.S.E.E. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, in 1978 and 1981, respectively. Since 1991, he has been a Professor of electrical engineering and an Affiliate Professor of computer science with the University of Maryland, College Park. He is the Director of the Center for Automation Research and a Permanent Member of the Institute for Advanced Computer Studies. Prior to joining the University of Maryland, he was an Associate Professor and Director of the Signal and Image Processing Institute with the University of Southern California, Los Angeles. During the last 22 years, he has published numerous book chapters and peer-reviewed journal and conference papers. Several of his journal papers have been reproduced in collected works published by IEEE Press, IEEE Computer Society Press, and MIT Press. He has edited a collection of papers on *Digital Image Processing* (Santa Clara, CA: IEEE Computer Society Press), co-authored a research monograph on *Artificial Neural Networks for Computer Vision* (with Y. T. Zhou) (Berlin, Germany: Springer-Verlag), and co-edited a book on *Markov Random Fields* (with A. K. Jain) (New York Academic). His current research interests are image compression, automatic target recognition from stationary and moving platforms, surveillance and monitoring, biometrics, human activity modeling, hyper spectral image understanding, and commercial applications of image processing and understanding.

Dr. Chellappa has served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON IMAGE PROCESSING, and IEEE TRANSACTIONS ON NEURAL NETWORKS. He also served as Co-Editor-in-Chief of *Graphical Models and Image Processing*; and a member of the IEEE Signal Processing Society Board of Governors from 1996 to 1999. He is currently serving as the Editor-in-Chief of IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and as the vice-president of the IEEE Signal Processing Society for Awards and Membership. He has received several awards, including the 1985 NSF Presidential Young Investigator Award, the 1985 IBM Faculty Development Award, the 1991 Excellence in Teaching Award from the School of Engineering, University of Southern California, the 1992 Best Industry Related Paper Award from the International Association of Pattern Recognition (with Q. Zheng), and the IEEE Signal Processing Society Technical Achievement Award in 2001. He was elected as a Distinguished Faculty Research Fellow (1996–1998) and recently elected as a distinguished Scholar-Teacher for 2003 at the University of Maryland. He is a Fellow of the International Association for Pattern Recognition. He has served as a General and Technical Program Chair for several IEEE international and national conferences and workshops.

Baback Moghaddam is a Senior Research Scientist at Mitsubishi Electric Research Laboratory (MERL). He received his Ph.D. in Electrical Engineering & Computer Science from the Massachusetts Institute of Technology (MIT) in 1997. During his doctoral studies at MIT he was a Research Assistant in the Vision & Modeling group at the MIT Media Laboratory, where he developed an automatic face recognition system which was the top competitor in DARPA's "FERET" face recognition competition. Since joining MERL, Dr. Moghaddam has worked on visual sensing for surveillance, face recognition and fingerprint analysis for biometrics, image retrieval and visualization for image libraries, factorized density models of local image structure and most recently on 3D face modeling and recognition. His research interests include computer vision, image processing, computational learning theory and statistical pattern recognition. Dr. Moghaddam is on the editorial board of the journal Pattern Recognition and is a member of IEEE and ACM.

LIST OF FIGURE/TABLE CAPTIONS

Figure 1 The general particle filter algorithm.

Figure 2 Particle configurations from (top row) the adaptive velocity model and (bottom row) the zero-velocity model.

Figure 3 The proposed visual tracking algorithm with occlusion handling.

Figure 4 The car sequence. Notice the fast scale change present in the video. Column 1: the tracking results obtained with an adaptive motion model and an adaptive appearance model ('adp'). Column 2: the tracking results obtained with an adaptive motion model but a fixed appearance model ('fa'). In this case, the corner shows the tracked region. Column 3: the tracking results obtained with an adaptive appearance model but a fixed motion model ('fm').

Figure 5 (a) The scale estimate for the car. (b) The 2-D trajectory of the centroid of the tracked tank. '*' means the starting and ending points and '.' points are marked along the trajectory every 10 frames. (c) The particle number J_t *vs.* t obtained when tracking the tank. (d) The MSE invoked by the 'adp' and 'fa' algorithms. (e) The scale estimate for the face sequence.

Figure 6 Tracking a moving tank in a video acquired by an airborne camera.

Figure 7 The face sequence. Frames 145, 148, and 155 show the first occlusion. Frames 470 and 517 show the smallest and largest face observed. Frames 685, 690, and 710 show the second occlusion.

Figure 8 Tracking results on the face sequence using the adaptive particle filter without occlusion analysis.

Figure 9 The visual tracking and recognition algorithm.

Figure 10 Row 1-3: the gallery set with 29 subjects in frontal view. Rows 4, 5, and 6: the top 10 eigenvectors for the FFS, IPS, and EPS, respectively.

Figure 11 Example images in 'Subject-2' probe video sequence and the tracking results.

Figure 12 Results on the 'Subject-2' sequence. (a) Posterior probabilities against time t for all identities $p(n_t|Y_{1:t})$, $n_t = 1, 2, ..., N$. The line close to 1 is for the true identity. (b) Scale estimate against time t.

Table I Comparison of tracking results obtained by particle filters with different configurations. 'A_t size' means pixel size in the component(s) of the appearance model. 'o' means success in tracking. 'x' means failure in tracking.