Appearance Tracking Using Adaptive Models in a Particle Filter

Shaohua Zhou, Rama Chellappa, Baback Moghaddam

TR2004-027 December 2004

Abstract

The particle filter is a popular tool for visual tracking. Usually, 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 used is typically fixed. All these factors make the visual tracker unstable. To stabilize the tracker, we propose the following measures: an observation model arising from an adaptive noise variance, and adaptive number of particles. The adaptive-velocity is computed via a first-order linear predictor using the previous particle configuration. Tracking under occlusion is accomplished using robust statistics. Experimental results on tracking visual objects in long video sequences such as vehicles, tank, and human faces demonstrate the effectiveness and robustness of our algorithm.

Proceedings of the 6th Asian Conference on Computer Vision (ACCV), Jeju Island, Korea, Jan. 27-30th, 2004.

Publication History:–

1. First printing, TR2004-027, January 2004

APPEARANCE TRACKING USING ADAPTIVE MODELS IN A PARTICLE FILTER

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

[1]Center for Automation Research & ECE Department
University of Maryland, College Park, MD 20742

[2] Mitsubishi Electric Research Laboratories
201 Broadway, Cambridge, MA 02139

ABSTRACT

The particle filter is a popular tool for visual tracking. Usually, 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 used is typically fixed. All these factors make the visual tracker unstable. To stabilize the tracker, we propose the following measures: an observation model arising from an adaptive appearance model, a velocity motion model with adaptive noise variance, and adaptive number of particles. The adaptive-velocity is computed via a first-order linear predictor using the previous particle configuration. Tracking under occlusion is accomplished using robust statistics. Experimental results on tracking visual objects in long video sequences such as vehicles, tank, and human faces demonstrate the effectiveness and robustness of our algorithm.

1. INTRODUCTION

Particle filter [3] 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

$$\theta_t = F_t(\theta_{t-1}, U_t), \;\; Y_t = G_t(\theta_t, V_t), \tag{1}$$

where U_t is the motion noise, $F_t(.,.)$ characterizes dynamics, 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 a minimum mean square error (MMSE) estimate,

$$\hat{\theta}_t = \theta_t^{mmse} = E[\theta_t|Y_{0:t}] \approx J^{-1} \sum_{j=1}^{J} w_t^{(j)} \theta_t^{(j)}, \tag{2}$$

or a maximum a posteriori (MAP) estimate,

$$\hat{\theta}_t = \theta_t^{map} = \arg\max_{\theta_t} p(\theta_t|Y_{1:t}) \approx \arg\max_{\theta_t} w_t^{(j)}, \tag{3}$$

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

The state transition model characterizes the motion change between frames. 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 [6]. However such a model may overfit the training data and may not necessarily succeed with the testing video where objects can move arbitrarily at different times and places. Also one cannot always rely on the availability of training data in the first place. (ii) Secondly, a fixed constant-velocity model with fixed noise variance is fitted for simplicity [12, 13].

$$\theta_t = \theta_{t-1} + U_t, \tag{4}$$

where U_t has a fixed noise variance, say $U_t = R_0 * U_0$ where R_0 is a fixed constant measuring the noisy extent and U_0 is a 'standardized' random variable/vector [1]. If R_0 is small, it is very hard to model the rapid movement; 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 the 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 [6], another class of tracking approaches [4, 10, 13] exploits an appearance model A_t. In its simplest (yet mostly used) form, we have the following observation equation [2],

$$Z_t = \mathcal{T}\{Y_t; \theta_t\} = A_t + V_t, \tag{5}$$

where Z_t is the image patch of interest in the video frame Y_t, parameterized by θ_t. In [4], 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 [10], say, $A_t = \hat{Z}_{t-1}$, i.e., the 'best' patch of interest in the previous frame. However, a fixed template cannot handle the appearance changes in the video, while a rapidly changing model is susceptible to drift. Recent research efforts make a compromise. The approach proposed in [7] uses a mixture appearance model, consisting of a slow-varying appearance component, a fast-changing appearance component, and an occlusion component as well. The mixture appearance model is also recursively updated.

Supported in part by Mitsubishi Electric Research Laboratories and Advanced Sensors Consortium sponsored by U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0008. The authors thank Mike Jones for providing the face detector [11].

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

[2] For the sake of brevity in notation, 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.

Our approach to visual appearance tracking is to make both observation and state transition models adaptive in the framework of a particle filter, with occlusion handling embedded. It possesses the following features:

(i) Adaptive observation model (Section 2). We adopt an appearance-based approach using Eq. (5). To make the observation model adaptive, we make the appearance model A_t in Eq. (5) adaptive, i.e., the appearance model is updated incrementally with the incoming observations.

(ii) Adaptive state transition model (Section 3). 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 and the particle configuration of the previous frame. 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, and vary the number of particles based on the degree of uncertainty R_t in the noise component.

(iii) Handling occlusion (Section 4). Occlusion is handled using robust statistics [5]. We robustify the likelihood measurement and the velocity estimate by down-weighting the 'outlier' pixels. If occlusion is declared, we stop updating the appearance model and estimating motion velocity.

Section 5 presents our experimental results on tracking vehicles, tank, and human faces and Section 6 concludes the paper.

2. ADAPTIVE OBSERVATION MODEL

The adaptive observation model arises from the adaptive appearance model A_t inspired by [7]. In [7], the appearance model is based on phase information derived from the image intensity whose computation is quite time-consuming. The direct embedding of such model in a particle filter further increase the computational burden. Thus, in this paper, we simply use the intensity-based appearance model.

2.1. Mixture appearance model

The mixture appearance model assumes that the observations are explained by different causes, thereby indicating the use of a mixture density of components. In [7], 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. However, in our implementation, we have not incorporated the L-component because (i) it is not easy to characterize the statistics of outlier pixels if image intensities are used and (ii) we will model the occlusion in a different manner as shown in Sec. 4.

As an option, in order to further stabilize our tracker one could use an F-component which is a fixed template that we observe most often. 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.

We now describe our 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 is independent of each other. In summary, the observation likelihood is written as

$$p(Y_t|\theta_t) = \prod_{j=1}^{d} \{ \sum_{i=w,s,f} m_{i,t}(j) \mathsf{N}(Z_t(j); \mu_{i,t}(j), \sigma^2_{i,t}(j)) \}, \quad (6)$$

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

2.2. Model update and initialization

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

We assume that all past data observations are exponentially 'forgotten' with respect to their contributions to the current appearance model. Denote the 'forgetting' factor by α. Below, we just sketch the updating equations as follows and refer the interested readers to [7] for technical details and justifications.

The EM algorithm is invoked. Firstly, the posterior responsibility probabilities $\{o_{i,t}(j);\ i = w, s, f\}$ (with $\sum_i o_{i,t}(j) = 1$) are computed as

$$o_{i,t}(j) \propto m_{i,t}(j) \mathsf{N}(\hat{Z}_t(j); \mu_{i,t}(j), \sigma^2_{i,t}(j));\ \ i = w, s, f. \quad (7)$$

Then, the mixing probabilities are updated as

$$m_{i,t+1}(j) = \alpha\, o_{i,t}(j) + (1-\alpha)\, m_{i,t}(j);\ \ i = w, s, f, \quad (8)$$

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

$$M_{p,t+1}(j) = \alpha\, \hat{Z}^p_t(j) o_{s,t}(j) + (1-\alpha)\, M_{p,t}(j);\ \ p = 1, 2. \quad (9)$$

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

$$S_{t+1}(j) = \mu_{s,t+1}(j) = \frac{M_{1,t+1}(j)}{m_{s,t+1}(j)},\ \sigma^2_{s,t+1}(j) = \frac{M_{2,t+1}(j)}{m_{s,t+1}(j)} - \mu^2_{s,t+1}(j). \quad (10)$$

$$W_{t+1}(j) = \mu_{w,t+1}(j) = \hat{Z}_t(j),\ \ \sigma^2_{w,t+1}(j) = \sigma^2_{w,1}(j), \quad (11)$$

$$F_{t+1}(j) = \mu_{f,t+1}(j) = F_1(j),\ \ \sigma^2_{f,t+1}(j) = \sigma^2_{f,1}(j). \quad (12)$$

To initialize A_1, we set $W_1 = S_1 = F_1 = T_0$ (with T_0 supplied by a detection algorithm), $\{m_{i,1}, \sigma^2_{i,1};\ i = w, s, f\}$, $M_{1,1} = m_{s,1} T_0$, and $M_{2,1} = m_{s,1}(\sigma^2_{s,1} + T^2_0)$.

3. ADAPTIVE STATE TRANSITION MODEL

3.1. Adaptive velocity

With the availability of the sample set $\Theta_{t-1} = \{\theta^{(j)}_{t-1}\}^J_{j=1}$ and the image patches of interest $\mathcal{Z}_{t-1} = \{Z^{(j)}_{t-1}\}^J_{j=1}$, 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 [4, 8, 1].

The constant brightness constraint tells that there exists a θ_t such that $\mathcal{T}\{Y_t; \theta_t\} = \hat{Z}_{t-1}$. Approximating $\mathcal{T}\{Y_t; \theta_t\}$ via a first-order Taylor series expansion around $\hat{\theta}_{t-1}$ yields

$$\begin{aligned} \mathcal{T}\{Y_t; \theta_t\} &\simeq \mathcal{T}\{Y_t; \hat{\theta}_{t-1}\} + C_t(\theta_t - \hat{\theta}_{t-1}) \\ &= \mathcal{T}\{Y_t; \hat{\theta}_{t-1}\} + C_t \nu_t, \end{aligned} \quad (13)$$

where C_t is the Jacobi matrix. Substituting $\hat{Z}_{t-1}$ into (13) gives

$$\hat{Z}_{t-1} \simeq \mathcal{T}\{Y_t; \hat{\theta}_{t-1}\} + C_t \nu_t, \tag{14}$$

i.e.,

$$\nu_t \simeq -B_t(\mathcal{T}\{Y_t; \hat{\theta}_{t-1}\} - \hat{Z}_{t-1}), \tag{15}$$

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:

$$\Theta^{\delta}_{t-1} = [\theta^{(1)}_{t-1} - \hat{\theta}_{t-1}, \ldots, \theta^{(J)}_{t-1} - \hat{\theta}_{t-1}], \tag{16}$$

$$\mathcal{Z}^{\delta}_{t-1} = [Z^{(1)}_{t-1} - \hat{Z}_{t-1}, \ldots, Z^{(J)}_{t-1} - \hat{Z}_{t-1}]. \tag{17}$$

The least square (LS) solution for B_t is

$$B_t = (\Theta^{\delta}_{t-1} \mathcal{Z}^{\delta\,\mathsf{T}}_{t-1})(\mathcal{Z}^{\delta}_{t-1} \mathcal{Z}^{\delta\,\mathsf{T}}_{t-1})^{-1}, \tag{18}$$

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 exceeds the data dimension). To overcome this, we use the singular value decomposition (SVD) of $\mathcal{Z}^{\delta}_{t-1}$, say $\mathcal{Z}^{\delta}_{t-1} = USV^{\mathsf{T}}$. It can be easily shown that $B_t = \Theta^{\delta}_{t-1} V S^{-1} U^{\mathsf{T}}$. Also, we can gain some computational efficiency by sacrificing some accuracy, i.e., we can further approximate B_t by retaining the top q components: $B_t = \Theta^{\delta}_{t-1} V_q S_q^{-1} U_q^{\mathsf{T}}$.

The following state transition model is the one we used,

$$\theta_t = \hat{\theta}_{t-1} + \nu_t + U_t. \tag{19}$$

The choice of U_t is discussed below.

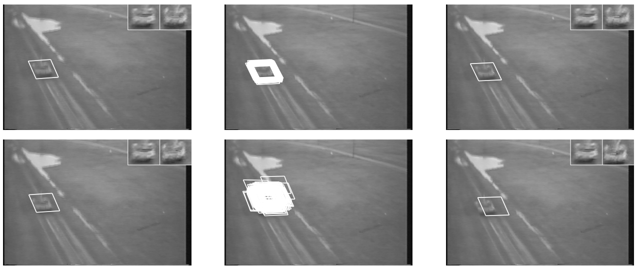

time $t-1$ particles time t

Fig. 1. Particle configurations from (top row) adaptive velocity model and (bottom row) zero-velocity model. Each particle contributes one bounding box imposed on the images (central column).

3.2. Adaptive noise and adaptive number of particles

In practice, the above prediction cannot be exact and usually results in a prediction error ϵ_t which determines the quality of prediction. If ϵ_t is small, which implies a good prediction, we only need tightly-supported noise to absorb the residual motion; if ϵ_t is large, which implies a poor prediction, we then need widespread noise 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. In our algorithm, we compute ϵ_t as the average of the square of the error residual $\mathcal{T}\{Y_t; \hat{\theta}_{t-1} + \nu_t\} - \hat{Z}_{t-1}$, which is a 'variance'-type measure. Thus, we use

$$R_t = \max(\min(R_0\sqrt{\epsilon_t}, R_{max}), R_{min}), \tag{20}$$

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

If the noise variance R_t is large, we need more particles to cover the spreading density, while conversely, fewer particles are needed for noise with small variance R_t. Based on the principle of asymptotical relative efficiency (ARE) [2], we should adjust the particle number J_t in a similar fashion, i.e., $J_t = J_0 R_t / R_0$.

We demonstrate the necessity of the adaptive velocity model by comparing it with the zero velocity model. Fig. 1 shows the particle configurations created from the adaptive velocity model (with $J_t < J_0$ and $R_t < J_0$) 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, which leads to an unsuccessful tracking since widely distributed particles could get trapped in local minima.

```
**Initialize** a sample set $\mathcal{S}_0 = \{\theta^{(j)}_0, 1/J_0)\}^{J_0}_{j=1}$ according to prior distribution $p(\theta_0)$ and the appearance model $A_1$. Set $R_0$ and $J_0$. Let $OCC_{FLAG} = 0$, indicating no occlusion.
**For** $t = 1, 2, \ldots$
  **If** ($OCC_{FLAG} == 0$)
    **Calculate** the state estimate $\hat{\theta}_{t-1}$ by Eq. (2) or (3), adaptive velocity $\nu_t$ by Eq. (15), noise variance $R_t$ by Eq. (20), and particle number $J_t$.
  **Else**
    $R_t = R_{max}, J_t = J_{max}, \nu_t = 0.$
  **End**
  **For** $j = 1, 2, \ldots, J_t$
    **Draw** the sample $U^{(j)}_t$ for $U_t$ with variance $R_t$.
    **Construct** the sample $\theta^{(j)}_t = \hat{\theta}_{t-1} + \nu_t + U^{(j)}_t$ by Eq. (19).
    **Compute** the transformed image $Z^{(j)}_t$.
    **Update** the weight using $w^{(j)}_t = p(Y_t|\theta^{(j)}_t)$.
  **End**
  **Normalize** the weight using $w^{(j)}_t = w^{(j)}_t / \sum^{J}_{j=1} w^{(j)}_t$.
  **Set** $OCC_{FLAG}$ according to the number of outlier pixels in $\hat{Z}_t$.
  **If** ($OCC_{FLAG} == 0$)
    **Update** the appearance model $A_{t+1}$ using $\hat{Z}_t$.
  **End**
**End**
```

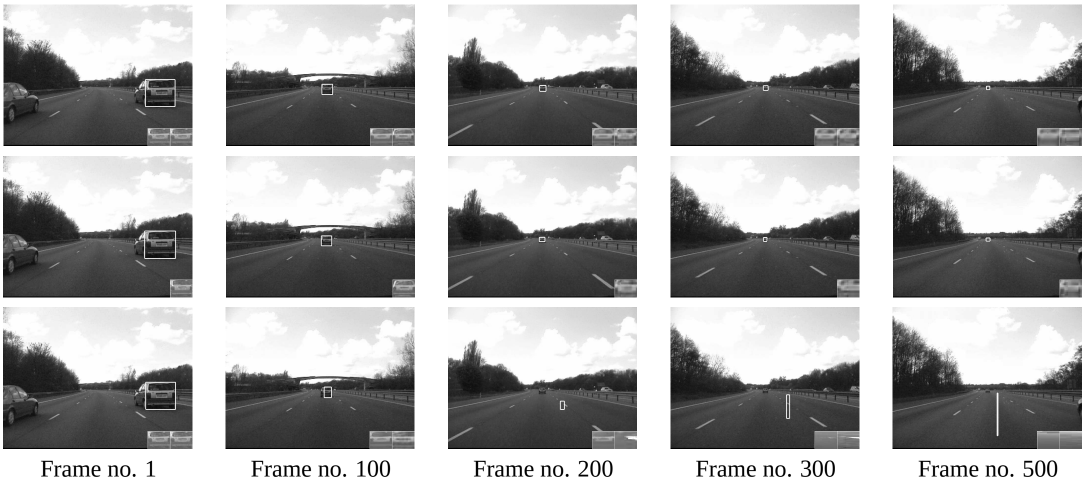
Fig. 2. The proposed algorithm.

4. HANDLING OCCLUSION

Occlusion is usually handled in two manners. One way is to use joint probabilistic data associative filter (JPDAF) [9]; and the other is to use robust statistics [5]. We use robust statistics here.

4.1. Robust statistics

We assume that occlusions produce large image differences which can be treated as 'outlier' that cannot be explained by the underlying process or its influence on the estimation or measurement process should be reduced. Robust statistics provides such mechanisms.



Frame no. 1 Frame no. 100 Frame no. 200 Frame no. 300 Frame no. 500

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

We use the ρ function defined as follows:

$$\rho(x) = \left\{ \begin{array}{lll} x^2/2 & if & |x| \leq c \\ cx - c^2/2 & if & |x| > c \end{array} \right. , \qquad (21)$$

where x is normalized to have unit variance and the constant c control the outlier rate. In our experiment, we set $c = 1.435$. If $|x| > c$ is satisfied, we declare the corresponding pixel an outlier.

4.2. Robust likelihood measure and adaptive velocity estimate

The likelihood measure defined in Eq. (6) invovles a multi-dimensional normal density. Since we assume that each pixel is independent, we can only consider the one-dimensional normal density. To robustify our likelihood measure, we replace the one-dimensional normal density $\mathsf{N}(x; \mu, \sigma^2)$ as

$$\hat{\mathsf{N}}(x; \mu, \sigma^2) = (2\pi\sigma^2)^{-1/2} \exp(-\rho(\frac{x-\mu}{\sigma})). \qquad (22)$$

Note that there 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 brightness constancy 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 with its i^{th} element being $L_i = \eta(x_i)$ where x_i is the pixel intensity of the difference image $(\mathcal{T}\{Y_t; \hat{\theta}_{t-1}\} - \hat{Z}_{t-1})$ normalized by the variance of the W_t component, and

$$\eta(x) = \frac{1}{x}\frac{d\rho(x)}{dx} = \left\{ \begin{array}{lll} 1 & if & |x| \leq c \\ c/|x| & if & |x| > c \end{array} \right. , \qquad (23)$$

Eq. (15) becomes

$$\nu_t \simeq -B_t L(\mathcal{T}\{Y_t; \hat{\theta}_{t-1}\} - \hat{Z}_{t-1}). \qquad (24)$$

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

4.3. Occlusion declaration

If the number of the outlier pixels in $\hat{Z}_t$ (compared with the appearance model), say d_{out}, exceeds a certain threshold, i.e., $d_{out} > \lambda d$ where $0 < \lambda < 1$ (we take $\lambda = 0.15$), we declare an occlusion. Since the appearance model has more than one components, we count the number of the 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 motion velocity to zeros, 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 $J_t = J_{max}$.

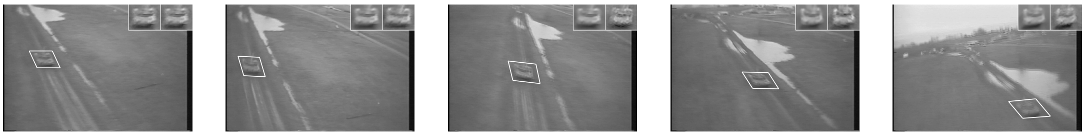
Finally, our adaptive particle filtering algorithm with occlusion analysis is summarized in Fig. 2.

5. EXPERIMENTAL RESULTS

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.

We demonstrate the effectiveness of our algorithm by tracking a disappearing car, and an arbitrarily-moving tank, and a moving face under occlusion. Table 1 summarizes some statistics about the video sequence, and the appearance mode size used. Accompanying video sequences with tracking results are available are http://www.cfar.umd.edu/∼shaohua/research.html.

We initialize the particle filter and the appearance model with a detector algorithm (The face detector [11] is used for the face



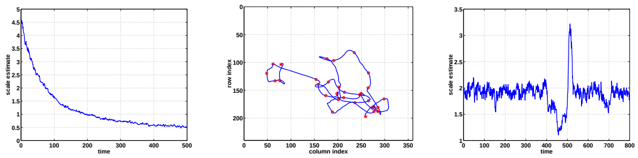
Frame no. 1 Frame no. 31 Frame no. 49 Frame no. 228 Frame no. 300

Fig. 4. The tank sequence.

sequence) or a manually specified image patch in the first frame. R_0 and J_0 are also manually set, depending on the sequence.

5.1. Car tracking

We first test our algorithm to track a vehicle with the F-component but without the occlusion analysis. The tracking result of a fast moving car is shown in Fig. 3 with a bounding box. We also show the stable and wandering components 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 filters successfully tracked this rapid change in scale (where scale is a function of all four affine parameters). Fig. 5(a) plots the scale estimate recovered by our algorithm.[3] It is clear that the scale follows a decreasing trend as time proceeds. The size of the image block containing the car in the final frame is about 12 by 15, which makes the vehicle almost invisible. In this sequence we set $J_0 = 50$. The average number of particles used is about 40, which means that in this case we actually saved about 20% in computation by using an adaptive J_t instead of a fixed particle number J_0. The algorithm's Matlab implementation needs about 1.2 frames per second running on a PC with a PIII 650 CPU and 512M memory.



(a) (b) (c)

Fig. 5. (a) The scale estimate for the car. (b) The 2-D trajectory of the centroid of the tank. '*' means the starting and ending points and '.' points are marked along the trajectory every 10 frames. (c) The scale estimate for the face sequence.

5.2. Tank tracking in an aerial video

Fig. 4 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 tank every 10 frames, covering from the left to the right in 300 frames. One extreme is that the tank moved about 100 pixels in column index in 10 frames, which might fail trackers with fixed models. But, our tracking is still successful.

5.3. Face tracking

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

Fig. 6 presents the tracking results on the video sequence featuring the following variations: moderate pose and 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 appearance model remains fixed during occlusion.

Fig. 7 presents the tracking results obtained by 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) dominates the image difference $(\mathcal{T}\{Y_t; \hat{\theta}_{t-1}\} - \hat{Z}_{t-1})$. Updating the appearance model deteriorates the situation.

Fig. 5(c) 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).

5.4. Comparison

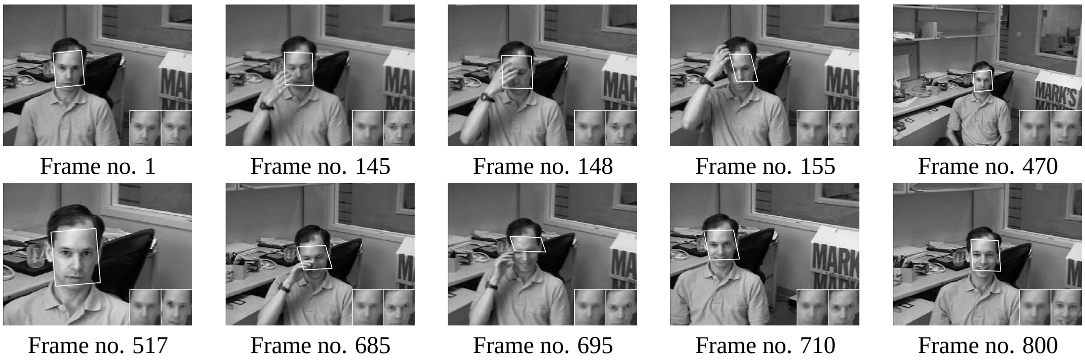
We illustrate the effectiveness of our adaptive approach ('adp') by comparing the particle filter either with (a) an adaptive motion model but a fixed appearance model ('fa'), or with (b) a fixed motion model but an adaptive appearance model ('fm'); or with (c) a fixed motion model and a fixed appearance model ('fb'). Table 1 lists the tracking results obtained using particle filters under all the above situations, where 'adp & occ' indicates an adaptive approach with occlusion handling. Fig. 3 shows the tracking results on the car sequence when 'fa' and 'fm' options are used.

Table 1 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. When occlusion exists in the video sequence, we must use occlusion handling technique.

6. CONCLUSIONS

We have presented an adaptive paradigm for visual tracking which stabilizes the tracker by embedding deterministic linear prediction into stochastic diffusion. Numerical solutions have been provided

[3] The scale estimate is calculated as $\sqrt{(a_1^2 + a_2^2 + a_3^2 + a_4^2)/2}$.


Frame no. 1 Frame no. 145 Frame no. 148 Frame no. 155 Frame no. 470

Frame no. 517 Frame no. 685 Frame no. 695 Frame no. 710 Frame no. 800

Fig. 6. The face sequence. Frames 145, 148, and 155 show the first occlusion. Frame 470 and 517 show the smallest and biggest face observed. Frame 685, 690, 710 show the second occlusion.

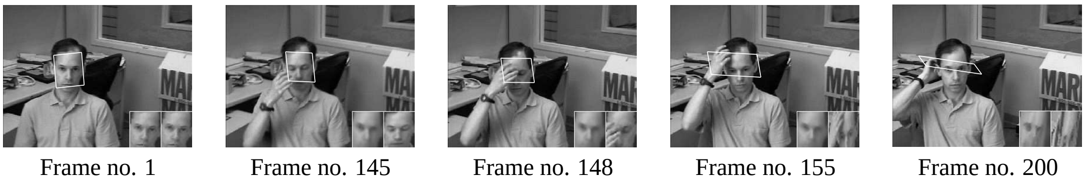

Frame no. 1 Frame no. 145 Frame no. 148 Frame no. 155 Frame no. 200

Fig. 7. Tracking results on the face sequence using the adaptive particle filter with the occlusion analysis.

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 1. 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.

by particle filters equipped with adaptivity: an adaptive observation model arising from the adaptive appearance model, an adaptive state transition model, and adaptive number of particles. Occlusion analysis is also embedded in the particle filter. Our algorithm was then tested on several tasks consisting of tracking visual objects such as car and human face in realistic scenarios. Good tracking results are obtained due to using appropriate choices for both state transition and observation models in a particle filter.

7. REFERENCES

[1] A. Bergen, P. Anadan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. *Proc. of ECCV*, pages 237–252, 1992.

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

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

[4] G. D. Hager and P. N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Trans. on PAMI*, 20(10):1025–1039, 1998.

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

[6] M. Isard and A. Blake. Contour tracking by stochatic propagation of conditional density. *Proc. of ECCV*, 1996.

[7] A. D. Jepson, D. J. Fleet, and T. El-Maraghi. Robust online appearance model for visual tracking. *Proc. of CVPR*, 1:415–422, 2001.

[8] F. Jurie and M. Dhome. A simple and efficient template matching algorithm. *Proc. ICCV*, 2:544–549, 2001.

[9] C. Rasmussen and G.D. Hager. Probabilistic data association methods for tracking complex visual objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):560–576, 2001.

[10] H. Sidenbladh, M. J. Black, and D. J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. *Proc. of ECCV*, 2:702–718, 2002.

[11] P. Voila and M. Jones. Robust real-time object detection. *Second Intl. Workshop on Stat. and Comp. Theories of Vision*, 2001.

[12] Y. Wu and T. S. Huang. A co-inference approach to robust visual tracking. *Int. Conf. on Computer Vision*, 2:26–33, 2001.

[13] S. Zhou and R. Chellappa. Probabilistic human recgnition from video. *Proc. of ECCV*, 3:681–697, 2002.