# A Comparative Study of Boosted and Adaptive Particle Filters for Affine-Invariant Target Detection and Tracking

Guoliang Fan, Vijay Venkataraman, Li Tang
School of Electrical and Computer Engineering
Oklahoma State University, Stillwater, OK
glfan@okstate.edu

Joseph Havlicek
School of Electrical and Computer Engineering
University of Oklahoma, Norman, OK
joebob@ou.edu

## Abstract

*This paper addresses a specific multi-aspect target detection and tracking problem where the dynamics of the target's aspect is modeled by an affine model following a first-order Markov process. We are interested in how to achieve robust and accurate Monte Carlo estimation in a high-dimensional state space with poor target visibility by re-visiting two recent improvements to particle filters, i.e., "boosting" and "adapting". The impetus of this work is a tracking indicator that estimates the tracking performance based on the observation model and may trigger either one of two actions when it is necessary. One is "boosting", i.e., the detector specified by the tracker's previous output is involved to induce more promising particles, and the original idea of "boosting" is extended here by encouraging positive interaction between the detector and the tracker. The other is "adapting", i.e., the system model can self-adjust to enhance the tracking capability. We compare two methods in the context of affine-invariant target tracking and with respect to their contributions to improve the particle quality. Experiments on simulated image sequences with real infrared background show that both techniques can improve the tracking performance by balancing the focus and the diversity of particle distribution.*

## 1. Introduction

Target tracking is usually formulated as an estimation problem of a dynamic state-space system, where the state stores the target's kinematic characteristics and two models are involved, i.e., the system model and the observation model. Given a probabilistic state-space formulation, the tracking problem is well suited for the recursive Bayesian approach which attempts to construct the posterior probability density function (PDF) of the state based on all state and observation information available. Recently particle filtering has received more and more attention because of its ability and flexibility to deal with nonlinear/non-Gaussian estimation problems by approximating a continuous density as a discrete one [8]. The key idea is to represent the required PDF by a set of weighted particles and to estimate the state based on these weighted particles. A recursive Bayesian filter can be implemented by Monte Carlo simulations to continuously update particles and associated weights. Also, it is important to control the quality of particles, i.e., the balance between diversity (having multiple distinct samples) and focus (having multiple copies of samples with large weights) [7].

State estimation in high-dimensional systems requires the particle number to increase exponentially. There are two different but relevant methodologies to improve the quality and efficiency of particles, "bottom-up" and "top-down". The bottom-up methods control the particle quality via direct particle modifications, e.g., particle re-weighting/re-sampling [7], the kernel based particle filter [5, 9], the hybrid particle filter and mean-shift tracker [11], and the annealed particle filter [6]. The top-down methods focus on the problem formulation, i.e., the system model (the prior) and the observation model (the likelihood). On the one hand, the target is assumed to follow a certain motion model, e.g., a white noise acceleration model [2] or a random walk model [4]. Adaptive motion models were also developed which can be learned from the arrival of the new observations, e.g., [17, 11]. The Adaboost particle filter [14] incorporates the detection hypothesis in the proposal distribution where the detector is involved as an independent process to track new targets. On the other hand, the observation model mainly depends on target descriptions and/or sensor models. For a constant target template, the function of Sum of Squared Distance (SSD) in intensity is minimized to determine the likelihood. Other features are also employed to describe targets/objects, such as edges or histograms in a region of ellipses [5] or rectangles [9] with a varying scale and a fixed aspect ratio. When the geometric contour is used [16], the boundary of an object evolves as an active contour, and the likelihood function is derived by minimizing an image-based energy function. Moreover, the affine model was used in many trackers to support robust and accurate tracking for deforming targets [17, 16].

Most of the aforementioned approaches are mainly for optical images where the objects of interest usually have relatively high visibility and large sizes. In the context of small target tracking in infrared image sequences or other remotely sensed imageries with low SNRs, poor target visibility and unknown dynamics of the target's aspects make the tracking problem more challenging. More prior knowledge about system/observation models are needed to ensure robust tracking performance. For example, the one proposed in [3, 4] assumes that the sensor clutter is modeled as the first-order Gaussian Markov Random Field (GMRF) and a finite set of multi-aspect target signatures are given and represented by discrete-valued indices in the state vector. High tracking performance was reported in [3] on simulated infrared image sequences under low SNRs ranging from 7dB to -5.7dB. Two key elements in [3, 4] are the highly selective likelihood function derived from the GRMF and the pre-defined multi-aspect target templates as well as their transition probabilities. Motivated by previous work, we consider a more general multi-aspect tracking problem for infrared image sequences by involving a continuous-valued affine model to accommodate dynamics of target's aspect, e.g., rotation and scaling. Specifically, we assume that the affine model follows the first-order Markov chain and is incorporated into the state/observation models. With this formulation, we can describe more realistic target motion and aspect dynamics. However, due to the complication of the state space (two additional continuous state variables), the traditional particle filtering algorithms, such as Sequential Importance Re-sampling (SIR) and Auxiliary Particle Filter (APF) discussed in [3] fail to provide satisfactory results under the new formulation.

In this work, we will re-visit two recent improvements to particle filters, "boosting" and "adapting". The former one is often used to involve an independent process for adding new tracks [14]. The latter one is able to adjust the system/observation models upon the arrival of new observations [17]. Both techniques are proven efficient for object tracking in optical images with relatively high object visibility, and here we are interested in how to achieve robust and accurate Monte Carlo estimation in a high-dimensional state space with poor target visibility. The impetus of this work is a tracking indicator that estimates the tracking performance based on the observation model and may trigger either one of two actions when necessary. One is "boosting", i.e., the detector specified by tracker's previous output is involved to induce more promising particles, and the original idea is extended here by encouraging positive interaction between the detector and the tracker. The other is "adapting", i.e., the system model can self-adjust to enhance the tracking capability. We compare two methods in the context of affine-invariant target tracking and with respect to their contributions to improve the particle quality, i.e., the balance between focus and diversity.

## 2. Problem formulation

We first briefly discuss system and observation models that are detailed in [2], then we extend the formulation by introducing the affine-invariant target model.

### 2.1. System and Observation Models

Let $\Delta$ denote the time interval between two consecutive observation frames. The state vector at instant $t = k\Delta$ ($k \in \mathbb{N}$) of a target typically consists of position ($x_k$,$y_k$) and velocity ($\dot{x_k}$,$\dot{y_k}$), of its centroid in a 2D Cartesian coordinate system: $\mathbf{x}_k = [x_k \; \dot{x_k} \; y_k \; \dot{y_k}]^T$. The position and velocity in two directions are assumed to be independent and evolve over time according to the white noise acceleration model [3], and the state is updated according to,

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{w}_{k-1} \; . \quad (1)$$

Let the transitional matrix along $x$ and $y$ dimensions be $\mathbf{F_x} = \mathbf{F_y} = [1 \; \Delta; \; 0 \; 1]$, then $\mathbf{F} = [\mathbf{F_x} \; \mathbf{0}; \; \mathbf{0} \; \mathbf{F_y}]$ and the process noise $\mathbf{w}_k$ is assumed to be white and zero-mean Gaussian.

The observation matrix $\mathbf{z}_k$ collects the observations $\{(i,j)|1 \leq i \leq L, 1 \leq j \leq M\}$ at instant $t = k\Delta$:

$$\mathbf{z}_k = \mathbf{H}(\mathbf{x}_k) + \mathbf{v}_k \; . \quad (2)$$

This observation model is generated by adding a noise field $\mathbf{v}_k$ with a template of a specific intensity distribution. $\mathbf{H}$ is a function of the state vector $\mathbf{x}_k$ which produces a noise free template of the target with the exact position and velocity specified by $\mathbf{x}_k$.

We assume that $\mathbf{w}_k$ in the state model is statistically independent with $\mathbf{v}_k$ in the observation model [4]. The clutter frames $\{\mathbf{v}_k | k \in \mathbb{N}\}$ are assumed to be independent, identically distributed (i.i.d.) Gaussian random sequences with zero mean and non-singular covariance matrices. Each frame is described by the first order Gaussian Markov Random Field (GMRF) given in [13]:

$$\mathbf{v}_k(i,j) = \beta_v^c[v_k(i-1,j) + v_k(i+1,j)] + \beta_h^c \cdot[v_k(i,j-1) + v_k(i,j+1)] + \varepsilon_k(i,j) \; , \quad (3)$$

where the unknown parameters $\beta_v^c$ and $\beta_h^c$ are, respectively, the vertical and horizontal predictor coefficients, and $\varepsilon_k$ is the prediction error such that [2]

$$E[v_k(i,j)\varepsilon_k(l,r)] = \sigma_{c,k}^2 \delta_{i-l,j-r} \; . \quad (4)$$

We can estimate the GMRF parameters, i.e., $\hat{\beta}_h$, $\hat{\beta}_v$ and $\hat{\sigma}_c^2$, for each frame $\mathbf{z}_k$ via a suboptimal approximate maximum likelihood (AML) algorithm in [12].

## 2.2. Affine Invariant Target Model

In [2], a set of target templates with different aspects are used for multi-aspect target tracking, and an aspect index is introduced in the state vector, i.e., $\mathbf{x}_k$, to account for the aspect variability. In this work, we want to generalize multi-aspect tracking by introducing the continuous-valued affine model which is often used to track deforming objects [17, 16]. Hereby, we incorporate the affine model to account for target's various aspects in each time step, which is formulated by applying a $3 \times 3$ affine transformation $\mathbf{T}_a$ to every pixel $(x, y)$ of the base template of the target.

$$\mathbf{T}_a = \begin{bmatrix} 1 & \alpha & -\alpha y \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & (1-s_x)x \\ 0 & s_y & (1-s_y)y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & \sin\theta & (1-\cos\theta)x - (\sin\theta)y \\ -\sin\theta & \cos\theta & (1-\cos\theta)y + (\sin\theta)x \\ 0 & 0 & 1 \end{bmatrix}, \quad (5)$$

where $\alpha$, $s_x$, $s_y$ and $\theta$ are the shearing, scaling along x-axis, scaling along y-axis and rotation parameters, respectively. All of these are continuous-valued random variables which follow a first order Markov chain with equal transition probability (i.e., 1/3) of increasing, decreasing by a quantization step ($\Delta_\alpha$, $\Delta_{s_x}$, $\Delta_{s_y}$, $\Delta_\theta$), or staying at the same value in each time instant. A process noise ($\gamma_\alpha$, $\gamma_{s_x}$, $\gamma_{s_y}$, $\gamma_\theta$) is added to each random variable to reduce the quantization effect. For example, the transition probability function $p(\theta_k|\theta_{k-1})$ for the rotation angle $\theta$ is represented as:

$$p(\theta_k|\theta_{k-1}) = \begin{cases} 1/3 & \text{when } \theta_k = \theta_{k-1} - \Delta_\theta + \gamma_\theta, \\ 1/3 & \text{when } \theta_k = \theta_{k-1} + \gamma_\theta, \\ 1/3 & \text{when } \theta_k = \theta_{k-1} + \Delta_\theta + \gamma_\theta. \end{cases} \quad (6)$$

Therefore, we can define a new augmented state vector as:

$$\mathbf{x}_k = [x_k \ \dot{x_k} \ y_k \ \dot{y_k} \ s_x^k \ s_y^k \ \alpha_k \ \theta_k]^T. \quad (7)$$

The clutter free target frame $\mathbf{H}(\mathbf{x}_k)$ in the observation model is obtained by performing a specific affine transform on the original target template. It is worth mentioning that the computation of $\mathbf{H}(\mathbf{x}_k)$ involves linear interpolation of the original target template under the affine model. For simplicity, we set the shearing factor $\alpha = 0$ and $s = s_x = s_y$ in our simulations to avoid unrealistic distortion. It should be noted that although the above formulation seems to be a straightforward extension of the previous one, the complexity of the high dimensional state space makes conventional particle filters unsatisfactory in practice. Particularly, the continuous-valued affine parameters make the target detection/tracking problems more challenging at low/moderate SNRs. We are interested in how to remedy the limitation of traditional particle filters for this extended formulation.

## 2.3. Likelihood Functions of Observation

The likelihood function is rooted in the structured nature of the sensor noise in infrared images which can be modeled as a first-order GMRF [13]. Let $Z_k$ and $h(\mathbf{x}_k)$ be 1D representations of the observed frame $\mathbf{z}_k$ and the clutter free target frame $H(\mathbf{x}_k)$ in (2), obtained by reading a frame row by row and stacking the rows as one long vector. Similarly, let $V_k$ denote the 1D vector representation of the clutter frame $\mathbf{v}_k$ defined in (3). Then the likelihood function is given by

$$p(Z_k|\mathbf{x}_k) \propto \exp\left[\frac{2\lambda(Z_k) - \rho(\mathbf{x}_k)}{2\sigma_{c,k}^2}\right], \quad (8)$$

where $\rho(\mathbf{x}_k)$ is the *energy term* depending on the current target template (with certain rotation and scaling factors) as

$$\rho(\mathbf{x}_k) = h^T(\mathbf{x}_k)(\sigma_{c,k}^2 \Sigma_v^{-1})h(\mathbf{x}_k), \quad (9)$$

and $\lambda(Z_k)$ is the *data term* depending on observation $Z_k$ and target state $\mathbf{x}_k$,

$$\lambda(Z_k) = Z_k^T(\sigma_{c,k}^2 \Sigma_v^{-1})h(\mathbf{x}_k). \quad (10)$$

$\lambda(Z_k)$ is the match filtering result between the observed frame ($\mathbf{z}_k$) and a scaled/rotated target template ($H(\mathbf{x}_k)$). Both the likelihood function and the data term are a function of three continuous variables, i.e., the position, the rotation, and the scaling factors. We compare their sensitivity and selectivity with respect to every two variables in Fig. 1. It is seen that the likelihood function is much more sensitive and selective than the data term.

An ideal likelihood is suggested to have three characteristics in [10]. (1) It should have a flat peak area to provide enough robustness for the small difference between particles close to the true state. (2) There should be a significant difference between good particles and the bad ones. (3) Those truly bad particles should get equal weights to avoid being trapped in a local maximum of background clutter. As shown in Fig. 1, the likelihood function defined in (8) is highly selective with a very strong peak in the true state, and this warrants high accuracy and sensitivity of target detection/tracking under low SNRs [2]. However, when the state space is complicated by adding more continuous state variables, this characteristic may have some drawbacks. It is as if a person (Monte Carlo estimation) is looking for a needle (the optimal target state) in a big dark room (a high-dimensional state space) using a torch with needle-aperture (a highly selective likelihood function). Then particle filtering tends to be highly computationally expensive and could be easily trapped into a local optimum. In this work, we do not want to change the highly selective nature of the likelihood function to warrant high precision (sub-pixel) tracking. However, we need to introduce some new strategies to improve the robustness and efficiency of target tracking.

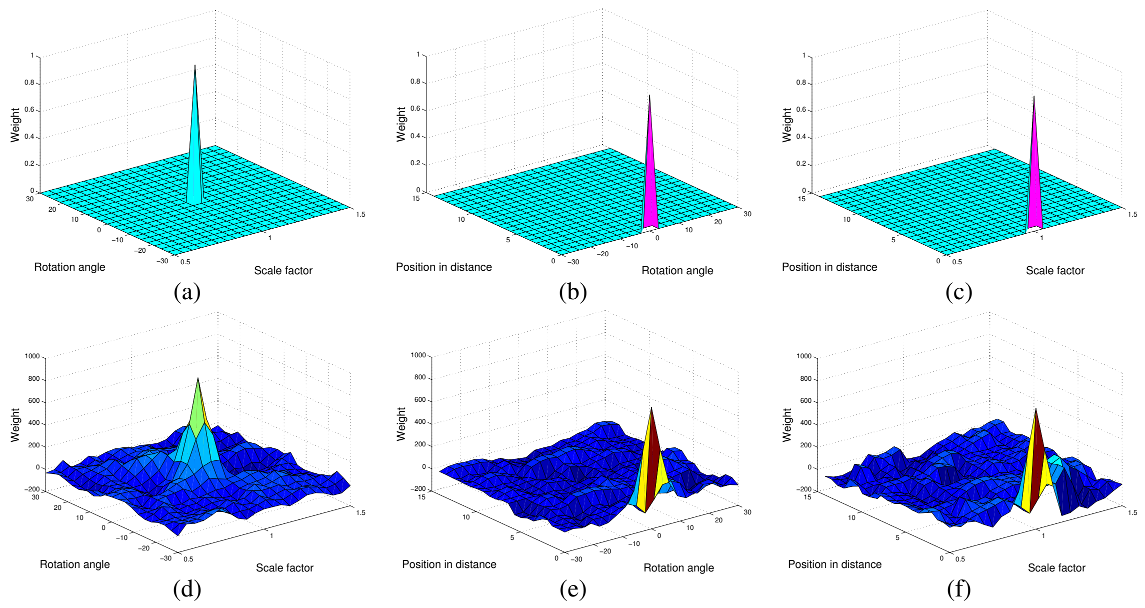

Figure 1. The likelihood function in (8) with respect to rotation-scaling (a), position-rotation (b) and position-scaling. The response of the data term defined in (10) with respect to rotation-scaling (d), position-rotation (e) and position-scaling (f).

## 3. Boosted and Adaptive Particle Filters

### 3.1. Particle Filters for Tracking

We will briefly review the particle filtering theory and details can be found in [1]. From the Bayesian perspective, the tracking problem is to recursively calculate some degree of belief in the state $\mathbf{x}_k$ at time $k$, given the observation $\mathbf{z}_{1:k}$ up to time $k$. Thus, it is required to construct $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ which may be obtained in two stages. With the Markov assumption, the prediction stage uses the system model to obtain the prior PDF of the state at time $k$ as

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})d\mathbf{x}_{k-1}, \tag{11}$$

where $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is given in the system model (1). Then the update stage modifies the prior via Bayes' rule based on the new observation $\mathbf{z}_k$:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) \propto p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1}), \tag{12}$$

where $p(\mathbf{z}_k|\mathbf{x}_k)$ is the likelihood function defined by the observation model in (2). The recursive relationship between (11) and (12) form the basis of the optimal solution.

Sequential Importance Resampling (SIR) is the basic particle filtering algorithm with *resampling* applied at every time step to reduce the degeneracy problem. Also, the prior $p(\mathbf{x}_k|\mathbf{x}_{k-1}^i)$ is used as the importance density for drawing samples. Since the optimal importance density relies on both the previous state ($\mathbf{x}_{k-1}$) and the present observation ($\mathbf{z}_k$) which is not considered in the SIR, the sampling process is not very effective in SIR. The Auxiliary Particle Filter (APF) involves a two-step sampling so that the present observation is considered during sampling [15]. First, we draw samples from the prior and compute weights $\widehat{w}_k^j$, $j = 1, 2, ..., N_p$ and $N_p$ is the particle number. Then we draw an auxiliary sample indices $k^j$ from $\{1, 2, ..., N_p\}$ with $p(n^j = i) = \widehat{w}_k^i$. This helps us identify those promising particles at time $k-1$ that will propagate with larger weights at time $k$. This information is stored in the auxiliary index table $n^j$, based on which we will re-draw samples by propagating those "promising" particles more times.

Unfortunately, we found that both SIR and APF algorithms cannot provide satisfactory results for the tracking problem in (5) due to two possible reasons: (1) The highly selective likelihood function requires high computational load in higher-dimensional state estimation; (2) The white noise acceleration model cannot accommodate the velocity drifting problem over time. These motivate us to improve the APF algorithm for this specific tracking problem.

### 3.2. Boosted APF (BAPF) Algorithm

We will extend the idea of "boosting" by introducing a match filter-based detector to facilitate the tracker. Unlike the one in [14] where the detector and the tracker are two independent processes, here we want to encourage positive interaction between them. One the one hand, detector's template needs to be specified by the tracker output. On the other hand, the tracker is able to mix particles generated from both the tracker and the detector for state estimation.

First, we want to monitor the tracking performance over time. Inspired by the error function used in [17], we introduce a tracking indicator that estimates the tracking performance based on particles's data terms and weights,

$$\phi_k = \sum_{j=1}^{N_p} w_k^j \lambda_k^j, \tag{13}$$

where $\lambda_k^j$ and $w_k^j$ are the data term and the weight, respectively, of particle $j$ at time $k$. $\phi_k$ is the mean estimate of the data term for all particles. A large $\phi_k$ indicates that the tracker is working well at time $k$ and vice versa. The reason we use the data term in (10) rather than the likelihood function in (8) is because that the shape of $\lambda(Z_k)$ is more like an ideal likelihood discussed in [10], and the high selectivity of $p(Z_k|\mathbf{x}_k)$ may not be good for a stable and comparable estimation of the tracking performance.

The initialization plays an important role in most particle filtering algorithms, especially when the state space is complicated. To initialize all four state variables (position and aspect) at the first frame, we generate a number of target templates with different rotation and scaling parameters, i.e., $\theta$ and $s$, which are uniformly drawn from $[\theta_{min}, \theta_{max}]$ and $[s_{min}, s_{max}]$ respectively. Given template $\mathbf{G}(s, \theta)$, we can compute a similarity map $\mathbf{M}(s, \theta)$ whose values are assumed to be proportional to the possibility of the presence of the template. We also need to remove the structured clutter by convolving the first frame $\mathbf{z}_1$ with the template of the GMRF model $\mathbf{K} = [0\ -\beta_v\ 0;\ -\beta_h\ 1\ -\beta_h;\ 0\ -\beta_v\ 0]$. $\mathbf{M}(s, \theta)$ is defined as,

$$\mathbf{M}(s, \theta) = [\mathbf{z}_1 * \mathbf{K}] * \mathbf{G}(s, \theta). \tag{14}$$

A set of initial particles are selected from $\mathbf{M}(s, \theta)$ and each particle is assigned with an initial affine model and position in accordance with the specification of $\mathbf{M}(s, \theta)$. Actually, this initialization also plays the role of target detection.

As we mentioned before, we want to boost the tracking performance by incorporating some contribution from the detector if the tracking indicator $\phi_k$ deteriorates at time $k$. That means the target detection process is specified by the tracking output at time $k-1$, i.e., detector's template at time $k$ is constrained by $(\theta_{k-1}^*, s_{k-1}^*)$ and the search neighborhood is located in a small window centered by $(x_{k-1}^*, y_{k-1}^*)$. Thus this operation can dramatically reduce the computational load of the boosting part. According to the dynamics of the target's aspect, we could believe that the target aspect at time $k$ will be within a range near $(\theta_{k-1}^*, s_{k-1}^*)$, i.e.,

$$\theta_k \in [\theta_{k-1}^* - 2\Delta_\theta, \theta_{k-1}^* + 2\Delta_\theta],$$

and

$$s_k \in [s_{k-1}^* - 2\Delta_s, s_{k-1}^* + 2\Delta_s].$$

Then we can uniformly draw samples from above regions to set detector's template that is convolved with a small window centered by $(x_{k-1}^*, y_{k-1}^*)$ in frame $k$, i.e.,

$$(x_k, y_k) \in [x_{k-1}^* \pm \Delta_x, y_{k-1}^* \pm \Delta_y],$$

where $\Delta_x$ and $\Delta_y$ are specified by the target's motion model in (1). Usually, the window size is small and the additional computational load is negligible. A set of boosting particles can be selected from the match filtering results whose velocities are assigned based on the position change from frames $k-1$ to $k$. All boosting particles whose number could be constant or variable, depending on $\phi_k$, are then mixed with tracker's particles for state estimation. The pesudo-code of the boosted APF filter (BAPF) is shown below.

```
1. Initialization:
    For j=1,$\cdots$, $N_p$
      Draw $X_0^j \sim \mathbf{M}(s,\theta)$ and set $w_0^j = 1/N_p$.
    End
2. For k=1,$\cdots$, $t$
      For j=1,$\cdots$, $N_p$
        Draw $\widetilde{\mu}_k^j \sim p(X_k|X_{k-1}^j)$ and
        compute $\widehat{w}_k^j \propto p(y_k|\widetilde{\mu}_k^j) \cdot w_{k-1}^j$
      End
      Normalize such that $\sum_{j=1}^{N_p} \widehat{w}_k^j = 1$.
      For $j = 1, \cdots, N_p$
        Draw $n^j \sim \{1, 2, ..., N_p\}$ such that
        $p(n^j = i) = \widehat{w}_k^i, (i = 1, 2, ..., N_p)$
        Draw $\widetilde{X}_k^j \sim p(X_k|X_{k-1}^{n^j})$
        compute $w_k^j \propto \frac{p(y_k|\widetilde{X}_k^j)}{p(y_k|\widetilde{\mu}_k^{n^j})}$ and $\lambda_k^j$
      End
      Normalize such that $\sum_{j=1}^{N_p} w_k^j = 1$.
      Compute $\phi_k = \sum_{j=1}^{N_p} \lambda_k^j w_k^j$ according to (13).
      If $\phi_k < T_\lambda$(poor tracking),
         ◇ A local detector specified by the previous state
           estimation induces $N_d$ boosting particles;
         For j=1,$\cdots$, $N_p + N_d$
             compute $w_k^j \propto p(y_k|\widetilde{X}_k^j)$
         End
         Normalize such that $\sum_{j=1}^{N_p+N_d} w_k^j = 1$;
         Compute the mean value of state;
         Do re-sampling to keep $N_p$ particles.
      Else (good tracking)
         Compute the mean value of state.
      End
   End
```

Table 1. Pseudo-code of the Boosted APF (BAPF) algorithm.

$\phi_k$ triggers boosting when it is necessary. Threshold $T_\lambda$ can be obtained from the first few frames where the tracking is usually stable and reliable, or it can be pre-defined according to the target template and the sensor noise. It was found that, for BAPF, an appropriate $N_d$ value is usually around $2\% - 5\%$ of $N_p$, the initial particle number.

## 3.3. Adaptive APF (AAPF) Algorithm

We found an interesting phenomenon of affine-invariant target tracking using the standard SIR and APF algorithms, i.e., the tracking performance is usually good at the beginning, and starts to deteriorate as time goes. We hypothesize that it is due to the drifting problem of the white noise acceleration model, i.e., the actual velocity may deviate from the motion model over time. That means the tracker may not propagate particles properly in the prediction stage. We could increase the noise variances of both the position and the velocity that control the particle diversity to accommodate the drifting problem. However, the computational complexity (the particle number) is increased and the tracking accuracy is reduced. Therefore, we want to adjust the system model appropriately. An appearance-adaptive particle filter was proposed in [17] to realize robust object tracking. The key idea is to make the observation model learnable from the new observations. Moreover, the adaptive-velocity motion model with adaptive noise variance and an adaptive particle number are developed. For simplicity and without loss of generality, we here only consider the adjustments of the two noise variances in the motion model (1) as well as the particle number by a constant scaling factor when it is necessary as indicated by $\phi_k$. For AAPF, $s$ can be a variable that is dependent on $\phi_k$ or a constant.

```
1. Initialization: (the same as BAPF)
2. For $k$=1,$\cdots$ ,$t$
     Do the normal APF and compute $\phi_k$ (the same as BAPF)
     If $\phi_k < T_\lambda$ (poor tracking),
        ◇ Set $\hat{N}_p = N_p * s$;
        ◇ Make a set $\hat{X}^l_{k-1}$ of $\hat{N}_p$ particles by making
          $s$ copies of each $X^j_{k-1}$, $(j = 1, 2, ..., N_p)$ and
          assign corresponding weight $W^l_{k-1}$ as $\frac{w^j_{k-1}}{s}$,
          $(l = 1, .., \hat{N}_p)$;
        ◇ Draw $\widehat{\mu}^l_k$ from $p_{adp}(X^l_k|X^l_{k-1})$ where the noise
          variances are increased by $s$ times, and
          assign weights as $\widehat{W}^l_k = p(y_k|\widehat{\mu}^l_k) \cdot W^l_{k-1}$;
        ◇ Normalize weights such that $\sum_{l=1}^{\hat{N}_p} \widehat{W}^l_k = 1$;
        ◇ Using $\widehat{W}^l_k$ to obtain another set of particles $\widehat{X}^l_k$ as in
          the 2'nd step of the APF, using $p_{adp}(X^l_k|X^{n^l}_{k-1})$, and
          assign weights as $\widetilde{W}^l_k \propto \frac{p(y_k|\widehat{X}^l_k)}{p(y_k|\widehat{\mu}^{n^l}_k)}$;
        ◇ Normalize weights such that $\sum_{l=1}^{\hat{N}_p} \widetilde{W}^l_k = 1$;
        ◇ Compute the mean value of state;
        ◇ Reduce the particle set to $N_p$ by choosing the
          highest $N_p$ valued particles as $X^j_k$ and their
          corresponding weights as $w^j_k$, $(j = 1, 2, .., N_p)$;
        ◇ Normalize weights such that $\sum_{j=1}^{N_p} w^j_k = 1$.
     End
   End
```

Table 2. Pseudo-code of the Adaptive APF (AAPF) algorithm.

## 3.4. More Discussions

It is interesting to see what kind of effects are introduced by the two APF algorithms in terms of the balance of particle distribution. Balancing the diversity and the focus of particles is essential to minimize the chance of particles being trapped into local minima, and this issue is more pertinent in the higher dimensional state space. The concept of the effective sample size is introduced to measure the degeneration problem, and it is defined in [1] as

$$N_{eff} = \frac{1}{\sum_{j=1}^{N_p} (w^j_k)^2}. \qquad (15)$$

A relatively small value indicates a diversified sample set where the particle weights have a large variance, and a relative large value means a focused sample size where the particle weights have a small variance. A good tracking performance requires a balanced particle distribution [7], and this could be indicated by a balanced value of $N_{eff}$. When $\phi_k < T_\lambda$, the tracking performance deteriorates, which can be explained as a consequence of an unbalanced particle distribution, i.e., an unbalanced value of $N_{eff}$ (relatively too large or too small). We expect both BAPF and AAPF will have positive effects on the balance of particle distribution, as shown by the comparison of $N_{eff}$ before and after. This fact can be elucidated by the improvement to $\phi_k$ after boosting or adapting, as shown in the following.

## 4. Simulation Results

To test our algorithms, we generated simulated infrared image sequences of 30 frames (sampling rate $\Delta = 0.04$) by adding GMRF noise fields with a real infrared image of $200 \times 200$ pixels. Then the base target template ($15 \times 35$ pixels) is added to the simulated sequence whose centroid moves according to the white noise acceleration model with initial velocity of (2,0.3) (pixel/frame) and whose aspect (rotation and scaling) varies according to the first-order Markov model. The simulated peak target-to-clutter ratio (PTCR) is 5.6dB with poor target visibility. Even for the largest scaling factor, the target is very small compared with the size of the observation. Additionally, the first-order Markov models of time-varying rotation and scaling are parameterized as follows. The rotation angles vary within $[-30°, 30°]$, and $\Delta_\theta = 2°$ with an additional uniform process noise depending on the step size. The scaling factors are changing within $[0.5, 1.5]$, and $\Delta_s = 0.05$ with an additional uniform process noise depending on the step size. The experiments on simulated videos allow us to quantitatively and objectively evaluate the tracking performance for different particle filter implementations under the GMRF noise assumption and motion/template models. Unfortunately, the standard APF and SIR algorithms, such as the ones in [2], are unable to handle this tracking problem.

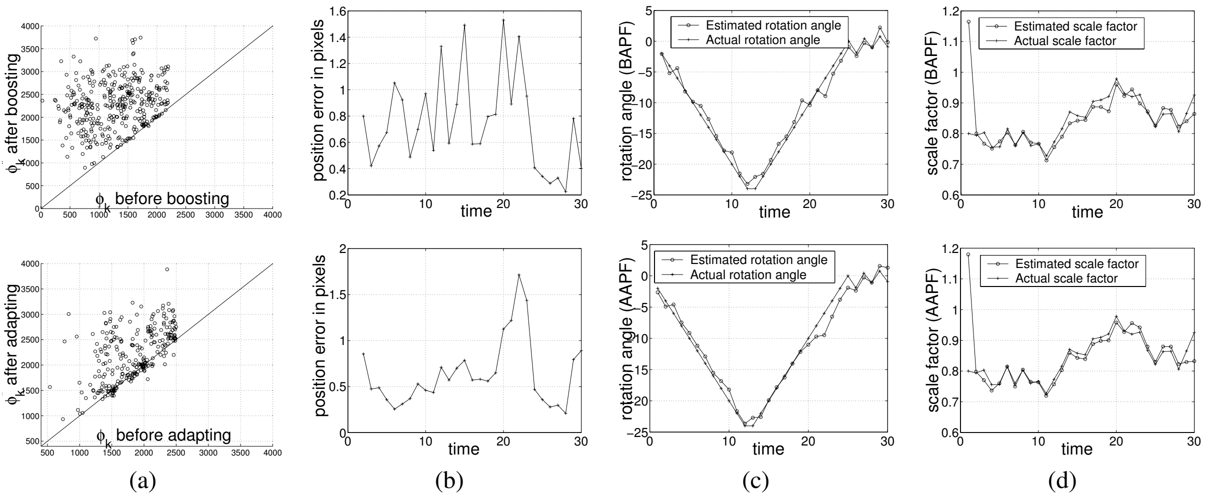

Figure 2. The tracking performance of BAFP (above) and AAFP (below) averaged over 30 Monte Carlo runs. (a) The improvements to the tracking indicator $\phi_k$ defined in (13), (b) BAFP's average tracking error is 0.65 pixel and that of AAFP is 0.64 pixel, (c) comparison of actual rotation angles and estimated ones, and (d) comparison of actual scaling parameters and estimated ones.

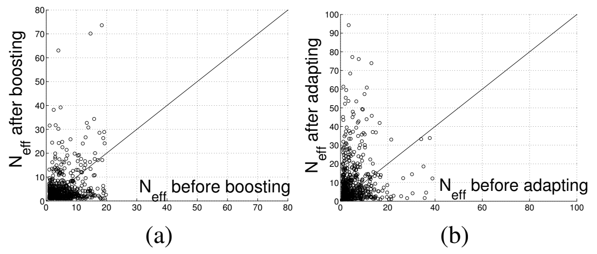

Figure 3. (a) The changes of $N_{eff}$ before and after BAPF (a) The changes of $N_{eff}$ before and after AAPF.

In this work, the BAPF involves 1000 particles from the tracker and additional 20 (2%) particles from the detector when it is necessary, and the AAPF uses either 500 or 5000 particles ($s = 10$). It is expected more intelligent particle number settings can make the two algorithms more efficient, as the one discussed in [17]. Given a pre-defined $T_\lambda$, nearly half of 30 frames involves "boosting" in the BAPF or "adapting" in the AAPF. Therefore, the average particles involved in the BAPF is slightly more than 1000, and that in the AAPF is around 2500. In order to examine the BAPF and the AAPF in terms of their capabilities of adjusting the balance of particle distribution, we compare the values of $N_{eff}$ before and after in Fig. 3. We can see that both BAPF and AAPF can effectively balance the diversity and the focus of particle distribution, which is the key to the good tracking performance. The detailed simulation results of target tracking are shown in Fig. 2 and Fig. 4, where it is shown both algorithms are able to handle the affine-invariant target tracking problem with similar tracking performance, as shown by the improved tracking indicator $\phi_k$. Moreover, the BAPF is more efficient than the AAPF due to the fact that less particles are involved. In addition to the accurate and stable estimation of rotation and scaling parameters, the tracking errors of position are comparable to the one in [2] where a finite set of multi-aspect target templates are involved. However, we are currently unable to handle very low SNRs due to the complicated state space.

## 5. Conclusions

We have discussed a specific multi-aspect target tracking problem, where we addressed the challenges of high-dimensional state estimation with poor target visibility. Specifically, we have re-visited two recent improvements to particle filters, i.e., "boosting" and "adapting". Experimental results show that both the boosted-APF (BAPF) and the adaptive-APF (AAPF) algorithms can handle the affine-invariant tracking problem appropriately by adjusting the balance of particle distribution. Compared with the AAPF, the BAPF is more efficient by involving less particles. This work also provides some experimental tools to further enhance the performance of particle filtering for multi-aspect target detection and tracking in a challenging environment.

## Acknowledgements

The authors would like to acknowledge the anonymous reviewers for their constructive comments that have improved this paper. This work is supported by the Army Research Office (ARO) under Grant W911NF-04-1-0221.

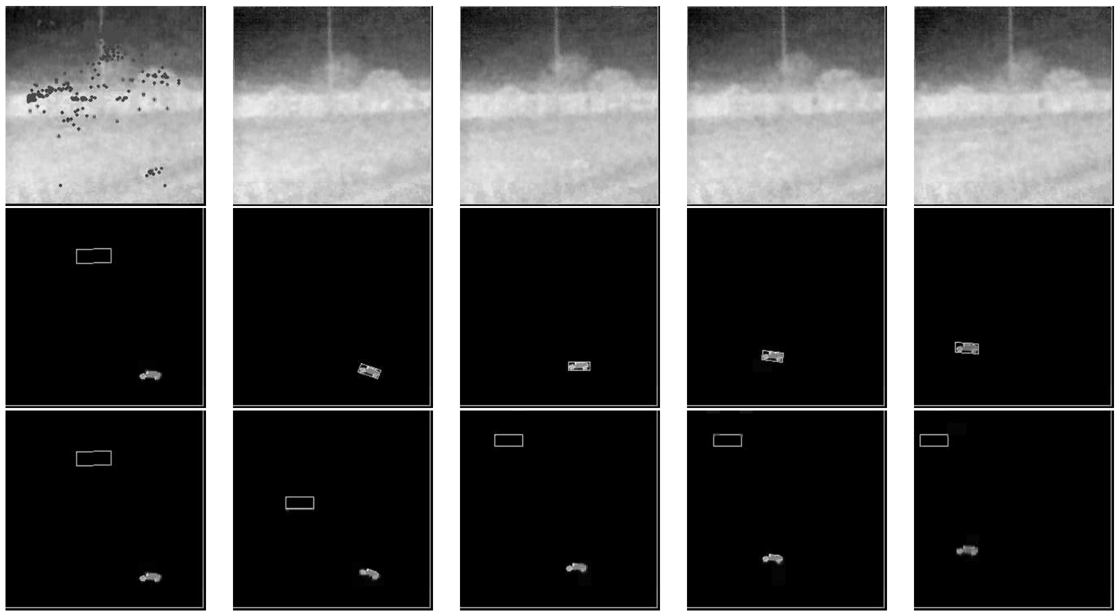

Figure 4. Tracking results (top row: the observed sequence, SNR=5.6dB; middle row: the BAPF tracking results; bottom row: the tracking results without consideration of multi-aspect motion). The dark dots represent the initial particle distribution in the first frame.

## References

[1] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Trans. Signal Processing*, 50(2):174–188, 2002. 4, 6

[2] M. G. S. Bruno. Sequential importance sampling filtering for target tracking in image sequences. *IEEE Signal Processing Letters*, 10(8):246–249, 2003. 1, 2, 3, 6, 7

[3] M. G. S. Bruno. Bayesian methods for multiaspect target tracking in image sequences. *IEEE Trans. Signal Processing*, 52(7):1848–1861, 2004. 2

[4] M. G. S. Bruno and J. M. F. Moura. Multiframe detection/tracking in clutter: optimal performance. *IEEE Trans. Aerosp. Electron. Syst.*, 37(3):925–946, 2001. 1, 2

[5] R. A. C. Chang and A. Khokhar. Multiple object tracking with kernel particle filter. In *IEEE Conf. Comput. Vision Pattern Recognit.*, volume 1, pages 566–573, 2005. 1

[6] J. Deutscher, A. Blake, and I. D. Reid. Articulated body motion capture by annealed particle filtering. In *IEEE Conf. Comput. Vision Pattern Recognit.*, volume 2, pages 126–133, 2000. 1

[7] A. Doucet, J. F. G. Freitas, and N. J. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, Eds. New York, 1st edition, 2001. 1, 6

[8] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings-F (Radar and Signal Processing)*, 140(2):107–113, 1993. 1

[9] B. Han, Y. Zhu, D. Comaniciu, and L. Davis. Kernel-based bayesian filtering for object tracking. In *IEEE Conf. Comput. Vision Pattern Recognit.*, volume 1, pages 227–234, 2005. 1

[10] J. Lichtenauer, M. Reinders, and E. Hendriks. Influence of the observation likelihood function on particle filtering performance in tracking applications. In *Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, volume 1, pages 767–772, 2004. 3, 5

[11] E. Maggio and A. Cavallaro. Hybrid particle filter and mean shift tracker with adaptive transition model. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Philadelphia, PA, March 2005. 1

[12] J. M. F. Moura and N. Balram. Recursive structure of noncousal gauss markov random fields. *IEEE Trans. Inform. Theory*, 38(2):334–354, 1992. 2

[13] J. M. F. Moura and N. Balram. Noncausal Gauss markov random fields:parameter structure and estimation. *IEEE Trans. Inform. Theory*, 39(4):1333–1355, 1993. 2, 3

[14] K. Okuma, A. Taleghani, N. de Freitas, J. J. Little, and D. G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *8th European Conference on Computer Vision*, volume 1, pages 28–39, 2004. 1, 2, 4

[15] M. K. Pitt and N.Shephard. Filtering via simulation: auxiliary particle filters. *J. Amer. Statistic. Assoc.*, 94(446):590–599, 1999. 4

[16] Y. Rathi, N. Vaswani, A. Tannenbaum, and A. Yezzi. Particle filtering for geometric active contours with application to tracking moving and deforming objects. In *IEEE Conf. Comput. Vision Pattern Recognit.*, volume 2, pages 2–9, 2005. 1, 2, 3

[17] S. Zhou, R. Chellappa, and B. Mogghaddam. Adaptive visual tracking and recognition using appearance-adaptive models in particle filters. *IEEE Transactions on Image Processing*, 13(11):1491–1505, 2004. 1, 2, 3, 5, 6, 7