Sequential Particle Generation for Visual Tracking

Yuanwei Lao, *Student Member, IEEE,* Junda Zhu, *Student Member, IEEE,* and Yuan F. Zheng, *Fellow, IEEE*

***Abstract*— A novel probabilistic tracking system is presented, which includes a sequential particle sampler and a fragment-based measurement model. Rather than generating particles independently in a generic particle filter, the correlation between particles is used to improve sampling efficiency, especially when the target moves in an unexpected and abrupt fashion. We propose to update the proposal distribution by dynamically incorporating the most recent measurements and generating particles sequentially, where the contextual confidence of the user on the measurement model is also involved. Besides, the matching template is divided into non-overlapping fragments, and by learning the background information only a subset of the most discriminative target regions are dynamically selected to measure each particle, where the model update is easily embedded to handle fast appearance changes. The two parts are dynamically fused together such that the system is able to capture abrupt motions and produce a better localization of the moving target in an efficient way. With the improved discriminative power, the new algorithm also succeeds in handling partial occlusions and clutter background. Experiments on both synthetic and real-world data verify the effectiveness of the new algorithm and demonstrate its superiority over existing methods.**

***Index Terms*— Haar, low-frame-rate videos, measurement confidence, occlusion, particle filter, proposal distribution, tracking.**

I. Introduction

With the increasing availability and popularity of video cameras, visual tracking is becoming even more important in many applications. Human tracking is used for behavior analysis or event detection in video surveillance, while vehicle tracking plays a significant role in intelligent traffic systems. It is also very useful in human–computer interface, object-based video compression, video motion capture, etc. Meanwhile, visual tracking is found to be a challenging problem due to various reasons, such as rapid nonlinear target motions, clutter background, occlusions, and so on.

According to [1], there are two kinds of approaches to tackle the problem of visual tracking, namely target representation and localization, and filtering and data association. Mean shift [2] is a typical one in the first category, which is efficient and robust in certain scenarios but limited in fast motions and occlusions. Especially, when it comes to multitarget tracking, researchers seem to have less confidence and resort to methods in the second category [3]–[5]. Kalman filter [6] is the classical one in this category, which renders a closed-form and optimal solution for linear Gaussian models. Among the state-of-the-art extensions, particle filter (PF) [7]–[9], or condensation algorithm [10], has achieved popularity due to its capability of handling nonlinear and non-Gaussian models and shown advantages in robustness, occlusion handling, flexibility, etc. Its main power originates from proposing a large number of samples (called *particles*) and making corresponding measurements, and the estimation error could be as small as needed if the number of particles is sufficiently large.

For a good PF-based tracker, the proposal distribution (abbreviated as *PD* or *proposal*, also called *importance density*) and the measurement model are the two key components. Due to convenience and simplicity, the motion model is usually selected as the PD to propagate particles, which implicitly assumes that the particles of the previous frame could be efficiently moved to the next frame. This may not hold for discontinuous motions, such as abruptly moving targets, in low-frame-rate videos, or due to unexpected camera motions. A simple yet common way to remedy this poor temporal coherence is to increase the searching space as well as the number of particles, but the prior information on how large the space should be is not available most of the time. So the filter may have to maintain a large number of particles all the time. Since PF is mainly burdened by the measurement calculation, increasing the number of particles will increase the computation dramatically. On the other hand, the measurement model evaluates each particle on how likely it represents the true target. A strong discriminative power as well as efficiency are highly expected for a robust model, and it should accommodate flexibility to handle appearance changes due to partial occlusions, target in-plane rotation, etc.

We propose an innovative tracking system, including a new sampling method and a fragment-based measurement model. The adoption of the former will greatly enhance the sampling efficiency, especially when the target moves abruptly, while the latter handles partial occlusions. Incorporating measurements into the particle generation, the concepts of detection and tracking are fused together to exploit the temporal coherence and the discriminative ability. In addition, by embedding prior contextual information using a new concept of the measurement confidence, the user is even able to adjust the tradeoff between efficiency and robustness in a flexible way.

Manuscript received April 2, 2008; revised September 12, 2008 and December 19, 2008. First version published May 12, 2009; current version published September 16, 2009. This work was supported by the U.S. National Science Foundation under Grant IIS-0328802, and the National Science Foundation of China under Grant 60632040. This paper was recommended by Associate Editor D. Schonfeld.

Y. Lao and J. Zhu are with the Department of Electrical and Computer Engineering, Ohio State University, Columbus, OH 43210 USA (e-mail: laoy@ece.osu.edu; zhuj@ece.osu.edu).

Y. F. Zheng is with the Department of Electrical and Computer Engineering, Ohio State University, Columbus, OH 43210 USA and also with the Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: zheng@ece.osu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2009.2022797

A. Related Work

Sampling efficiency is a primary issue for a PF-based tracker, since the module of visual tracking could only be assigned with a small portion of the computing resource in a typical real-time system. Researchers have done tremendous work to improve its efficiency as well as the robustness in recent years. One way is to introduce adaptation. A key parameter is the variance of the motion model, which directly determines the region size that particles are going to spread over. Kitagawa [11] laid a theoretical foundation for augmenting the state vector with the variance, which could be estimated simultaneously and is particularly helpful for low-dimensional systems. However, augmentation could easily double the dimensionality, which will increase the number of particles exponentially and thus the complexity significantly. Therefore, Oka *et al*. [12] introduced an adaptive diffusion control for head pose estimation by adjusting the standard deviation of each state element proportional to its change. For a further step, Zhou *et al.* [13] noticed that adjusting only the searching space but not the number of particles might sacrifice the estimation accuracy and proposed to adjust both dynamically according to the status of tracking. Pan and Schonfeld [14] proposed to adjust the proposal variance and the particle number simultaneously over multiple frames by introducing an optimized resource management given an overall computing resource. Thus, the target in a difficult frame will be assigned with more particles, and the overall tracking quality is maximized. Later, they extended a similar idea to multitarget tracking in [15] as well. Combining mean shift with PF is another way. Maggio and Cavallaro [16] presented a hybrid scheme in which every particle is applied with mean shift until it reaches a stable position and becomes more representative for the posterior modes. Cai *et al*. [17] introduced a similar boosted PF into multitarget tracking.

To handle abrupt motions, a detector is normally integrated to reconstruct the poor temporal coherence. Porikli and Tuzel [18] designed an extension to mean shift in low-frame-rate videos by first applying change detection to locate the possible target appearing regions. Bouaynaya and Schonfeld [19] proposed a motion-based PF, where motion detection based on the adaptive block matching is used to generate the proposal. The target motion is limited by the size of the window, which needs user's prior information. More recently, to deal with both the poor temporal coherence and swift appearance variations, Li and Ai [20] adopted a coarse-to-fine methodology and introduced a cascade PF to refine the filtering result in three steps, where each step is equipped with a more complex and discriminative measurement model. An interesting work to iteratively fuse multiple cues by using set theory was proposed by Chang and Ansari in [21]. Even though not for abrupt motions specifically, it also provides a feasible way to incorporate intermediate observations from different cues to refine the tracking result efficiently.

On the other hand, researchers are improving the measurement model for a better tradeoff between the discriminative power and the efficiency. Wang *et al*. [22] proposed to integrate a feature selection into the PF, where only the most discriminative features are selected for the particle search. Similarly, Shakunaga and Noguchi [23] introduced adaptation into a sparse template, where only feature points are used for measurement to outperform the template using all pixels. Otsuka *et al*. [24] used a sparse template in the head tracking for a conversation scene analysis. Rather than feature selection, Chen and Yang [25] introduced the idea of regional confidence, where several regions of target blocks were used to learn the confidence of tracking to provide different discriminative power. Similarly, Yang *et al*. [26] used salient image regions to construct a pool and dynamically selected a portion of it by ranking their discriminative abilities. Avidan [27] treated tracking as a binary classification problem by training an ensemble of classifiers against the background using Adaboost and localized the target with mean shift once the confidence map was obtained. Adam *et al*. [28] directly decomposed the target into fragments and localized the target by combining the vote maps. Among all these works, an efficient and effective way to extract features is the Haar-like wavelet transform since its successful introduction in face detection by Viola and Jones [29]. Besides its simplicity, Haar-like features provide a great flexibility and a sparse way for the feature selection. Since then, it has become very popular and been used widely in detection and tracking, such as [20], [22], [30], [31].

B. Overview of Our Approach

To improve the sampling efficiency and the discriminative power, we introduce a robust tracking system, where our primary contributions are threefold. Firstly, we introduce a novel sampling algorithm, named *sequential particle generation* (SPG), in which particles are proposed sequentially, rather than all at once. Based on the likelihood of the current particle, the proposal is dynamically updated for the next particle, such that particles are sampled to be either more concentrated in the high likelihood area or scattered to capture severe nonlinear motions. This is fundamentally different to the diffusion control methods [12], [13], [15] since no intermediate measurement results were involved previously. In [21], sequential groups of particles are proposed according to different cues, while all particles are generated sequentially here. Without resorting to other methods, the intrinsic resource, i.e., the knowledge of likelihood, is treated as an intermediate detection result and fully exploited to improve the sampling efficiency. Secondly, rather than a template using feature points in [23] and [24], we propose a new yet simple fragment-based measurement model, where the template is divided into non-overlapping regions, where Haar-like features [29] are extracted efficiently. By learning the target and the peripheral background, only a subset of the most distinctive and representative regions is dynamically selected for the observer, which also provides a sparse way to maintain and update the target appearance. Thirdly, we provide a mechanism for the user to integrate contextual confidence on the measurement model into the proposed sampling method for a specific application. This prior information, defined as the *measurement confidence*, helps to determine how the system will employ the intermediate measurement results, either aggressively or conservatively. Compared to the methods based on the generic PF, the proposed algorithm is able to: 1) fully exploit the

likelihood; 2) capture more abrupt motions; 3) handle partial occlusions; and 4) be more efficient in many applications.

The rest of the paper is organized as follows. Section II reviews the generic PF and introduces the proposed sampling algorithm by a theoretical formulation along with a verification on the 1-D case. Section III describes the key components of the whole tracking system, including the motion model and the new measurement model. In Section IV, comprehensive experiments are performed to verify the system and compare the new algorithm with existing methods. Section V discusses several relevant issues, and Section VI concludes the paper.

II. Sequential Particle Generation

This section introduces the new SPG algorithm. First we review the generic PF and its computational limitation to explain the motivation for the new algorithm. Then the concept the measurement confidence is introduced, followed by the presentation of the detailed mathematical formulation. Finally, a detailed illustration on the 1-D case is given for verification of the proposed algorithm.

A. Generic Particle Filter

Let x_k and z_k denote the state vector and the measurement at time k, respectively, and $Z_{1:k} = \{z_1, \ldots, z_k\}$ represent the set of measurements till time k. Under the Bayesian framework, the fundamental problem is to calculate the posterior probability $p(x_k|Z_{1:k})$ given the state transition model $p(x_k|x_{k-1})$ (also called the motion prior) and the measurement model $p(z_k|x_k)$. For solving the problem, the PF [7]–[9] is one of the most successful ways to handle the nonlinear/non-Gaussian models by implementing a Bayesian filtering based on the Monte Carlo method. The idea is to use sufficient number of particles in the state space and their corresponding weights $\{x_k^i, w_k^i\}_{i=1}^N$ to approximate the posterior distribution in a discrete way by $p(x_k|Z_{1:k}) \approx \sum_{i=1}^N w_k^i \delta(x_k - x_k^i)$, where N is the number of particles. A carefully designed *proposal distribution* $q(x_k|x_{k-1}, z_k)$ is used to generate all particles x_k^i for time k, while the associated un-normalized weights $\tilde{w}_k^i$ are calculated iteratively using the following equation based on the factorization of both the posterior and PDs:

$$\tilde{w}_k^i \propto w_{k-1}^i \cdot \frac{p\left(z_k|x_k^i\right) p\left(x_k^i|x_{k-1}^i\right)}{q\left(x_k^i|x_{k-1}^i, z_k\right)} \tag{1}$$

where $p(z_k|x_k^i)$ is the likelihood and $p(x_k^i|x_{k-1}^i)$ is the state transition probability. The sampling efficiency is directly determined by the proposal $q(x_k|x_{k-1}, z_k)$, which should ideally be selected as close as possible to the posterior distribution. As given in [7], the optimal PD is proven to be $p(x_k|x_{k-1}, z_k)$, which is not computationally feasible in most cases, while the most popular choice is the motion model $q(x_k|x_{k-1}, z_k) = p(x_k|x_{k-1})$, due to its simplicity. With this substitution, (1) is reduced to $\tilde{w}_k^i \propto w_{k-1}^i p(z_k|x_k^i)$. To prevent PF from degenerating, a re-sampling technique is usually added at the end of each iteration. Hence the weight calculation is further simplified to $\tilde{w}_k^i = p(z_k|x_k^i)$. Once all weights are obtained, they are normalized to $\{w_k^i\}_{i=1}^N$ according to $w_k^i = \tilde{w}_k^i / (\sum_{i=1}^N \tilde{w}_k^i)$. The estimated state vector therefore is given by $\bar{x}_k = \sum_{i=1}^N w_k^i \cdot x_k^i$. This is normally referred to as the bootstrap filter [8] or the generic PF (GPF).

B. Motivation of the New Approach

As mentioned in [8], the underlying assumption for an effective PF is that the system is evolving slowly and the difference between the consecutive posterior distributions is small, such that the PD is effectively used to explore the state space. However, when the target shows a discontinuous motion, this assumption will not hold. A large number of particles, generated independent identically distributed (i.i.d.) by the same PD, are likely to emerge in low likelihood regions, such that applying GPF to track the target will not be effective. We notice that the measurement model actually behaves like a detector, but in this sense not all the observations made are instantaneously used. Kreucher *et al*. [32] made a similar observation and proposed a particle screening scheme after generating a large number of particles. Unfortunately, the improvement of sampling is at the cost of significantly higher computational cost. Therefore, we propose to incorporate them sequentially into the proposal for particle generation, where the notions of sampling and detection are fused together to adjust the searching space adaptively.

This sequential idea has its root in sampling theory. In [33], Kong *et al*. proposed a method to impute missing data by iteratively using most recent measurements. Chopin [34] incorporated measurements sequentially into the parameter estimation for a higher efficiency. For a PF, a better PD is able to reduce its discrepancy with the posterior distribution, as indicated in [8], [9]. A proposal with the most recent observed information fused is normally preferred, especially when the discrepancy is inevitably large. However, exploiting all the most recent observations is not feasible in visual tracking. Our sequential scheme provides a way to overcome this difficulty, in which observations are incorporated one by one to improve the proposal and reduce the discrepancy. In other words, it bridges the GPF toward the filter with the optimal proposal. The former exploits no measurements, while the latter takes the full advantage of measurements.

C. Measurement Confidence

When a user analyzes a particular application, especially for video surveillance, he/she has to select a measurement model and have a confidence on how discriminative the selected model is going to behave in this particular application environment. Here we define this prior information as the *measurement confidence* (MC). In a relatively simple tracking scenario, where either the background is relatively plain or the target is quite unique in a certain feature space, the model based on that feature could possibly suffice this requirement and the confidence will be high, while in a relatively complex scenario, such as with clutter background, the confidence level could be much lower. Once this prior information is provided by the user, the system could determine how aggressively the intermediate measurement results are utilized. The higher the confidence, the more discriminative the model and the higher the efficiency the tracker could achieve. On the other hand, if the confidence is relatively low, it is more likely to have a multimodal likelihood and the system tends to perform in a conservative way. In this way, the proposed system could

be adjusted to a different tradeoff between efficiency and robustness for a wide range of applications.

D. Mathematical Formulation

This section presents the mathematical implementation of the new idea. A framework based on Bayes' Rule is introduced first, and a paradigm is presented to describe how the most recent observation and the user confidence are fused to update the PD, followed by a verification on the 1-D scenario.

Following the convention in Section II-A, we use x_k and z_k as the random variables for the state and measurement, and let x_k^i and z_k^i be the ith ($i = 1, 2, \ldots, N$) particle and the realized measurement (called likelihood as well) at time k, respectively. The PD for ith particle is $q_i(x_k)$. We denote the upper case X_k^i to be the set of particles from first to ith, and similarly we have Z_k^i. We introduce a framework for the SPG, where estimation results in previous frames are used to initialize the motion model and generate the first particle x_k^1, and the most recent observation is fused into current proposal for the next particle generation. Thus, we obtain

$$x_k^i \sim \begin{cases} p\left(x_k | X_k^0\right), & \text{if } i = 1 \\ p\left(x_k | Z_k^{i-1}, X_k^{i-1}, X_k^0\right), & \text{if } i > 1 \end{cases} \tag{2}$$

where $X_k^0 = \bar{X}_{k-1}$, and $\bar{X}_{k-1}$ denote the estimation results in previous frames. The realization is determined by how the motion is modeled, which will be discussed in Section III-A. When $i > 1$, the conditional distribution can be decomposed in an iterative way by applying the Bayes' Rule

$$p\left(x_k | Z_k^i, X_k^i, X_k^0\right) \tag{3}$$

$$= p\left(x_k | z_k^i, x_k^i, Z_k^{i-1}, X_k^{i-1}, X_k^0\right) \tag{4}$$

$$= \frac{p\left(x_k | Z_k^{i-1}, X_k^{i-1}, X_k^0\right)\, p\left(z_k^i, x_k^i | x_k, Z_k^{i-1}, X_k^{i-1}, X_k^0\right)}{p\left(z_k^i, x_k^i | Z_k^{i-1}, X_k^{i-1}, X_k^0\right)} \tag{5}$$

$$\propto p\left(x_k | Z_k^{i-1}, X_k^{i-1}, X_k^0\right)\, p\left(z_k^i, x_k^i | x_k\right) \tag{6}$$

where x_k is the only random variable here, and the denominator in (5) is a constant and can then be neglected in (6). $p(z_k^i, x_k^i | x_k)$ is the simplified from the corresponding term in (5) by applying the property of conditional independence. This provides a framework to iteratively update the PD by fusing the most recent observation.

1) An Instantiation: We denote $\beta \in [0, 1]$ as the confidence index predefined by the user to reflect the contextual confidence on the measurement model. Then we define an *updating distribution* $p_u(z_k^i, x_k^i | x_k)$ associated with the current particle x_k^i, to incorporate the most recent observation information. In order to introduce adaptiveness to the searching range of the proposal, a parameter λ_i is also defined to represent the relationship between the particle and the current proposal. Given the current proposal $q_i(x_k)$, and analogous to (3) by $p(z_k^i, x_k^i | x_k) \propto p_u(z_k^i, x_k^i | x_k)^\beta$ and $p(x_k | Z_k^{i-1}, X_k^{i-1}, X_k^0) \propto q_i(x_k)^{\lambda_i}$, we have the iterative PD updating equation as

$$q_{i+1}(x_k) \propto q_i(x_k)^{\lambda_i} \cdot p_u\left(z_k^i, x_k^i | x_k\right)^\beta \tag{7}$$

where $q_{i+1}(x_k)$ is the updated PD to generate the next particle x_k^{i+1}, and $p_u(\cdot)$ is generated by the observation result z_k^i and imposed on the current particle x_k^i. The definition of λ_i will be given shortly, and its physical meaning will become clearer in Section II-E, where we verify the whole scheme using the 1-D scenario. Given the generated particle and the current PD, two pieces of information, i.e., their relationship and the likelihood, are fused with the MC to update the PD for the next particle. As we can see in (7), the term $p_u(z_k^i, x_k^i | x_k)$ represents the new observation information, while the previous proposal, $q_i(x_k)^{\lambda_i}$, behaves as a prior term of fusing up to the $(i-1)^{th}$ observation. The application of Bayes' rule generates the new proposal by integrating the most recent observation. Thus, each intermediate PD is viewed as both an approximate posterior for the current iteration and a prior for the next round. The overall iteration can thus be considered as a series of posteriors to approach the true underlying posterior distribution.

2) A Gaussian Realization: According to the current observation, there are various ways to generate the updating distribution. To make the sequential scheme effective yet efficient, we select the imposed updating distribution $p_u(z_k^i, x_k^i | x_k)$ to have a form of the multivariate Gaussian as follows:

$$p(x | \mu, \Sigma) = \frac{(2\pi)^{-d/2}}{|\Sigma|^{1/2}} exp\left[-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right]. \tag{8}$$

It is important that this updating distribution should not be confused with the observation distribution $p(z|x)$ defined by the measurement model. The utilization of the Gaussian form for the updating distribution is only applied in the proposal updating step, which does not affect its capability of handling non-Gaussian and nonlinear models as the GPF. Specifically, we have $p_u(z_k^i, x_k^i | x_k) = N(x_k^i, \Sigma_u)$, where the covariance matrix Σ_u is determined by the observation result. First, the likelihood value z_k^i of particle x_k^i is obtained by the measurement model $p(z_k | x_k^i)$, and then we endue the peak value of a multivariate Gaussian with the likelihood value according to (8), i.e.,

$$z_k^i = (2\pi)^{-d/2} |\Sigma_u|^{-1/2} exp(0) = (2\pi)^{-d/2} |\Sigma_u|^{-1/2}. \tag{9}$$

In the 2-D case for visual tracking, we solve $z_k^i = (2\pi)^{-1} \cdot |\Sigma_u|^{-1/2}$ for the covariance matrix Σ_u and obtain $|\Sigma_u| = (2\pi z_k^i)^{-2}$. We therefore select

$$\Sigma_u = \begin{pmatrix} (2\pi z_k^i)^{-1} & 0 \\ 0 & (2\pi z_k^i)^{-1} \end{pmatrix}. \tag{10}$$

The advantage of having both the updating and PDs in the form of multivariate Gaussian is to apply the *product enclosure* property. More explicitly, the multiplication of two Gaussian distributions is still an un-normalized Gaussian, which holds even when there is an exponent in the distribution. Since the Gaussian is uniquely determined by its mean and covariance, the iteration of (7) could thus be quickly calculated. Let $q_i(x_k) = N(\mu_i, \Sigma_i)$, plus $p_u(z_k^i, x_k^i | x_k) = N(x_k^i, \Sigma_u)$. Then the updated proposal after the normalization is given by $q_{i+1}(x_k) = N(\mu_{i+1}, \Sigma_{i+1})$ according to (7), where

$$\begin{cases} \Sigma_{i+1} = \left(\lambda_i \Sigma_i^{-1} + \beta\, \Sigma_u^{-1}\right)^{-1} \\ \mu_{i+1} = \Sigma_{i+1}\left(\lambda_i \Sigma_i^{-1} \mu_i + \beta\, \Sigma_u^{-1} x_k^i\right). \end{cases} \tag{11}$$

3) Selection of λ_i: Besides β, another key parameter in (7) is the relation parameter λ_i. Since all the PDs are in the form of Gaussian, we only have to consider the distance between the particle x_k^i and the proposal mean μ_i. The basic idea is to utilize λ_i to adjust the dynamic searching range of the PD, especially when the likelihood is low. The case that the particle is close to the PD mean should be differentiated from the case that it is far away. In the former one, it implies that the possibility that the posterior mode exists around the center of the current PD should be reduced, and therefore the searching space should be enlarged to increase the chance of capturing the posterior mode. The closer the particle is to the mean, the less likely the posterior mode exists in that area and the larger the searching space should be enlarged to. When the particle x_k^i is far away from the PD mean μ_i, it provides no extra information, and the searching range should be kept approximately the same as the previous one. Therefore, we select

$$\lambda_i = 1 + \epsilon - exp\left(-\alpha||x_k^i - \mu_i||^2\right) \tag{12}$$

where $\epsilon \geq 0$ is a small positive quantity protecting λ_i from being zero. $\alpha \geq 0$ is the parameter adjusting the convergence speed of λ_i and determines the amplitude of variation of the searching space. The smaller the α, the smaller the λ_i and the larger the space the updated PD exploits. The key is when x_k^i is close to μ_i, $\lambda_i \to 0$ since $\alpha \geq 0$; otherwise $\lambda_i \to 1$. As shown in the experiments, $\alpha \sim 1$ suffices most cases. How λ_i affects the scheme will be further discussed in Section II-E.

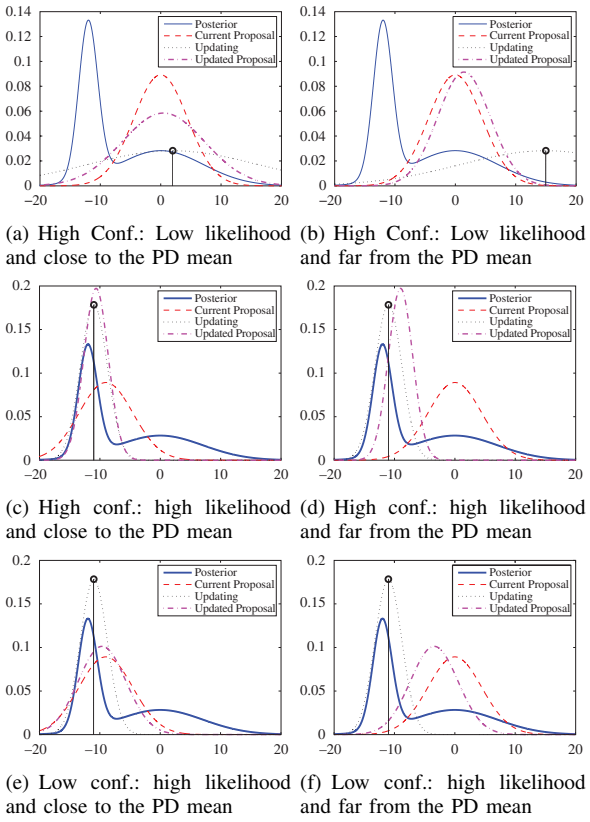
E. Verification on 1-D Example

To verify the proposed method as well as to have a visualized understanding, we simplify the proposed scheme by reducing (11) into the 1-D case and obtain

$$\frac{1}{\sigma_{i+1}^2} = \lambda_i \frac{1}{\sigma_i^2} + \beta \frac{1}{\sigma_u^2} \quad \text{and} \quad \frac{\mu_{i+1}}{\sigma_{i+1}^2} = \lambda_i \frac{\mu_i}{\sigma_i^2} + \beta \frac{x_k^i}{\sigma_u^2}. \tag{13}$$

Then we present six typical scenarios to verify the proposal updating scheme in the 1-D case and show how the measurement result and the user confidence are involved. As shown in Fig. 1, the solid blue curve represents the underlying posterior distribution, while the dashed red Gaussian denotes the current PD, which generates the current particle (denoted as the x-axis of the black circle). The updating distribution is represented by the black dotted curve imposing onto this particle in terms of its likelihood value. According to (13), the particle location with respect to the current PD and the updating distribution are integrated into the current PD to generate the updated PD (plotted as the dash-dotted magenta curve).

First, let us consider the situation where the measurement confidence is very high, $\beta = 1$. Corresponding to Fig. 1(a) and (b), if the current particle yields a low likelihood z_k^i (the y-axis coordinate of the black circle), the variance of the updating distribution σ_u^2 is very large according to (10), and its inverse $1/\sigma_u^2$ becomes very small. The case that the particle is close to the PD mean should be differentiated from the case that it is far away. For the former case in Fig. 1(a), the particle, appearing in the high-probability area according to the current PD, yields a low likelihood. Thus, the possibility



(a) High Conf.: Low likelihood and close to the PD mean
(b) High Conf.: Low likelihood and far from the PD mean
(c) High conf.: high likelihood and close to the PD mean
(d) High conf.: high likelihood and far from the PD mean
(e) Low conf.: high likelihood and close to the PD mean
(f) Low conf.: high likelihood and far from the PD mean

Fig. 1. Six typical scenarios for proposal updating: (a) the variance increases; (b) the previous PD almost remains; (c) and (d) the updated proposal is dragged toward the current particle when the confidence is high. (e) and (f) the updated proposal stays back to the current PD due to the low confidence even though the likelihood is high.

that the posterior mode exists around the current PD mean should be reduced. A wise way is to enlarge the searching space to increase the chance of capturing the posterior mode, which is mathematically equivalent to amplifying the variance of the updated PD (dash-dotted). At this moment, how far the updated PD is dragged toward the updating distribution is determined by the distance between x_k^i and the mean μ_i. The smaller the distance, the more λ_i approaches zero, and σ_{i+1}^2 is more determined by the term β/σ_u^2 and becomes smaller. Physically, the less likely the posterior mode exists in that area and the more the variance should be enlarged to. In the latter case shown in Fig. 1(b), since the particle is far away from the PD mean, it provides no extra information, and the best guess is still the current PD or its approximate. So the updated PD almost coincides with the previous one. Mathematically, $\lambda_i \to 1$, and $1/\sigma_{i+1}^2$ is dominated by the term λ_i/σ_i^2. Thus σ_{i+1}^2 is close to or almost the same as σ_i^2.

On the other hand, when the likelihood is high, σ_u^2 becomes really small and is likely to dominate σ_{i+1}^2 with $\beta = 1$. Especially, when the particle is close to the PD mean in Fig. 1(c), $\lambda_i \to 0$, and the dominance becomes more obvious. In Fig. 1(d), when $\lambda_i \to 1$, the updated PD still leans toward the updating distribution due to the relatively large variance of the current PD. Physically, in both cases, it implies that the posterior mode is likely to be around the current particle, especially when the measurement model is highly

discriminative. Therefore, the updated PD is dragged toward the current particle in both cases with a reduced variance. The higher the likelihood, the smaller the variance.

Second, if the confidence is rather low (let $\beta = 0.1$), the probability of a local maxima due to clutter background at the position with a high likelihood is higher, and the term β/σ_u^2 will be less significant in (13). Therefore a safer choice would be a less aggressive movement of the updated proposal from the current PD, as shown in Fig. 1(e) and (f). Compared with their counterparts in the second row of Fig. 1, the updated PD tends to stay back to the current proposal somehow and is less leaning toward the updating distribution. By preserving more properties from the initial motion prior, the scheme behaves in a more conservative way in a complex background. One extreme case is that the updated and current PD almost coincide if the confidence is extremely low, which basically reduces the proposed scheme to GPF.

F. Summary of SPG

SPG has two significant parameters, the MC β and the adjusting parameter α. Basically, β determines how aggressively the measurement is used in the proposal updating, while α adjusts the significance of the distance between the particle and the proposal mean and controls how large the search range should be enlarged to. An extreme case is when $\beta = 0$ and $\alpha = +\infty$, then $\lambda = 1$ and $\Sigma_{i+1}^2 = \Sigma_i^2$, $\mu_{i+1} = \mu_i$, which is almost equivalent to the GPF. In summary, SPG presents a new way to integrate the most recent observation information into the particle generation by updating a series of PDs, and a mechanism is provided for to equip the updating scheme with the contextual confidence, which is an index of the tradeoff between robustness and efficiency. For each new frame, the first PD is initialized with the motion prior $q_1(x_k) = p_m(x_k|\bar{X}_{k-1})$, which will be given in Section III-A. The first particle is thus obtained by sampling $x_k^1 \sim q_1(x_k)$, and its likelihood is used to generate the updating distribution $p_u(z_k^1, x_k^1|x_k)$. With an exponent, $q_1(x_k)^{\lambda_1}$ could still be considered as an un-normalized prior probability, so is $p_u(\cdot)^\beta$. Thus, similar to (4)–(6), we have the updated PD as

$$q_2(x_k) \propto q_1(x_k)^{\lambda_1} \cdot p_u\left(z_k^1, x_k^1|x_k\right)^\beta \propto p\left(x_k|X_k^0\right) \cdot p\left(z_k^1, x_k^1|x_k\right) \propto p\left(x_k|z_k^1, x_k^1, X_k^0\right).$$

Similarly, we have the second particle $x_k^2 \sim q_2(x_k)$ and its corresponding updating distribution $p_u(z_k^2, x_k^2|x_k)$. In this way, a series of PDs could be obtained iteratively by

$$q_i(x_k) \propto q_{i-1}(x_k)^{\lambda_i} \cdot p_u\left(z_k^{i-1}, x_k^{i-1}|x_k\right)^\beta \propto p\left(x_k|Z_k^{i-1}, X_k^{i-1}, X_k^0\right).$$

In this way, particle generation could be summarized as in (2). For each particle, we have the corresponding weight $\tilde{w}_k^i = p(z_k|x_k^i)$, and its normalized version is obtained through $w_k^i = \tilde{w}_k^i/(\sum_{i=1}^N \tilde{w}_k^i)$. Once we collect $\{x_k^i, w_k^i\}_{i=1}^N$, we can perform the same estimation as the GPF, $\bar{x}_k = \sum_{i=1}^N w_k^i \cdot x_k^i$.

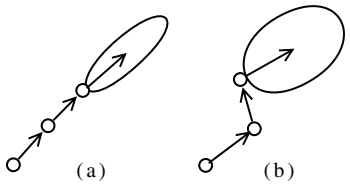


Fig. 2. Motion prior initialization.

III. Proposed Tracking Algorithm

Besides the proposed SPG, several important components are introduced in this section to complete the design of the whole tracking system, including the motion prior initialization and the fragment-based measurement model.

A. Motion Prior Initialization

Previously, the most frequently used motion models are the ones with isotropic Gaussian distributions, which are not efficient in many cases. As shown in Fig. 2(a), when the target motion is relatively linear, a polarized Gaussian will greatly enhance the sampling efficiency by constraining the particles to spread along the moving direction. In the case of nonlinear target motions in Fig. 2(b), a more isotropic one is preferred to accommodate more flexibility. Therefore we propose an adaptive anisotropic Gaussian based on the motion pattern of the target, which is quite analogous to the inverse procedure of the principal component analysis [35].

Similar to [5], a second-order auto-regressive model is employed here: $x_k = \bar{x}_{k-1} + \bar{v}_{k-1} + u_k$, where $\bar{x}_{k-1}$ and $\bar{v}_{k-1}$ are the estimated state and speed at time $k-1$, respectively, and u_k is the motion transition noise, normally following a zero-mean Gaussian $N(0, \Sigma_k)$. First, given the state estimations from previous frames, we calculate the estimated speed $\bar{v}_{k-1} = \bar{x}_{k-1} - \bar{x}_{k-2}$. To obtain an estimate of Σ_k, we use $\bar{v}_{k-1}$ as the major eigenvector, and let its normalized version be $V_1 = [\tau_1\ \tau_2]^T$. Then the second eigenvector should satisfy $V_1^T V_2 = 0$, by which we obtain $V_2 = [\tau_2\ -\tau_1]^T$. Based on the average displacement of the previous few frames, we have an estimate of the square magnitude of the target speed as $\rho_k = (|\bar{x}_{k-1} - \bar{x}_{k-2}|^2 + |\bar{x}_{k-2} - \bar{x}_{k-3}|^2)/2$. Then along the major axis, we have the first eigenvalue $\lambda_1 = \rho_k$, while on the minor axis we assign $\lambda_2 = \gamma\rho_k$, where $\gamma \in [0, 1]$ adjusts the tradeoff between efficiency and nonlinearity. When $\gamma \to 1$, the method is the least efficient and accommodates most nonlinearity. When γ is close to zero, it is more efficient while considering less nonlinearity. Once we have $D = diag(\lambda_1, \lambda_2)$ and $V = [V_1\ V_2]$, the covariance is given by $\Sigma_k = VDV^{-1}$. Thus we obtain the constant-speed model

$$p_m(x_k|\bar{X}_{k-1}) = N((\bar{x}_{k-1} + \bar{v}_{k-1}), \Sigma_k). \tag{14}$$

This adaptive covariance can also be considered as a form of diffusion control like [12], [13], [15]. However, this is only an initialization for the proposal distribution in each new frame, and particles are not generated from this initial version all at once but sequentially from the most recently updated PDs.

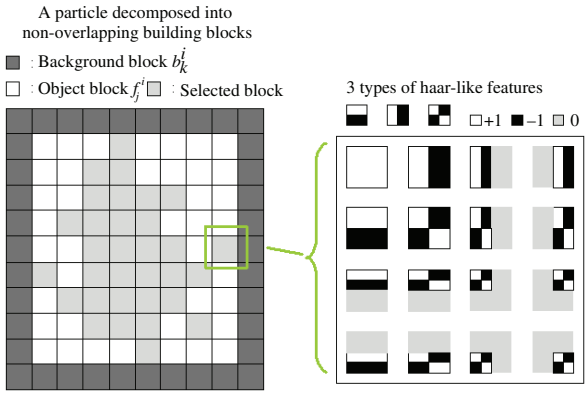


Fig. 3. Fragment-based likelihood and Haar feature extraction.

B. Fragment-Based Likelihood Model

Similar to GPF, a key issue of the proposed algorithm is to define the measurement model $p(z|x)$ given the current state x. Once a particle is proposed by the PD, it has to be evaluated as to how it represents the real target. Among the appearance-based models used in tracking, such as color, edge, texture, contour, etc., color histogram is the simplest and has been widely used in many previous works, such as [1], [17], [36]. In simple scenarios, the color histogram suffices the needs, and the user has a high confidence on it. As the task becomes more difficult, the confidence will inevitably drop. Especially for complex scenarios, such as in cluttered background or with occlusions, it requires the model to provide a stronger discriminative power against the background as well as a better target representation. Without utilizing any spatial information, color histogram has severe limitation in these scenarios.

Following the decomposition idea in [37], we propose a fragment-based measurement model, where the matching template is divided into non-overlapping *building blocks* (BB) according to its center location, as shown on the left of Fig. 3. The template is initialized in the first frame by extracting a feature vector with great discriminative ability for each block and then updated in each successive frame after the state estimation. It is represented by a set of template feature vectors, $g_{k,j}$ $(j = 1, \ldots, N_{BB})$, where N_{BB} is the total number of the foreground (target) building blocks. During the measurement, each particle is also decomposed into corresponding blocks, and a set of feature vectors $f_{k,j}^i$ $(j = 1, \ldots, N_{BB})$ are extracted to represent the ith particle x_k^i. Each particle BB is compared with the corresponding template one by norm-2, and all the local measurements are fused together to give the overall evaluation, which is given by a weighted summation as

$$E_k^i = \sum_{j=1}^{N_{BB}} s_j \eta_{k,j} ||f_{k,j}^i - g_{k,j}||^2 \tag{15}$$

where the weight s_j denotes the spatial factor and $\eta_{k,j}$ is the selection indicator. s_j assigns larger weights to the central blocks and smaller weights to the peripheral ones. A typical choice is the Epanechnikov profile, as mentioned in [1]. $\eta_{k,j}$ is a binary indicator for each block and determined by the template block selection in Section III-B.2 since only a certain portion of building blocks is selected in the measurement depending on the need of applications. Then the likelihood function is given by a form of Gibbs distribution as

$$p\left(z_k|x_k^i\right) = exp\left(-E_k^i/\sigma_E^2\right) \tag{16}$$

where σ_E^2 is the evaluation variance. This region-based decomposition provides an example of using a sparse template to enhance its both discriminative and representative power by exploiting the adaptation of the target, where a feature-point-based template, such as [23] and [24], could also be employed. Even in many difficult scenarios with clutter background, the user could still have a fairly high measurement confidence, which helps the proposed scheme utilize the intermediate measurement in a more aggressive and efficient way. The detailed method of feature extraction as well as the template block selection and update are introduced below.

1) Haar Feature Extraction: Since Viola and Jones [29] demonstrated the surprising performance in face detection by using the over-complete Haar wavelet-like features in a boosted structure, the method has been used extensively in object tracking and detection, such as [20], [22], [28], and [30], [31] due to its computational efficiency and the ability to capture local information. Similar to the scheme in [30], we propose to use three kinds (up-down, left-right, and diagonal) of typical Haar-like features in different resolutions for each block, as shown on the right of Fig. 3. Pixels in the white are associated with weight $+1$, while those in the dark are assigned with -1. The gray pixels are associated with 0 and thus not used. Therefore, for the upper left feature, the summation of all pixel values is calculated, while for the rest features the difference between the summation in the white and that in the dark is calculated. In this way, with two-level decomposition, we obtain 16 feature values per channel for each block, as shown in Fig. 3. Hence, we have $g_{k,j}, f_{k,j}^i \in \mathcal{R}^{48}$ for chromatic (RGB) images and $g_{k,j}, f_{k,j}^i \in \mathcal{R}^{16}$ for gray-scale images. As described in [38], the advantage of Haar-like features is their great efficiency once the integral image obtained from the original image by $ii(x, y) = \sum_{x' \le x, y' \le y} i(x', y')$, which could be done in an iterative way. Based on the integral image, every summation of pixels in a certain rectangle area could be obtained by three simple additions or subtractions.

2) Template Block Selection: Due to target irregular shape or partial occlusions, the power of the foreground blocks representation varies, and their discriminative abilities against the background are different as well. For example, some peripheral foreground blocks may contain mostly background information, while in the case of occlusion, only a certain (even small) portion of blocks is capable of rendering useful target information. Therefore, how to select both representative and discriminative blocks plays a significant role in a robust tracker. In Fig. 3, we denote peripheral blocks outside the foreground as the background blocks, and the corresponding feature vectors are $b_{k,l}(l = 1, \ldots, N_{BK})$ for the kth frame. Then for a given template block with the feature vector $g_{k,j}$,

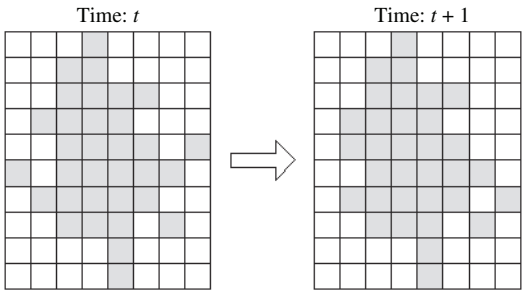


Fig. 4. Selected blocks evolving over frames.

we traverse the background blocks, $b_{k,l}$, and find the smallest difference h_j by norm-2, i.e.,

$$h_{k,j} = \min_l ||g_{k,j} - b_{k,l}||^2 \tag{17}$$

where $h_{k,j}$ is a measure of the similarity between the template block $g_{k,j}$ and the background. Afterwards, we sort all the difference values $h_{k,j} (j = 1, \ldots, N_{BB})$, namely the similarities to the background, in a descending order, and the template blocks corresponding to the top P percentage (i.e., the most discriminative-against-background blocks) are therefore selected and assigned with $\eta_{k,j} = 1$ (otherwise $\eta_{k,j} = 0$), as shown (the gray blocks) in Fig. 3. In this way, only the most discriminative blocks account for the measurement calculation in (15). Due to the appearance change or background variations, selected blocks are expected to evolve over frames, as shown in Fig. 4. For each new frame, the track uses the selection template from the previous frame for the measurement of each newly generated particle. After an estimation is reached for the current target location, this template block selection procedure is performed again on the estimated state, and the newly selected blocks are used in obtaining the measurements in the next frame.

The block size and the percentage P are the two key parameters determining the computational complexity, which increases as the block size decreases and/or P increases. The number of foreground blocks N_{BB} is determined by the target size and the block size, and that number for the background satisfies $\lceil 4\sqrt{N_{BB}} + 4 \rceil \leq N_{BK} \leq 2N_{BB} + 6$, where $\lceil \cdot \rceil$ is the ceiling of the input. Since the selection involves sorting of all the background blocks for each foreground block and another overall sorting for all the foreground blocks, the computational cost is at the scale of $O(N_{BB}N_{BK}^2 + N_{BB}^2)$ if we use the simplest $O(n^2)$ sort algorithm, such as the bubble sort. In many typical applications, where the frame size is 320×240, 20×60 pixels for a median-size human target and a block size of 8×8 pixels (used in all experiments), the computational cost is affordable since we have $N_{BB} = 24$ and $N_{BK} = 26$, and it is performed only once per frame. Besides, the user has the flexibility to tune both the block size and the selection percentage. A smaller P could significantly reduce the sorting cost in the measurement calculation, and P is empirically selected as 0.7 in the experiments.

3) Adaptive Template Update: When the target has abrupt motions, in-plane rotation, illumination change, etc., the same target may have a poor appearance continuity. This requires the target template to be updated periodically and effectively, which is another key issue. Block selection has helped the update partially since different blocks could be selected for the model. Here we take the advantage of a simple linear model to modify the target template, similar to the one in [39], which updates the color histogram. Once we obtain the location estimation in every frame, besides the template block selection, we extract feature vectors for the blocks around the estimated state and obtain $r_{k,j}, (j = 1, \ldots, N_{BB})$. Thus, the target template is updated for the next frame blockwise as

$$g_{k+1,j} = \xi \cdot g_{k,j} + (1 - \xi) \cdot r_{k,j} \tag{18}$$

where $\xi \in (0, 1)$ is the adjusting parameter.

TABLE I
PSEUDO-CODE OF THE SPG TRACKING ALGORITHM

Algorithm I: SPG(k, N, $\bar{X}_{k-1}$)
1) Step 1: Initialize the motion prior by (14) and the PD: $q_1(x_k) = p_m(x_k|\bar{X}_{k-1})$
2) Step 2: Sequential particle generation
   for $i = 1$ **to N**
   a) $x_k^i \sim q_i(\cdot)$: Generate the current particle
   b) z_k^i: Complete the likelihood calculation (Section III-B)
   c) $p_u(\cdot)$: Generate the updating distribution by (9)
   d) $q_{i+1}(\cdot)$: Update the proposal distribution using (11)
   end
3) Step 3: Estimate the current State: $\bar{x}_k = \sum_{i=1}^N w_k^i \cdot x_k^i$
4) Step 4: Block selection for the next frame (Section III-B.2)
5) Step 5: Template update for the next frame (Section III-B.3)

return ($\bar{x}_k$)

C. Summary of the Tracking Algorithm

The pseudocode of this algorithm is in Table I.

IV. NUMERICAL RESULTS

We carry out a series of experiments to test the proposed tracking system in a step-by-step way. We manually define the target by a rectangle in the first frame, where the reference template is immediately obtained. To show how abruptly the target is moving, solid and dashed rectangles are used to mark the target locations in the current and previous frames, respectively. Firstly, we focus on the novel sampling algorithm, and adopt the color histogram (a joint RGB histogram with 8 bins per channel) as the appearance model. With a high MC in the randomly generated synthetic and several real-world applications, the system is able to demonstrate its capability of either capturing abrupt motions or increasing sampling efficiency. When the scenario becomes more complex, the confidence drops until it is no longer suitable. Then we utilize the new fragment-based model and show its discriminative power and adaptability in handling partial or heavy occlusions in several challenging sequences. An important note of the implementation is that we actually select a small value (0.0001) as the likelihood threshold for z_k^i to avoid numerical problems in the matrix inversion during the MATLAB simulation. When the likelihood value z_k^i in (9) is less than this threshold, the step of updating the proposal, (11), is reduced to $\Sigma_{i+1} = \Sigma_i / \lambda_i$. In other words, when the likelihood is too

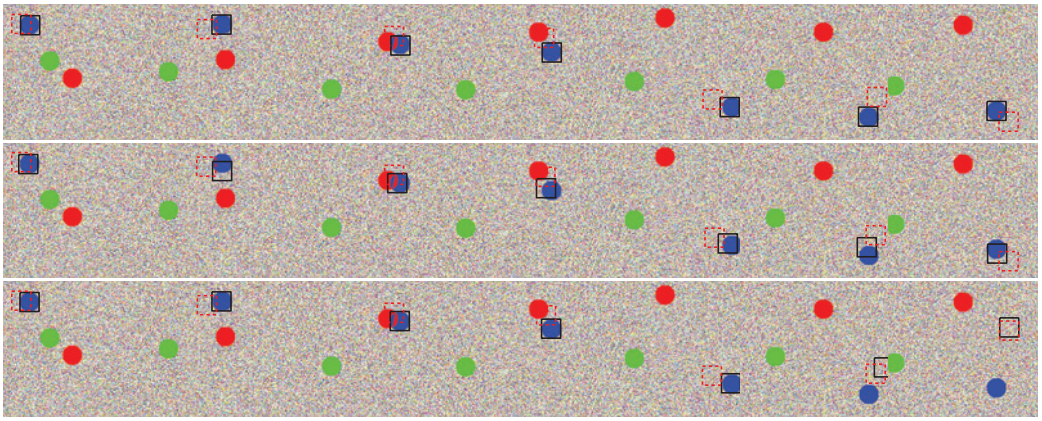


Fig. 5. Tracking results on the *Synthetic* sequence for Frames 4, 20, 28, 36, 64, 76, and 98 (with color histogram). The blue sphere is the target of interest, while the red and green spheres are interference. To show the target motion pattern, solid black and dashed red rectangle denote the estimation results in the current and previous frames, respectively. First row: SPG; second row: GPF; third row: mean shift. Parameters for SPG and GPF: $\Sigma_0 = [100\ 0; 0\ 100]$, $\sigma_E^2 = 1/30$, $\alpha = 0.2$, $\beta = 1$, and 60 particles.

small, we simply discard the effect of the likelihood and update the covariance matrix by a scaler determined by the distance between the particle and the proposal mean.

A. Synthetic Image Sequences

We first generate various synthetic sequences where a spherical target (blue) imitates an unpredictable nonlinear movement from the top-left to the bottom-right by imposing an additive zero-mean Gaussian with a large variance onto a random speed. We apply the proposed SPG, GPF, and mean shift-based tracking algorithms to these sequences respectively, and repeat the experiments with different initial conditions over 100 times. In Fig. 5, several sequences of tracking results are shown for comparison, where the proposal covariance is initialized to be $\Sigma_0 = [100\ 0; 0\ 100]$ for both SPG and GPF, where 100 is the axial variance for each direction and thus 10 is the axial standard deviation (STD). With a high MC $\beta = 1$ and only 60 particles, SPG is able to locate the target exactly in almost every frame, even when the target is partially occluded around Frame 28 and moves so irregularly. Though GPF keeps up with the target, it encounters obvious tracking errors in the frames with abrupt movements, especially around Frames 20 and 76. Mean shift, on the other hand, traces the target in a perfect way as long as the target does not move too far away from its previous location, i.e., the target needs to have overlapping area in two consecutive frames. When it jumps too fast around Frame 76, mean shift simply loses the target and cannot recover it.

To further evaluate the effect of different initial conditions and different MCs in a quantitative way, we calculate the average tracking error between the estimation results and the ground truth for each frame after repeating the simulation over 50 times. In the Fig. 6(a), SPG achieves a much smaller error for almost every frame than GPF. When we increase the initial axial STD from 10 to 20 pixels for GPF, the tracking error decreases significantly for those frames of abrupt target motions and increases a little for others with predictable motions. With the same number of particles, a larger searching space will increase the probability of capturing the abruptly moving target, but the sparser coverage of the particles will inevitably decrease the estimation accuracy. Meanwhile, to verify how the confidence will affect the performance, we repeat the quantitative experiments with $\beta = 0.1$ and $\beta = 0.01$. As shown in Fig. 6(b), as we decrease the confidence, the tracking error of SPG increases due to the less effect from the updating distribution. Till the confidence is very small $\beta = 0.01$, where the intermediate measurement have negligible effect on the PD, SPG behaves more like a GPF, and the tracking error, which is the squared line in Fig. 6(b), has almost the same level as that of GPF, which is the squared line in Fig. 6(a). Even then, SPG still shows some advantages when the target moves abruptly around Frame 66 and Frame 76 due to its dynamic adjustment on the searching space. By tuning the key parameter $\alpha = 0.2$ for the proposal update, we obtain an average tracking error of only 1.127 pixels/frame for SPG with $\beta = 1$. We also vary the number of particles with the same initial conditions, and the results are plotted in Fig. 6(c). With only 30 particles, SPG is able to achieve a similar performance as GPF with over 120 particles in this simple application.

B. Real-World Applications

In the real-world, abrupt motions could result from three scenarios, i.e., fast moving, in low-frame-rate videos, and by an unstable camcorder. We select a sequence from each of these three categories and compare SPG with GPF and mean shift in the following subsection.

1) Fast Moving Targets: As shown in Fig. 7, the table-tennis ball is moving very fast up and down, and changes directions frequently. Since the white is quite unique in the color space, we have a very high confidence and select $\beta = 1$. With 60 particles and $\Sigma_0 = [100\ 0; 0\ 100]$, GPF (the second row) easily loses track due to the rapid movement though it could recapture the target from time to time, while SPG (the first row) successfully traces the target in every frame by capturing the motion information and adapting the sampling. Especially around Frames 10 and 25, when the previous location (the dashed rectangle) has no overlapping area with the current

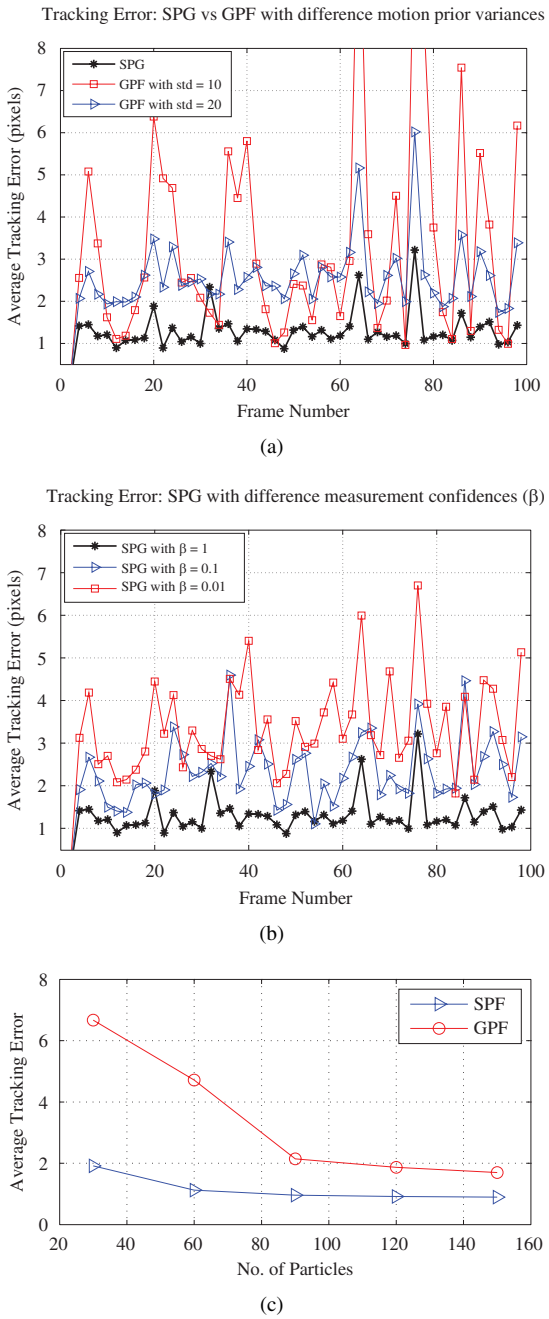


Fig. 6. Quantitative comparison on the *Synthetic* sequence: average tracking errors for Frames 1 ∼ 99. (a) SPG versus GPF, (b) SPG with different MCs, and (c) SPG versus GPF: different number of particles.

position, it is expected that mean shift tracking is unable to capture the motion.

2) Traffic Surveillance Videos: With limited storage, surveillance video usually has a low-frame-rate. A typical sequence for traffic monitoring is shown in Fig. 8 with frame rate 2.5 frames/s. A white car is moving in the snow from the far side toward the video camera, where the speed in the view is increasing due to the perspective effect. In this case, we have less confidence on the color histogram due to the complex background, but we want to test the extreme of the proposed method and still select $\beta = 1$. Amazingly, with 100 particles, SPG (the first row) can easily trace the car in every frame despite a noticeable change in the target scale. With an initial covariance, $\Sigma_0 = [100\ 0; 0\ 100]$, GPF is able to keep track in the first half, but loses track when the target is moving faster, where a large portion of particles are attracted by the second car right behind it. Mean shift experiences difficulties from the very beginning, due to the interference from the white snow and the use of color histogram. We also increase the axial STD from 10 to 20 pixels, and the average tracking errors for both SPG and GPF are plotted in Fig. 9. Again, enlarging the searching space does reduce the tracking error when the car is moving fast, as shown in the late frames in Fig. 8, but leads to larger tracking errors when the target moves slowly with the same number of particles. Compared with both cases, SPG still achieves lower errors in almost every frame.

3) The Handheld Camcorder: The third one is from a video taken by a handheld camcorder, which is common for a lot of outdoor activities nowadays. Hand shaking could easily cause abrupt motions in the camera view. As shown in Fig. 10, due to the instability of the camera holder, the target shows a random and drastic movement in the view. Mean shift (the third row) is not able to handle this situation any more, while SPG and GPF still work. The results around Frame 2467 shows that SPG is more capable of capturing unpredicted motions.

C. Fragment-Based Model versus Color Histogram

So far we have not embedded the proposed fragment-based measurement model into the tracking system. That is because in the sequences given above the targets themselves are relatively small, and the color histogram suffices for the tracking demands even with the highest confidence. However, when the targets are relatively large and experience a complex appearance change in a complex background, tuning the confidence level with the simple model is no longer enough, and a measurement model with a stronger discriminative power is expected. As shown in the first two rows of Fig. 11, people are walking along the shopping center hallway, where two of them have similar color histograms but different textural features. As expected, by decomposing the target into small blocks and extracting spatial features for the measurement, the SPG-1 (the first row) and $\beta = 1$ achieves a high tracking accuracy in almost every frame, while SPG-2 with the color histogram (the second row) yields an unsatisfactory result even though the MC has been tuned down to $\beta = 0.1$. Specifically, around Frame 3135 when target is partially occluded, SPG-2 is somehow dragged downside, and switches to the wrong target after the total occlusion on Frame 3185. In the bottom two rows of Fig. 11, the target is frequently occluded by different walking people. Again, SPG with the fragment-based model (the third row) successfully traces the target, while mean shift (the fourth row) has great difficulty to follow the target due to the interference caused by occlusions and the clutter background, as shown on Frames 2053 to 2103.

In summary, color histogram is efficient and has a high MC in relatively simple scenarios. However, without spatial information, it becomes insufficient for complex background, even though the confidence level can be tuned down. In this case, a more discriminative and representative model, such as

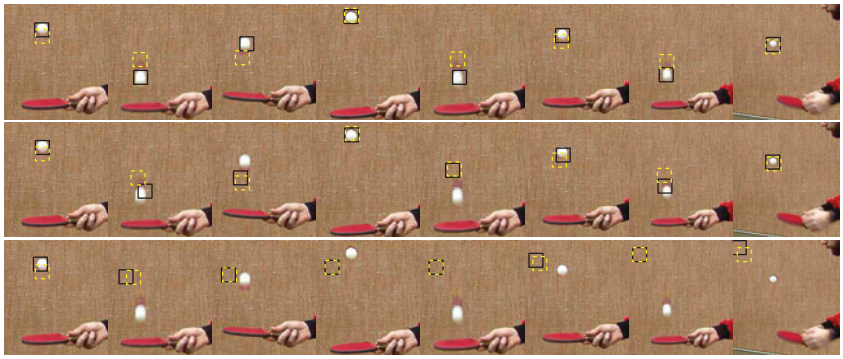


Fig. 7. Tracking results on the *Stennis* sequence for Frames 2, 10, 14, 20, 25, 31, 39, and 47 (with color histogram). To show the target motion pattern, a solid black and a dashed yellow rectangle denote the estimation results in the current and previous frames, respectively. First row: SPG; second row: GPF; third row: mean shift. Parameters for SPG and GPF: $\Sigma_0 = [100\ 0; 0\ 100]$, $\sigma_E^2 = 1/30$, $\alpha = 0.7$, $\beta = 1$, and 60 particles.

Fig. 8. Tracking results on the *Traffic* sequence (from http://i21www.ira.uka.de/image_sequences/) for Frames 88, 128, 178, 198, 218, 238, 248, and 258 (with color histogram) withe frame rate of 2.5 frames/s. To show the target motion pattern, solid black and dashed yellow rectangle denote the estimation results in the current and previous frames, respectively. First row: SPG; second row: GPF; third row: mean shift. Parameters for SPG and GPF: $\Sigma_0 = [100\ 0; 0\ 100]$, $\sigma_E^2 = 1/100$, $\alpha = 1$, $\beta = 1$, and 100 particles.

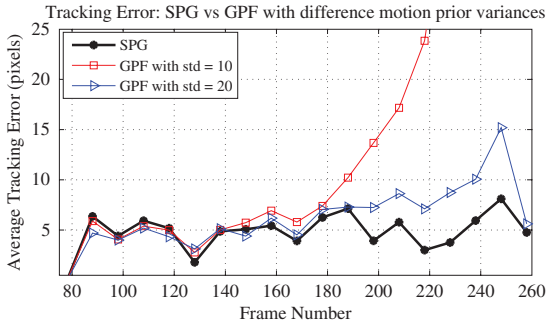


Fig. 9. Quantitative comparison on the *Traffic* sequence: average tracking errors for Frames 88–258.

the proposed fragment-based model, is able to embed more spatial information to enhance the MC. Furthermore, it is easy to switch between global and partial representations of a target, which brings great performance advantages in the case of occlusions. So ideally, for each specific application, the user is expected to carry out some research before selecting an appropriate measurement model. The more confident the user is, the more efficiently the SPG could perform.

V. Discussion

Several key issues are further discussed as follows.

A. Initialization Issue

With no prior on how fast the target is moving, the proposal initialization is a key issue for any method based on PF. With a small initial covariance, GPF is unable to capture abrupt motions, while the tracking errors will inevitably increase with a larger covariance given the same number of particles. The only way to prevent performance degradation is to employ a large number of particles. Even in some adaptive versions of GPF, where this number is adjusted dynamically based on the previous motion, the tracker is not guaranteed to work when the target suddenly changes its motion pattern. However, the adaptive SPG succeeds by combining temporal coherence and sequential detections. By adapting its PD to the most recent measurement, it maximizes the usage of all particles.

B. Updating Distribution

The updating distribution $p_u(\cdot)$ is introduced to incorporate the intermediate measurements into the proposal updating, and it is totally different from the measurement model defined

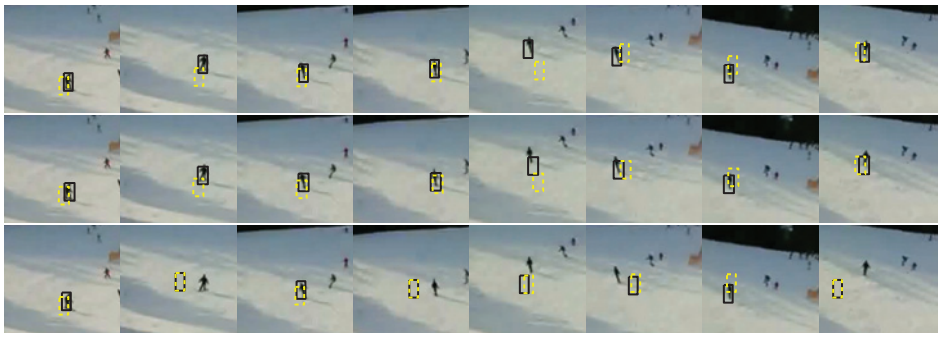


Fig. 10. Tracking results on the *Snowboarding* sequence for Frames 2417, 2431, 2453, 2461, 2467, 2473, 2481, and 2489 (with color histogram). To show the target motion pattern, solid black and dashed yellow rectangles denote the estimation results in the current and previous frames, respectively. First row: SPG; second row: GPF; third row: mean shift. Parameters for SPG and GPF: $\Sigma_0 = [100\ 0; 0\ 100]$, $\sigma_E^2 = 1/100$, $\alpha = 1$, $\beta = 1$, and 120 particles.

Fig. 11. Tracking results on the *Hallway1* sequence (top two rows, Frames 3075, 3105, 3135, 3185, 3215, 3245, 3275, and 3355) and the *Hallway2* sequence (bottom two rows, Frames 1683, 1723, 1763, 1793, 1833, 1953, 2053, and 2103) (CAVIAR: http://homepages.inf.ed.ac.uk/rbf/CAVIAR/). To show the target motion pattern, solid black and dashed yellow rectangles denote the estimation results in the current and previous frames, respectively. First row: SPG-1 (with the fragment-based measurement model, $\beta = 1$); second row: SPG-2 (with color histogram, $\beta = 0.1$); third row: SPG (with the fragment-based measurement model, $\beta = 1$); fourth row: mean shift. Frame rate: 2.5 frames/s. Parameters: $\sigma_E^2 = 50$, $\alpha = 0.8$, $P = 0.7$, 120 particles, and 8×8 pixels for BB.

by $p(z|x)$. The realization of the updating distribution in Gaussian form greatly simplifies the computation, which will neither affect the non-Gaussian property of the measurement model nor sacrifice the PF-like ability to handle nonlinear and multimodal problems. The updating covariance Σ_u is mathematically determined by the measurement result, where σ_E^2 is the measurement variance.

C. Fairness of Sampling

Two aspects are paid special consideration to preserve the fairness of the sampling, i.e., prevent the tracker from being trapped in a local optimum. First, an appropriate measurement model should be carefully selected for each tracking application. The more discriminative the model is, the less likely SPG is going to be trapped. Second, the introduction of the MC provides the user with another layer of protection mechanism to prevent the likelihood information from being utilized excessively, by which the tracker is less likely to be trapped in the local optimum. The user could tune the confidence index β to control how this scheme behaves like the GPF. For applications with varying scenarios, users may tend to be more conservative and select β toward 0. In a word, this scheme is quite flexible in achieving a better tradeoff between efficiency and robustness according to the user's understanding of the application scenario.

D. Computational Complexity

Complexity is always a major concern for online tracking algorithms. First of all, all the experiments of SPG in Section IV are running quasi or total real-time in MATLAB with 5–23 frames/s depending on the measurement complexity and the number of particles. This gives us a prediction on how well the algorithm performs if implemented in an executable file. Secondly, due to the extra overhead to update the proposal, SPG needs more computation than GPF given the same number of particles. The extra overhead involves the variance calculation of the updating distribution in (10) and the proposal update in (11). For a general state vector $x_k \in \Re^n$, the extra overhead of SPG is dominated by the matrix inversion, which is $O(n^3 N)$, where N is the number of particles. For the single target tracking ($n = 2$), we have counted $44N$ scalar multiplications and $15N$ additions as the extra overhead for each frame. We also repeat the SPG and GPF with color histogram on the *Synthetic* sequence for over 200 times on a workstation with an Intel Xeon CPU and 3G RAM. The average costs of SPG and GPF in MATLAB without any optimization are approximately 43.3 and 38.6 ms per frame for 60 particles, respectively. Due to significantly fewer particles needed in SPG, as shown in Fig. 6(c), the computational cost of SPG will be lower to achieve a similar performance as GPF in relatively simple applications or when the target shows abrupt motions.

However, it should be noted that the parallelism property of the GPF could be utilized to boost the computation in an optimized way using multicore CPUs, which SPG is not capable of. In applications where the motion can be well predicted, SPG may have no significant advantage or is even outperformed by the highly optimized GPF, but there are still two primary reasons for us to believe that SPG will bring significant benefits. Firstly, the main advantage of SPG is its sampling adaptation to capture various kinds of abrupt motions, where GPF could easily fail. Secondly, the module of visual tracking has been installed in many embedded systems in which multicore computing is not often employed and computation ability is still limited. For such types of systems, SPG is clearly more advantageous than GPF.

VI. Conclusion

We have introduced a novel probabilistic tracking system, named *sequential particle generation*, in which a new adaptive sampling algorithm and a fragment-based measurement model have been proposed. Particles are generated sequentially through dynamic adjustment of a series of proposal distributions, which is achieved by employing the most recent likelihood information and the measurement confidence. The likelihood is generated by using a novel fragment-based measurement model, where each hypothesized target is decomposed into blocks, and local feature information is extracted and fused together to improve the discriminative strength.

Through experiments, the effectiveness of the new sampling algorithm has been verified. In particular, this algorithm is able to automatically gather particles based on the confidence level for a linearly moving target or to disperse particles to increase search space for unpredictable target motions. Furthermore, by embedding with a stronger discriminative model, SPG is able to survive difficult tracking scenarios, either fast motions or heavy occlusions. Comparison with several existing tracking methods further demonstrates the superiority of SPG in terms of efficiency and adaptability to nonlinear and abrupt motions.

References

[1] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 564–577, May 2003.

[2] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.

[3] J. MacCormick and M. Isard, "Bramble: A Bayesian multiple blob tracker," in *Proc. IEEE Int. Conf. Comput. Vision (ICCV)*, vol. 2. Vancouver, BC, Jul. 2001, pp. 34–41.

[4] C. Hue, J.-P. Le Cadre, and P. Perez, "Sequential monte carlo methods for multiple target tracking and data fusion," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 309–325, Feb. 2002.

[5] W. Qu, D. Schonfeld, and M. Mohamed, "Real-time distributed multi-object tracking using multiple interactive trackers and a magnetic-inertia potential model," *IEEE Trans. Multimedia*, vol. 9, no. 3, pp. 511–519, Apr. 2007.

[6] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. New York: Wiley, 1st ed., 2001.

[7] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters For Tracking Applications*, Norwood, MA: Artech House, 1st ed., 2004.

[8] J. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems," *J. Amer. Statist. Assoc.*, vol. 93, no. 443, pp. 1032–1044, Sep. 1998.

[9] O. Cappe, S. J. Godsill, and E. Moulines, "An overview of existing methods and recent advances in sequential Monte Carlo," *Proc. IEEE*, vol. 95, no. 5, pp. 899–924, May 2007.

[10] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," in *Proc. Eur. Conf. Comput. Vision*, Apr. 1996, pp. 343–356.

[11] G. Kitagawa, "Self-organizing state space model," *J. Amer. Statist. Assoc.*, vol. 93, no. 443, pp. 1203–1212, Sep. 1998.

[12] K. Oka, Y. Sato, Y. Nakanishi, and H. Koike, "Head pose estimation system based on particle filtering with adaptive diffusion control," in *Proc. IAPR Conf. Mach. Vision Applicat. (MVA 2005)*, May 2005, pp. 586–589.

[13] S. K. Zhou, R. Chellappa, and B. Moghaddam, "Visual tracking and recognition using appearance-adaptive models in particle filters," *IEEE Trans. Image Process.*, vol. 13, no. 11, pp. 1491–1506, Nov. 2004.

[14] P. Pan and D. Schonfeld, "Dynamic proposal variance and optimal particle allocation in particle filtering for video tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 9, pp. 1268–1279, Sep. 2008.

[15] P. Pan and D. Schonfeld, "Resource management in particle filtering for multiple object tracking," in *Proc. SPIE Conf. Visual Commun. Image Process.*, San Jose, CA, Jan. 2008, pp. 1–8.

[16] E. Maggio and A. Cavallaro, "Hybrid particle filter and mean shift tracker with adaptive transition model," in *Proc. Int. Conf. Acoustics, Speech, Signal Process.*, vol. 2. Mar. 2005, pp. 221–224.

[17] Y. Cai, N. De Freitas, and J. J. Little, "Robust visual tracking for multiple targets," in *Proc. Eur. Conf. Comput. Vision*, May 2006, pp. 107–118.

[18] F. Porikli and O. Tuzel, "Object tracking in low-frame-rate video," in *Proc. SPIE Image Video Commun. Process.*, vol. 5685. Mar. 2005, pp. 72–79.

[19] N. Bouaynaya and D. Schonfeld, "A complete system for head tracking using motion-based particle filter and randomly perturbed active contour," in *Proc. SPIE, Image Video Commun. Process.*, vol. 5685. Mar. 2005, pp. 864–873.

[20] Y. Li, H. Ai, T. Yamashita, S. Lao, and M. Kawade, "Tracking in low-frame-rate video: A cascade particle filter with discriminative observers of different lifespans," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, Minneapolis, MN, Jun. 2007, pp. 1–8.

[21] C. Chang and R. Ansari, "Real-time tracking with multiple cues by set theoretic random search," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, vol. 1. Jun. 2005, pp. 932–938.

[22] J. Wang, X. Chen, and W. Gao, "Online selecting discriminative tracking features using particle filter," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*. vol. 2. Jun. 2005, pp. 1037–1042.

[23] T. Shakunaga and K. Noguchi, "Robust tracking of appearance by sparse template adaptation," in *Proc. 8th IASTED Int. Conf. Signal Image Process. (SIP '06)*, Aug. 2006, pp. 85–90.

[24] K. Otsuka, J. Yamato, Y. Takemae, and H. Murase, "Conversation scene analysis with dynamic Bayesian network based on visual head tracking," in *Proc. IEEE Int. Conf. Multimedia Expo*, Toronto, ON, Jul. 2006, pp. 949–952.

[25] D. Chen and J. Yang, "Online learning of region confidences for object tracking," in *Proc. 2nd Joint IEEE Int. Workshop Visual Surveillance Performance Eval. Tracking Surveillance*, Oct. 2005, pp. 1–8.

[26] M. Yang, J. Yuan, and Y. Wu, "Spatial selection for attentional visual tracking," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, Minneapolis, MN, Jun. 2007, pp. 1–8.

[27] S. Avidan, "Ensemble tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 2, pp. 261–271, Feb. 2007.

[28] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*. vol. 1. Jun. 2006, pp. 798–805.

[29] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, vol. 1. Dec. 2001, pp. 511–518.

[30] F. Zuo and P. H. N. de With, "Real-time facial feature extraction using statistical shape model and haar-wavelet based feature search," in *Proc. IEEE Int. Conf. Multimedia Expo*, vol. 2. Taipei, Taiwan, Jun. 2004, pp. 1443–1446.

[31] T. Mita, T. Kaneko, and O. Hori, "Joint Haar-like features for face detection," in *Proc. IEEE Int. Conf. Comput. Vision*, vol. 2. Beijing, Oct. 2005, pp. 1619–1626.

[32] C. Kreucher, K. Kastella, and A. O. Hero, III, "Multitarget tracking using the joint multitarget probability density," *IEEE Aerosp. Electron. Syst. Mag*, vol. 41, no. 4, pp. 1396–1414, Oct. 2005.

[33] A. Kong, J. Liu, and W. H. Wong, "Sequential imputation and Bayesian missing data problems," *J. Amer. Statist. Assoc.*, vol. 89, no. 425, pp. 278–288, Mar. 1994.

[34] N. Chopin, "A sequential particle filter method for static models," *Biometrika*, vol. 89, no. 3, pp. 539–552, 2002.
[35] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer-Verlag, 1st ed., 2006.
[36] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *Proc. Eur. Conf. Comput. Vision*, May 2002, pp. 661–675.
[37] Y. Liu and Y. F. Zheng, "Video object segmentation and tracking using psi-learning classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 7, pp. 885–899, Jul. 2005.
[38] P. Viola and M. Jones, "Robust real-time face detection," *Int. J. Comput. Vision*, vol. 57, no. 2, pp. 137–154, May 2004.
[39] K. Nummiaro, E. B. Koller-Meier, and L. V. Gool, "Object tracking with an adaptive color-based particle filters," in *Proc. Symp. Pattern Recognition DAGM*, Sep. 2002, pp. 353–360.

Yuanwei Lao (S'06) received the B.S. and M.S. degrees in information science and electronics engineering at Zhejiang University, Hangzhou, China, in 2000 and 2003, respectively.

He is currently pursuing the Ph.D. degree in the Department of Electrical and Computer Engineering, Ohio State University, Columbus. His research interests include image/video processing, pattern recognition, and their multimedia applications.

Junda Zhu (S'08) received the B.S. and M.S. degrees in information science and electronics engineering, Zhejiang University, Hangzhou, China, in 2004 and 2006, respectively.

He is currently pursuing the Ph.D. degree in the Department of Electrical and Computer Engineering, Ohio State University, Columbus. His research interests are in the field of image and video processing, with emphasis on video object tracking.

Yuan F. Zheng (F'97) received the B.S. degree in engineering physics from Tsinghua University, Beijing, China, in 1970 and the M.S. and Ph.D. degrees in electrical engineering from Ohio State University, Columbus, in 1980 and 1984, respectively.

From 1984 to 1989, he was with the Department of Electrical and Computer Engineering at Clemson University, Clemson, SC. Since 1989, he has been with the Department of Electrical and Computer Engineering, Ohio State University, where currently he is a Professor and was the Chairman of the Department from 1993 to 2004. From 2004 to 2005, he spent a sabbatical year at the Shanghai Jiao Tong University, Shanghai, China, where he continued to be involved as Dean of the School of Electronic, Information, and Electrical Engineering devoting to part-time administrative and research activities until 2008. His research interests include image and video processing for compression, object classification, and object tracking, and robotics, and his current activities are in robotics and automation for high-throughput applications in biological studies. His research has been supported by the National Science Foundation, Air Force Research Laboratory, Office of Naval Research, Department of Energy, DAGSI, and ITEC-Ohio. He has been on the Editorial Board of five international journals.

Dr. Zheng received the Presidential Young Investigator Award from Ronald Reagan in 1986, and the Research Awards from the College of Engineering of the Ohio State University in 1993, 1997, and 2007, respectively. He and his students received the Best Student Paper or Best Conference Paper Awards several times, and received the Fred Diamond Award for Best Technical Paper from the Air Force Research Laboratory in 2006.