Sequential Particle Swarm Optimization for Visual Tracking

Abstract

Visual tracking usually can be formulated as a graphical model and involve a searching process for inferring the motion of an object from uncertain and ambiguous observations. This paper brings a new view to tracking problem from a swarm intelligence perspective. Inspired by the animal swarm intelligence in the evolutionary computing, we propose a sequential particle swarm based searching strategy for robust visual tracking. Unlike the independent particles in the conventional particle filter, the particles in our searching strategy cooperate with each other and evolve according to the *cognitive effect* and *social effect* in analogy with the cooperative and social aspects of animal populations. In the graphical models, we conduct a theoretical analysis from Bayesian inference view, and shows that our searching strategy is essentially a hierarchial importance sampling process which is guided by the swarm intelligence extracted from the particle configuration. The process can incorporate the newest observations into the state transition distribution $p(x_t|x_{t-1})$ to approximate to the 'optimal' importance distribution $p(x_t|x_{t-1}^{(n)}, y_t)$, and thus overcome the sample impoverishment problem suffered by particle filters. Experimental results demonstrate that, compared with the state-of-the-art particle filter and its variation-the unscented particle filter, the proposed tracking algorithm is more robust and effective with both synthetic data and real world sequences.

Index Terms

Particle swarm optimization, visual tracking, graphical model, particle filter, importance sampling

I. Introduction

Visual tracking, as a fundamental task in many visual systems, is to locate the specified region in the video sequence. It has received significant attention due to its crucial value in visual applications, such as surveillance [1], [2], [3], human-computer interaction [4], [5], smart rooms [6], [7], intelligent transportation [8], augmented reality and video compression [9], [10], etc.

Specially, the tracking process usually can be formulated as a graphical model (as shown in Fig. 1), where x_t and y_t represent temporal state and observation respectively. Based on the following two conditional independence assumptions:

1) States follow a first order Markov process

$$p(x_t|x_{t-1}, x_{t-2}, \cdots, x_0) = p(x_t|x_{t-1})$$

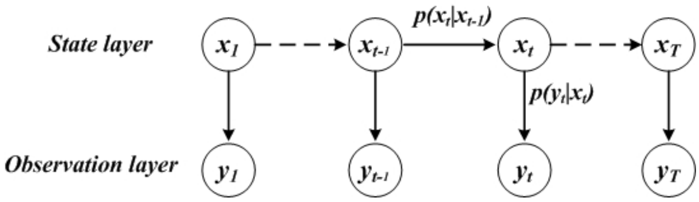


Fig. 1. A graphical model of tracking process

2) Observations depend only on the current state

$$p(y_t|x_t, A) = p(y_t|x_t)$$

the graphical model can be described by the following dynamical state-space model [11]:

$$\text{state transition model } x_t = f_t(x_{t-1}, \epsilon_t) \leftrightarrow p(x_t|x_{t-1}) \tag{1}$$

$$\text{observation model } y_t = h_t(x_t, \nu_t) \leftrightarrow p(y_t|x_t) \tag{2}$$

where ϵ_t, ν_t are the system noise and observation noise, $f_t(.,.)$ and $h_t(.,.)$ are the state transition model and observation model, which characterize probability distributions $p(x_t|x_{t-1})$ and $p(y_t|x_t)$ respectively. As a result, the tracking process can be viewed as the following probabilistic inference problem: given an initial object state density $p(x_0)$, transition density $p(x_t|x_{t-1})$, and observation likelihood $p(y_t|x_t)$, the objective of the tracking process is to infer the real motion state of the object at time t given the uncertain and ambiguous observations up to time t, which is in essence amount to estimating the posterior density $p(x_t|y_{0:t})$. Although the posterior density provides a complete solution to the visual tracking problem, the problem still remains challenging since it contains integration and marginalization operation in the estimating process of the posterior density. When the dynamical system is nonlinear and non-Gaussian, the estimating process is intractable. To address the above issue, we propose a swarm intelligence based solution to approximate and propagate posterior density with a set of particles. The main idea of our work is to approximate the optimal solution through the cooperation between particles which are inspired from the swarm intelligence of animal populations.

A. Related Work

Much effort has been expended to tackle the above problem in recent years, and these approaches can be divided into two categories: parametric ones and non-parametric ones.

Parametric approaches need to assume or approximate $f_t(.,.)$ and $h_t(.,.)$ with a parametric model. When the equations (1)(2) reduce to the linear dynamic systems with Gaussian noise, the seminal the Kalman filter [12] provides an analytical solution, in which the sufficient statistics of mean and covariance matrix are calculated and propagated. However, the use of Kalman filter is limited by the ubiquitous nonlinearity and non-Gaussianity of real world. By linearizing the nonlinear functions with first-order Taylor expansion around the predicted values, extended Kalman filter (EKF) [13], [14] is proposed to solve nonlinear system problems, but such a first order approximation has significant limitations for accurate state estimation. Later, Julier et al. [15] propose an alternative approach to approximate the nonlinearity called unscented Kalman filter (UKF). It first generate a set the sigma points by unscented transformation, and then propagated through the state transition model and observation model to obtain the final filtered result. While UKF is significantly better than EKF in density statistics estimation, it still assumes a Gaussian parametric form of the posterior, thus can not handle multi-modal distributions.

To maintain the multi-modal properties, the non-parametric techniques based on Monte Carlo simulations are introduced, among which the particle filter [11], [16] is the most typical one and is firstly applied for visual tracking in [17]. The basic idea of particle filtering is to use a number of independent random variables called particles, sampled from a proposal distribution, to represent the posterior probability, and to update the posterior by involving the new observations. Although particle filtering has achieved a considerable success in tracking literature, it is faced with a fatal problem-sample impoverishment due to its 'suboptimal sampling' mechanism in the importance sampling process. When the samples are drawn from a proposal distribution which lies in the tail of the observation distribution or the observation distribution is peaked, the performance of the particle filter will be very poor since most particles have low weights, thereby leading to the well-known sample impoverishment problem. Moreover, the sample size needed for estimation will grow exponentially as the dimension of the state space increases.

To improve the sampling efficiency, various techniques are proposed to design an effective proposal distribution, which can be roughly divided into two categories: explicit methods and implicit methods. The explicit approaches attack this problem by explicitly designing a sophisticated proposal distribution. Unscented particle filter [18], [19] adopts the UKF to design the a Gaussian proposal distribution, whose mean and covariance are calculated from UKF filtering. However, the proposal distribution is still limited to Gaussian. Meanwhile, the generation of sigma points and the updating of the covariance for all particles are time-consuming, which do not fulfill the real time requirement. Later, Han et al. [20] improve Merwe's work by modeling the proposal distribution with Gaussian mixture, in which the parameters of the mixture model are found by the well-known kernel method-mean shift. An alternative explicit method model the proposal distribution with an autoregressive (AR) model [17], whose parameters are learnt from pre-labeled video sequences. However, this method often suffers a over fitting problem, consequently feasible only to the training sequences. In contrast, there is no explicit proposal distribution employed in the implicit methods, but the particles are selected based on some heuristic rules. As in ICONDENSATION algorithm [21], the particles in main contour tracker are guided by an auxiliary color tracker. A similar work is the Co-inference approach [22], which formulates the multiple cue integration as a graphical model and factories to achieve feasible inference. In factorized process, the sampling of particles for one cue from other cue's filtering distributions. This process is iterated until convergence. However, a major problem suffered by the both work is that, the cue for guiding the particles must be reliable. In fact, sampling from an unreliable cue's filtering distribution offers no benefit and might even be detrimental to the performance. Another elegant work is the auxiliary particle filter (APF) [23], [24] generates particles from a two-stage procedure: at the first stage, simulate the particles with large predictive likelihoods; at the second stage, reweigh the particles and draw the final states.

Although considerable work have already been made above, a more effective and efficient sampling strategy is still of significant importance for the tracking system.

B. Our Work

Recently PSO (particle swarm optimization) [25], [26], [27], [28], [29], [30], [31], [32], a new population based stochastic optimization technique, has received more and more attention because of its considerable success. Unlike the independent particles in the particle filter, the particles in a PSO interact locally with one another and with their environment in analogy with the cooperative and social aspects of animal populations, for example as found in birds flocking. Starting from a diffuse population, now called a swarm, individuals, now termed as particles, tend to move about the searching space and eventually cluster in regions where optimal states are located. As a result, the advantages of this mechanism are, on one hand, the robustness and sophistication of the obtained group behavior and, on the other hand, the simplicity and low costs of the computations associated with each particle.

Inspired by the forgoing discussions, we propose a sequential particle swarm optimization based algorithm for robust and efficient visual tracking, in which the particles are viewed as intelligent individuals, e.g. birds, and evolve through communicating and cooperating with each other. Meanwhile, we also conduct a theoretical analysis of the proposed approach in the graphical model and Bayesian inference perspective, and find that the proposed algorithm is essentially a novel graphical model with a hierarchial importance sampling based inference process. The hierarchial importance sampling process consists of two stages: 1) a coarse sampling from the state transition distribution $p(x_t|x_{t-1})$, 2) a fine sampling carried out by the PSO iterations which are extracted from the the 'cognitive' and 'social' aspects of particle populations. In this way, the newest observations are gradually taken into consideration to approximate the optimal proposal distribution $p(x_t|x_{t-1}^{(n)}, y_t)$ [16], and thereby greatly overcomes the sample impoverishment problem suffered by the convectional particle filters.

The rest of this paper is structured as follows. A brief introduction to the traditional PSO algorithm is presented in Section 2. In Section 3, the proposed sequential PSO framework is described in detail and the theoretical analysis of our framework is shown in Section 4. Section 5 presents the proposed tracking algorithm in the sequential PSO framework. Experimental results are shown in Section 5, and Section 6 is devoted to conclusion.

II. Particle Swarm Optimization

Particle swarm optimization, originally developed by Kennedy and Eberhart in 1995 [25], is a population based stochastic optimization technique, which is inspired by the social behavior of bird flocking. In detail, a PSO algorithm is initialized with a group of random particles $\{x^{i,0}\}_{i=1}^{N}$ (N is the number of particles). Each particle $x^{i,0}$ has a corresponding fitness value which is evaluated by a fitness model $f(x^{i,0})$, and has a relevant velocity $v^{i,0}$ which directs the movement of the particle. In each iteration, the ith particle moves with the adaptable velocity $v^{i,0}$, which is a function of the best state found by that particle (p^i, for individual best), and of the best state found so far among all particles (g, for global best). Given these two best values, the particle updates its velocity and state with following equations in the nth iteration (as shown in Fig. 2),

$$v^{i,n+1} = \mathcal{X}(v^{i,n} + \varphi_1 u_1(p^i - x^{i,n}) + \varphi_2 u_2(g - x^{i,n})) \tag{3}$$

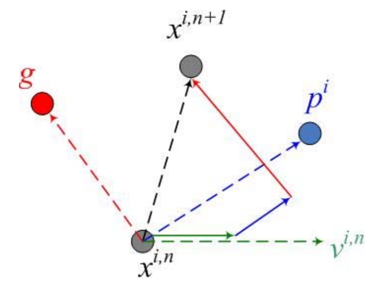


Fig. 2. The nth iteration of particle i

$$x^{i,n+1} = x^{i,n} + v^{i,n+1} \tag{4}$$

where φ_1, φ_2 are acceleration constants, $u_1, u_2 \in (0, 1)$ are uniformly distributed random numbers, and $\mathcal{X}$ is a constriction factor to confine the velocity within a reasonable range: $||v^{i,n}|| \leq v^{max}$. In Equation (12), the three different parts represent *inertial velocity*, *cognitive effect* and *social effect* respectively. The *cognitive effect* and *social effect* are in analogy with the cooperative and social aspects of animal populations. After the nth iteration, the fitness value of each particle is evaluated by a predefined fitness model as follows.

$$f(x^{i,n+1}) = p(y^{i,n+1}|x^{i,n+1}) \tag{5}$$

where $y^{i,n+1}$ is the observation corresponding to the state $x^{i,n+1}$. Then the individual best and global best of the particles are updated based on the fitness value of each particle in the following equations:

$$p^i = \begin{cases} x^{i,n+1}, & \textbf{if } f(x^{i,n+1}) > f(p^i) \\ p^i, & \textbf{else} \end{cases} \tag{6}$$

$$g = \arg\max_{p^i} \ f(p^i) \tag{7}$$

In this way, the particles search for the optima through the above iterations until convergence. In conventional PSO algorithm, there are several parameters to be tuned: constriction factor $\mathcal{X}$, maximum velocity v^{max}, acceleration constants φ_1, φ_2, the maximum number of iterations T, and the initialization of the particles.

III. Sequential Particle Swarm Optimization

A. Motivation

In this section, an interpretation of the tracking process in a stochastic optimization view is presented to show why PSO can achieve good performance in tracking applications.

Let's consider the following version of the tracking problem: suppose there is a groundtruth corresponding to the object (food) in the image (state space) being searched. Suppose a group of particles (birds) are randomly generated in the image (state space), and none of the particles (birds) knows where the object (food) is. But each particle (bird) knows how far it is from the object (food) by evaluating the observation in each iteration. What is the best strategy to find the object (food), and how can the information obtained by each particle (bird) be used efficiently? The PSO framework, inspired by the swarm intelligence–birds flocking, provides an effective way to answer these questions, which motivates us to design a PSO based framework for robust and efficient visual tracking.

However, the tracking task has several major properties which distinguish it from traditional optimization problems:

1) Dynamic: The cost function is influenced by both the object state and the time, and optima may shift spatially, change both height and shape, or come into or go out of existence according to the time.

2) Sequential: The object state follows a first order markov process, which means that the current optimization process is closely related to the previous convergent results.

3) Real-time: The real-time requirement is another major difference between tracking problems and optimization problems.

To effectively tackle such a dynamic optimization problem, we need to answer these questions: a) how to utilize the temporal continuity information between two consecutive frames, b) how to maintain the diversity of the particles in the optimization process, c) how to design a effective convergent criterion to fulfill the real-time requirement.

B. Sequential PSO Based Framework

Motivated by the above discussion, we propose a sequential PSO based framework for visual tracking. To give a clear view, the flowchart of the sequential PSO based framework is schematically shown in Fig. 3. First, the individual best of particles from the previous optimization round are randomly propagated to enhance their diversities. Then, a modified PSO with parameters adaptively tuned is carried out. Finally, an effective convergence criterion is checked to decide whether the PSO iteration stops or not. There are three major stages in the sequential PSO based

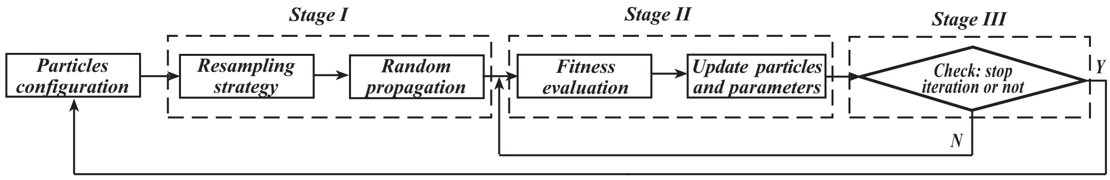


Fig. 3. Overview of the sequential PSO algorithm

framework: random propagation, modified PSO and convergence criterion, which are described in the following sections.

1) Random Propagation: When PSO is applied to such dynamic optimization problems, the major difficulty is the *diversity loss* of particles due to the convergence of the previous optimization process. Thus, a re-diversification mechanism must be employed when the particles are propagated to the next image frame.

The re-diversification process is used to enhance the diversity of particle set, so they can have a higher probability to cover the real object state. Thus, an effective re-diversification mechanism needs to know the prior knowledge of the object motion. In this paper, the particle set is randomly propagated according to a Gaussian transition model whose mean is the previous individual best particle and covariance matrix is determined by the predicted velocity of the object motion.

Given the individual best of particle set $\{p_t^i\}_{i=1}^N$ converged at time t, the re-diversification strategy is carried out as follows.

$$x_{t+1}^{i,0} \sim \mathcal{N}(p_t^i, \Sigma) \tag{8}$$

where Σ is the covariance matrix of the Gaussian distribution, whose diagonal elements are proportional to the predicted velocity v_t^{pred} of the optimum at time t.

$$v_t^{pred} = g_{t-1} - g_{t-2} \tag{9}$$

In our re-diversification strategy, *resampling process* is not needed because the individual best of particle set converged at time t provides a compact sample set for propagation (for the reason, please see the Section *'Convergence Criterion'*). Although randomly propagation according to the predicted velocity is simple, it is sufficient because it is only used to produce an initial value for a subsequent search for the optimal state.

2) Modified PSO: A drawback of the aforementioned version of PSO is the lack of a reasonable mechanism for controlling the acceleration parameters φ_1, φ_2 and the maximum velocity

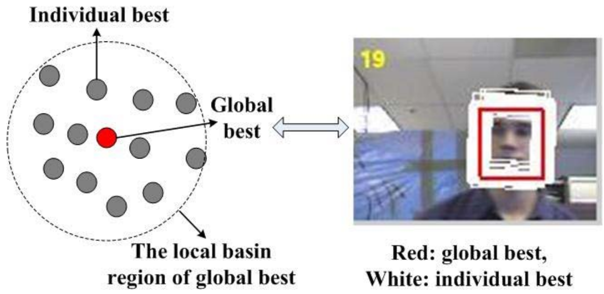


Fig. 4. The convergence criterion of the sequential PSO algorithm

v_t^{max}, fostering the danger of swarm explosion and divergence especially in high-dimensional state space. To overcome this deficiency, we introduce a modified Gaussian swarm optimization version, where the particles and their velocity are updated in the following way,

$$v^{i,n+1} = |randn|(p^i - x^{i,n}) + |Randn|(g - x^{i,n}) + \varepsilon \tag{10}$$

$$x^{i,n+1} = x^{i,n} + v^{i,n+1} \tag{11}$$

where $|randn|$ and $|Randn|$ are the absolute values of the independent samples from the Gaussian probability distribution, such as $\mathcal{N}(0,1)$, and ε is zero-mean Gaussian perturbation noise to avoid trapping in local optimal, which is adaptively controlled by the system noise ϵ.

3) Convergence Criterion: The goal of tracking is to find the object as soon as possible. It is not necessary for all the particles to converge to the object. As a result, the convergence criterion is designed as follows:

$f(g_t)$ >*Th* , where *Th* is a predefined threshold, and all the individual best $\{p_t^i\}_{i=1}^N$ are in a neighborhood of g_t, as shown in Fig. 4, or the maximum iteration number is encountered. According to this criterion, the object to be searched can be efficiently identified and the convergent particle set $\{p_t^i\}_{i=1}^N$ provides a compact initialization without sample impoverishment for the next optimization process, and the temporal continuity information can be naturally incorporated into the sequential PSO framework.

IV. THEORETICAL ANALYSIS OF OUR APPROACH

We will conduct a theoretical analysis of the proposed approach in the graphical model and particle filtering perspective, and show the reason why our approach improves on other sampling based inference techniques.

To make this paper self-contained, we first briefly review the standard particle filtering framework and its major limitation, which are described in more detail in [11]. We then present the detail theoretical analysis.

A. Particle Filtering framework

As shown in Fig. 1, the tracking process can be formulated as a graphical model, and usually can be tackled by the sampling based techniques, in which particle filter is most typical one.

1) Standard Particle Filter: Particle filter [11] is an online Bayesian inference process for estimating the unknown state x_t at time t from a sequential observations $y_{1:t}$ perturbed by noises. Here, let us recall the formulation of the dynamic state-space model employed in the Bayesian inference framework

$$\text{state transition model } x_t = f_t(x_{t-1}, \epsilon_t) \leftrightarrow p(x_t|x_{t-1})$$

$$\text{observation model } o_t = h_t(x_t, \nu_t) \leftrightarrow p(y_t|x_t)$$

The Bayesian inference process is achieved by

$$p(x_t|y_{1:t}) \propto p(y_t|x_t)p(x_t|y_{1:t-1}) \tag{12}$$

where the prior $p(x_t|y_{1:t-1})$ is the propagation of the previous posterior along the temporal axis,

$$p(x_t|y_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1} \tag{13}$$

When the state transition and observation models are nonlinear and non-Gaussian, the above integration is intractable and one has to resort to sampling techniques such as particle filters. The basic idea of particle filter is to use a number of independent random variables called particles $\{x_t^{(n)}\}_{n=1}^N$, sampled directly from the state space, to approximate the posterior distribution. Thus the posterior can be formulated as $p(x_t|y_{1:t}) = \frac{1}{N}\sum_{n=1}^N \delta(x_t - x_t^{(n)})$, where $\delta(\cdot)$ is the Dirac function. Since it is usually impossible to sample from the true posterior, a common solution is to sample from an easy-to-implement distribution, the so-called *proposal distribution* denoted by $q(\cdot)$, hence $x_t^{(n)} \sim q(x_t|x_{t-1}^{(n)}, y_{1:t}), (n = 1, \cdots, N)$, then each particle's weight is set to

$$w_t^{(n)} \propto \frac{p(y_t|x_t^{(n)})p(x_t^{(n)}|x_{t-1}^{(n)})}{q(x_t|x_{t-1}^{(n)}, y_{1:t})}. \tag{14}$$

Finally, the posterior probability distribution is approximated as $p(x_t|y_{1:t}) = \sum_{n=1}^{N} w_t^{(n)} \delta(x_t - x_t^{(n)})$. After the importance sampling step, a re-sampling step is adopted to ensure the efficiency of the particles' evolution. To summarize, the detail process of standard particle filter is presented in Algorithm 1.

Algorithm 1 Standard Particle Filter

1. Initialization: for $n = 1, \cdots, N$, sample $x_0^{(n)} \sim p(x_0)$, $w_0^{(n)} = 1/N$.
2. For time steps $t = \omega 1, 2, \cdots$
3. Importance Sampling: for $n = 1, \cdots, N$, draw samples from the importance proposal distribution as follows:
$$\hat{x}_t^{(n)} \sim q(x_t|x_{t-1}^{(n)}, y_{1:t})$$
4. Weight update: evaluate the importance weights with Equation (14).
5. Normalize the importance weights: $\tilde{w}_t^{(n)} = \frac{w_t^{(n)}}{\sum_{i=1}^{N} w_t^{(i)}}$.
6. Output the statistics of the particles: MMSE or MAP estimate.
7. Resampling: generate N new particles $x_t^{(n)}$ from the set $\{\hat{x}_t^{(n)}\}_{n=1}^{N}$ according to the importance weights $\{\tilde{w}_t^{(n)}\}$.
8. Repeat Steps 3 to 7.

2) Limitation: The proposal distribution $q(\cdot)$ is critically important for a successful particle filter since it concerns putting the sampling particles in the useful area where the posterior is significant. In practice, the dynamic transition distribution $p(x_t|x_{t-1})$ is usually taken as the proposal distribution. However, when $p(x_t|x_{t-1})$ lies in the tail of $p(y_t|x_t)$ (as shown in Fig. 5) or the $p(y_t|x_t)$ is peaked, the performance of the particle filter will be very poor since most particles have low weights, thereby leading to the well-known sample impoverishment problem. In fact, Doucet et al. [16] show that the 'optimal' proposal distribution is $p(x_t|x_{t-1}^i, y_t)$. So the question is, how to incorporate the current observation y_t into the transition model to approximate the optimal proposal distribution in reasonable computation cost.

B. Theoretical Analysis in a Graphical Model and Particle Filtering View

1) A Novel Graphical Model: From the description of Section III, we can see that our tracking process consists of three parts: time propagation, PSO iteration and fitness evaluation.

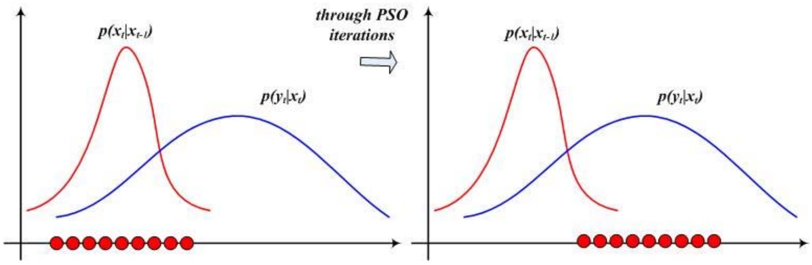


Fig. 5. An illustration of importance sampling (left: sample from $p(x_t|x_{t-1})$, right: after PSO iterations)

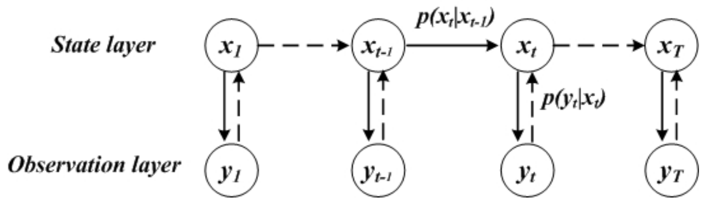


Fig. 6. A novel graphical model of our tracking process

Time propagation and fitness evaluation are corresponding to the state transition model and observation model respectively. The PSO iteration carries out the particle's evolution according to its observation. Based on the three parts, we can formulate the graphical model of our tracking process, which is shown in Fig. 6. Different from the former graphical model illustrated in Fig. 1, our graphical model introduce an arrow from observation node to the state node, aiming at incorporate the current observation into the particle evolution procedure. This procedure is effectively conducted by the PSO iterations.

2) A Particle Filtering Interpretation: From the above graphical model, we can see that our tracking process is a combination of the PSO iterations and the standard particle filter. Unlike the traditional particle filter algorithm which directly samples the particles from the state transition distribution, the particle evolution process in our tracking process is essentially a two-stage sampling strategy to generate samples that approximate to the 'optimal' proposal distribution: first, the particles are sampled from the state transition distribution $p(x_t|x_{t-1})$; second, the sampled particles evolve through the PSO iterations to obtain the final importance sampling.

In more detail, our strategy is essentially a hierarchical importance sampling process. In the coarse importance sampling stage, the particles are firstly sampled from the state transition distribution as in conventional particle filters to enhance their diversity.

$$x_t^{i,0} \sim p(x_t|\tilde{p}_{t-1}^i) \tag{15}$$

In the fine importance sampling stage, the particles evolve through PSO iterations, and are updated according to the newest observations. In fact, this stage is essentially a latent multi-layer importance sampling process with an implicit proposal distribution. Suppose $x_t \in \mathbb{R}^d$ be d-dimensional state, let's focus on one PSO iteration, which is equivalent to the following sampling process.

$$x_t^{i,n+1} \sim \mathcal{N}(x_t^{i,n} + v_t^{i,n}, \Sigma_t^{i,n}) \tag{16}$$

$$\Sigma_t^{i,n} = \begin{pmatrix} \varphi_1^2(p_{t,1}^i - x_{t,1}^{i,n})^2 + \varphi_2^2(g_{t,1} - x_{t,1}^{i,n})^2 & & 0 \\ & \ddots & \\ 0 & & \varphi_1^2(p_{t,d}^i - x_{t,d}^{i,n})^2 + \varphi_2^2(g_{t,d} - x_{t,d}^{i,n})^2 \end{pmatrix}$$

where $\mathcal{N}(\cdot,\cdot)$ is a Gaussian distribution with mean $x_t^{i,n}$, and covariance matrix $\Sigma_t^{i,n}$, in which $x_{t,k}^{i,n}, p_{t,k}^i, g_{t,k}, \varepsilon_k$ are the *k*th element of $x_t^{i,n}, p_t^i, g_t, \varepsilon$ respectively. Although the implicit proposal distribution is also limited to Gaussian distribution, the parameters of the Gaussian distribution is heuristically changed by the individual best state and global best state, thus obtaining a favorable performance.

After this sampling, the newest observations are incorporated by evaluating the fitness values of the particles and updating the two best particles. The detail of this latent multi-layer importance sampling strategy is presented in Algorithm 2.

As shown in Fig.5, when the transition distribution is situated in the tail of the observation likelihood, the particles directly drawn from this distribution do not cover a significant region of the likelihood, and thus the importance weights of most particles are low, resulting to unfavorable performance. In contrast, through hierarchial sampling process in our algorithm, the particles are moved towards the region where the likelihood of observation has larger values, and are finally relocated to the dominant modes of the likelihood, demonstrating the effectiveness of our sampling strategy.

V. Proposed Tracking Algorithm

In this section, we introduce the proposed tracking algorithm and demonstrate how the above-mentioned sequential PSO algorithm is adopted for tracking. In the implementation, our algorithm

Algorithm 2 Latent Multi-layer Importance Sampling

1. Particle set $X_t = \{x_t^{i,0}\}_{i=1}^N$ from the coarse importance sampling stage.
2. **for** $n = 0 : T$ **do**
3. Carry out the sampling
   $x_t^{i,n+1} \sim \mathcal{N}\left(x_t^{i,n}, \Sigma\right)$
4. Incorporate the current observations by evaluating the fitness values
   $f(x_t^{i,n+1}) = p(y_t^{i,n+1}|x_t^{i,n+1})$
5. Update the parameters of the proposal distribution
   $p_t^i = \begin{cases} x_t^{i,n+1}, & \textbf{if } f(x_t^{i,n+1}) > f(p_t^i) \\ p_t^i, & \textbf{else} \end{cases}$
   $g_t = \arg\max_{p_t^i} \ f(p_t^i)$
6. Check the stop criterion: if satisfied, **break**;
7. **end for**
8. Output the sampling result $X_t = \{x_t^{i,n}\}_{i=1}^N$

localizes the tracked object in each video frame using a rectangular window, and the motion of a target object between two consecutive frames is approximated by an affine image warping. Specifically, the motion is characterized by the state of the particle $x_t = (x, y, \theta, s, \alpha, \beta)$ where $\{x, y\}$ denote the 2-D translation parameters and $\{\theta, s, \alpha, \beta\}$ are deformation parameters. The fitness value of each particle is evaluated by a spatial constraint MOG based appearance model. In the following parts, we first introduce the spatial constraint MOG based appearance model, then give a detailed description of the proposed tracking algorithm in the species based sequential PSO based framework.

A. Spatial Constraint MOG Based Appearance Model

The appearance of the target is modeled by a spatial constraint MOG, with the parameters estimated by an on-line EM algorithm.

1) *Appearance Model:* Similar to [33],[34], the appearance model consists of three components S, W, F, where the S component captures temporally stable images, the W component characterizes the two-frame variations, and the F component is a fixed template of the target to prevent the model from drifting away. However, this appearance model treats each pixel

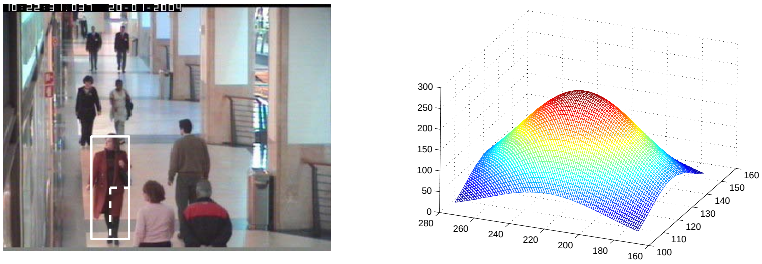


Fig. 7. A 2-D gaussian spatial constraint MOG based appearance model

independently and discards the spatial outline of the target. So it may fail in the case that, for instance, there are several similar objects close to the target or partial occlusion. In our work, we apply a 2-D gaussian spatial constraint to the SWF based appearance model, whose mean vector is the coordinate of the center position and the diagonal elements of the covariance matrix are proportional to the size of the target in the corresponding spatial directions, as illustrated in Fig. 7. Thus the fitness function of particles can be formulated as follows,

$$f_t = f(y_t|x_t) = \prod_{j=1}^{d} \left\{ N(x(j); x_c, \Sigma_c) * \sum_{i=s,w,f} \pi_{i,t}(j) N(y_t(j); \mu_{i,t}(j), \sigma_{i,t}^2(j)) \right\} \tag{17}$$

where $N(x; \mu, \sigma^2)$ is a Gaussian density

$$N(x; \mu, \sigma^2) = (2\pi\sigma^2)^{-1/2} exp \left\{ -\frac{(x-\mu)^2}{2\sigma^2} \right\} \tag{18}$$

$\{\pi_{i,t}, \mu_{i,t}, \sigma_{i,t}, i = s, w, f\}$ represent mixture probabilities, mixture centers and mixture variances of the S, W, F components respectively, y_t is the candidate region corresponding to state of particle x_t and d is the number of pixels inside y_t. $x(j)$,x_c and Σ_c represent the coordinate of the pixel j, the center coordinate of the target and the variance matrix in the spatial space.

Such a spatial constraint appearance model introduces the space information, and it works based on the underlying assumption that the closer the pixel to the center, the more important it is for the model representation. Fortunately, such an assumption is almost always satisfied in real applications.

2) *Parameter Estimation:* In order to make the model parameters depend more heavily on the most recent observation, we assume that the past appearance is exponentially forgotten and new information is gradually added to the appearance model. To avoid having to store all the

data from previous frames, a on-line EM algorithm is used to estimate the parameters of the S, W, F components as follows.

Step1: During the E-step, the ownership probability of each component is computed as

$$m_{i,t}(j) \propto \pi_{i,t}(j) N(y_t(j); \mu_{i,t}(j), \sigma^2_{i,t}(j)) \tag{19}$$

which fulfills $\sum_{i=s,w,f} m_{i,t} = 1$.

Step2: The mixing probability of each component is estimated as

$$\pi_{i,t+1}(j) = \alpha m_{i,t}(j) + (1-\alpha)\pi_{i,t}(j);\ \ i = s, w, f \tag{20}$$

and a recursive form for moments $\{M_{k,t+1}; k = 1, 2\}$ are evaluated as

$$M_{k,t+1}(j) = \alpha o_t^k(j) m_{s,t}(j) + (1-\alpha) M_{k,t}(j);\ \ k = 1, 2 \tag{21}$$

where $\alpha = 1 - e^{-1/\tau}$ acts as a forgotten factor and τ is a predefined constant.

Step3: Finally, the mixture centers and the variances are estimated in the M-step

$$\mu_{s,t+1}(j) = \frac{M_{1,t+1}(j)}{\pi_{s,t+1}(j)},\ \sigma^2_{s,t+1} = \frac{M_{2,t+1}(j)}{\pi_{s,t+1}(j)} - \mu^2_{s,t+1}(j)$$

$$\mu_{w,t+1}(j) = o_t(j),\ \ \sigma^2_{w,t+1}(j) = \sigma^2_{w,1}(j)$$

$$\mu_{f,t+1}(j) = \mu_{f,1}(j),\ \ \sigma^2_{f,t+1}(j) = \sigma^2_{f,1}(j)$$

In fact, updating of the appearance model every frame may be dangerous in case that, for instance, some backgrounds are misplaced into the target or the target is partially occluded. Thus, we developed a selective adaptation scheme to tackle such cases, which is described detailedly in the following section.

B. Selective Adaptation for Appearance Model

In most tracking applications, the tracker must simultaneously deal with the changes of both the target and the environment. So it is necessary to design a adaption scheme for the appearance model. However, over updating of the model may gradually introduce the noise of background into the target model, causing the model drift away finally. Thus, a proper updating scheme is of significant importance for the tracking system.

In this part, we propose a selective updating scheme based on three different confidence measures of the appearance model. First the MAP estimated state is respectively evaluated

TABLE I

SELECTIVE ADAPTATION FOR THE APPEARANCE MODEL

if $(\pi_a > T_a)$
    if $(\pi_{sw} > T_{sw})\&\&(\pi_f > T_f)$
     Update the appearance model of the target;
    else if$(\pi_{sw} > T_{sw})\&\&(\pi_f \leq T_f)$
    Only update the SW components of the appearance model;
    else if$(\pi_{sw} \leq T_{sw})\&\&(\pi_f > T_f)$
    Only update the F components of the appearance model;
    else if$(\pi_{sw} \leq T_{sw})\&\&(\pi_f \leq T_f)$
    Keep the appearance model of the target
    end if
end if

by the appearance model, the SW combined components, and the F component, denoted as π_a, π_{sw}, π_f. And $\{T_a, T_{sw}, T_f\}$ represent three thresholds correspondingly. Each component of the appearance model is updated selectively as shown in Table. I.

It is investigated that S together with W components effectively capture the variations of the target and F prevents the model from drifting away. As a result, such a selective updating strategy not only effectively captures the variations of the target, but also reliably prevents the drifting away problem during the tracking process.

C. Sequential PSO Based Tracking Algorithm

Sequential PSO has provided a general and effective tracking framework. Therefore, we embed the spatial constraint MOG based appearance model into this framework for the fitness value evaluation. The detail of the sequential PSO based tracking algorithm is presented as follows.

VI. EXPERIMENT RESULTS

We compare the performance of our algorithm to several non-linear filters on two different tasks: 1) a 1D state tracking with synthetic data; 2) real world visual tracking problem. All of the experiments are carried out on a CPU Pentium IV 3.2GHz PC with 512M memory.

Algorithm 3 Sequential PSO Based Tracking Algorithm

Input: Given the individual best particles $\{p_t^i\}_{i=1}^N$ at time t;

1. Randomly propagate the particle set to enhance their diversities according to the following transition model

$$x_{t+1}^{i,0} \sim \mathcal{N}(p_t^i, \Sigma)$$

where Σ is a diagonal covariance matrix whose elements are the corresponding variances of affine parameters, i.e., $\sigma_x^2, \sigma_y^2, \sigma_\theta^2, \sigma_s^2, \sigma_\alpha^2, \sigma_\phi^2$.

2. The fitness value of each particle is evaluated by the spatial constraint MOG based observation model as follows.

$$f(x_{t+1}^{i,n}) = p(y_{t+1}^{i,n}|x_{t+1}^{i,n}), i = 1 \cdots N, n = 0 \cdots T$$

3. Update $\{p_{t+1}^i\}_{i=1}^N$ and g_{t+1} by the fitness values obtained above,

$$p^i = \begin{cases} x^{i,n+1}, & \textbf{if } f(x^{i,n+1}) > f(p^i) \\ p^i, & \textbf{else} \end{cases}, \; g = \arg\max_{p^i} \; f(p^i)$$

4. Carry out the PSO iteration.

$$v^{i,n+1} = |randn|(p^i - x^{i,n}) + |Randn|(g - x^{i,n}) + \varepsilon$$

$$x^{i,n+1} = x^{i,n} + v^{i,n+1}$$

5. Check the convergence criterion: if satisfied, continue, otherwise go to step 2;

Output: Global optimum: g_{t+1};

A. 1D State Tracking

Our approach is firstly tested on a non-linear state tracking problem, which is described as benchmark in many papers [18]. Consider the following nonlinear state transition model given by

$$x_{t+1} = 1 + \sin(w\pi t) + \phi_1 x_t + v_t, \qquad x_t \in \mathbb{R} \tag{22}$$

where v_t is a Gamma $\mathcal{G}a(3, 2)$ random variable modeling the process noise, and $w = 4e-2$ and $\phi = 0.5$ are scalar parameters. A non-stationary observation model is as follows

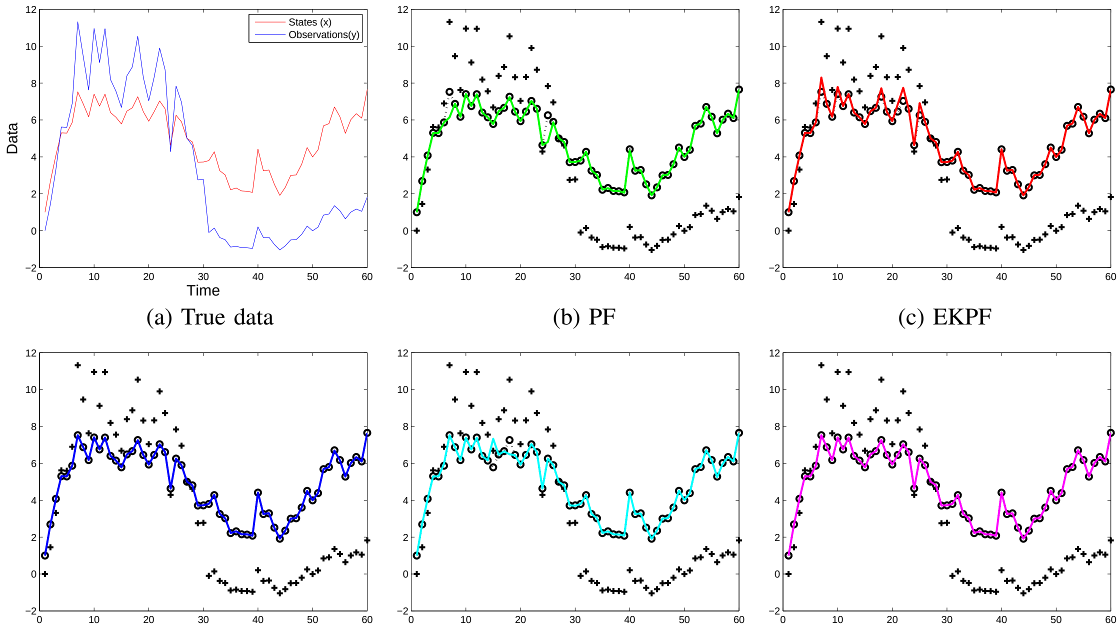
$$y_t = \begin{cases} \phi_2 x_t^2 + n_t, & t \leq 30 \\ \phi_3 x_t - 2 + n_t, & t > 30 \end{cases} \tag{23}$$

TABLE II

EXPERIMENTAL RESULTS OF STATE TRACKING

Algorithm	Proposal	MSE mean	MSE var	Time(s)
Particle filter (PF)	$p(x_t\|x_{t-1})$	0.43272	0.051286	2.1763
Extended Kalman particle filter (EKPF)	$N(\bar{x}_t, \bar{P}_t)$	0.29632	0.011964	6.0026
Unscented particle filter (UPF)	$N(\bar{x}_t, \bar{P}_t)$	0.069229	0.0062162	12.3884
Auxiliary particle filter (APF)	$p(x_t\|x_{t-1})$	0.5563	0.034481	3.4996
Our algorithm	$p(x_t\|x_{t-1})$	0.043998	0.038742	6.2499

where $\phi_2 = 0.2, \phi_3 = 0.5$, and the observation noise n_t is drawn from a Gaussian distribution $N(0, 0.00001)$. Given only the noisy observation y_t, several filters are used to estimate the underlying clean state sequence x_t for $t = 1 \cdots 60$. Here, we compare our approach with conventional particle filter, extended Kalman based particle filter [14], unscented particle filter [18], and auxiliary particle filter [23]. For each algorithm, a proposal distribution is chosen as shown in Table II. Generally, the covariance matrix of the perturbation noise ε is determined by the system noise of the state transition model. Since the system noise in this example is not Gaussian, and the state is 1-dimension, so the variance of ε is set to 0.4. In order to qualitatively gauge performance and discuss resulting issues, the MSE (mean square error) between the tracked points and the true state and the execute time of each run are computed. Fig. 9 gives an illustration of the estimates generated from a single run of the different filters. Since the result of a single run is a random variable, the experiment is repeated 100 times with re-initialization to generate statistical averages. Table 1 summarizes the performance of all the different filters in the following aspects: the means, variances of the mean-square-error (MSE) of the state estimates and the average execute time for one run. It is obvious that the average accuracy of our algorithm is better than generic PF, EKPF, APF and comparable to that of UPF, and the MSE variance of our algorithm is a little higher than other filters, this is because our algorithm takes the g_t as output unlike the mean output of the particles from other filters, so the estimation fluctuates according to the severity with which the observations are contaminated by the noise. Meanwhile, the real-time performance of our algorithm is much better than UPF as Table 1 shows. So the total performance of our algorithm prevails over that of other nonlinear filters.



(a) True data (b) PF (c) EKPF

(d) UPF (e) APF (f) Our algorithm

Fig. 8. An illustration of a single run of different filters

B. Visual Tracking

The affine motion is considered in real motion tracking task. Each candidate image is rectified to a 30×15 patch, and the feature is a 450-dimension vector of gray level values subject to zero-mean-unit-variance normalization. The testing video sequence[1] contains a human face with a rapid motion. In tracking application, $p(x_t|x_{t-1})$ is used to model the object motion, when $p(x_t|x_{t-1})$ is not coincident with the accutal motion, the sampling directly from $p(x_t|x_{t-1})$ will be not efficient. Therefore, although this sequence seems simple, its rapid and arbitrary motion is an effective testing the different improvements of sampling strategy.

In this experiment,, we firstly apply these filters (except EKPF) to this video sequence to demonstrate the effectiveness of the sampling strategy in our algorithm. Then, a detail investigation is presented to show why SPSO has advantages over the other nonlinear filters.

In our implementation, the parameters in the particle filter, auxiliary particle filter and unscented particle filter are set to $\{N = 200, \Sigma_s = diag(8^2, 8^2, 0.02^2, 0.02^2, 0.002^2, 0.002^2)\}$ corresponding to the number of particles and the covariance matrix of the transition distribution

[1]The sequence is available at http://vision.stanford.edu/ birch/headtracker/seq/.

TABLE III

QUANTITATIVE RESULTS OF THE TRACKING PERFORMANCE

Algorithm	Frames Tracked	MSE of Position (by pixels)
PF	5/31	26.1551
UPF	31/31	3.2097
APF	12/31	20.8094
Our algorithm	31/31	2.0829

respectively. To give a convincing comparison, the sequential PSO algorithm is calibrated in the same metric, implementing with the same covariance matrix and the same particle number in each iteration.

As shown in Fig. 9(a) and Fig. 9(c), the PF based tracker and APF based fail to track the object very soon, because it can not catch the rapid motion of the object. More particles and an enlargement for the diagonal elements of the covariance matrix would improve its performance, but this strategy involves more noises and a heavy computational load, and it may trap in the curse of dimensionality when the dimension of the state increases. Fig. 9(b) shows the tracking performance of the unscented particle filter, from which we can notice that the tracker follows the object throughout the sequence, but the localization accuracy is unsatisfactory. In comparison, our method, which utilizes individual and environmental information in the search space, never loses the target and achieves the most accurate results (see Fig. 9(d)). Furthermore, we have conducted a quantitative evaluation of these algorithms, and have a comparison in the following aspects: frames of successful tracking, MSE (mean square error) between the estimated position and the labeled groundtruth. In table 1, it is clear that the PF tracker and APF tracker fails at frame 5, while the UPF and SPSO trackers succeed in tracking throughout the sequence. Additionally, the SPSO tracker outperforms the UPF tracker in term of accuracy.

A investigation shows the underlying reasons for the above experimental results. The undesired behavior of particle filter in Fig. 9(a) is caused by the sample impoverishment in its particle generation process. Let's focus on the frame 19 when the PF tracker loses the target. Here, the particles are sampled from the Gaussian based transition distribution to catch the object motion.

(a) Particle filter

(b) Unscented particle filter

(b) Auxiliary particle filter

(c) Sequential PSO

Fig. 9. Tracking performances of a human face with rapid motion

When the object has rapid and arbitrary motion, the particles drawn from this distribution do not cover a significant region of the likelihood (as shown in top-left of Fig.10), and thus the weights of most particles are low, leading to the tracking failure. As for the unscented particle filter, the sigma-states are generated by UT (unscented transformation) and propagated (as shown in top-right Fig.10), and the weighted mean and covariance are calculated to form a better proposal distribution, thus enhancing the tracking performance to some degree. However, the estimation accuracy of UT is only to the second-order for non-Gaussian data, which may not be coincident with actual motion and thus leads to inaccurate localization. Meanwhile, the generation of sigma-states and the updating of the covariance are time-consuming. While the SPSO framework extracts the local and global information in the particle configuration, and incorporates the newest observations into the proposal distribution, resulting in a better performance. The bottom row of Fig.10 shows the multi-layer importance sampling processes in SPSO framework, which pulls the particles to significant regions of likelihood. As a result, the SPSO framework can handle this rapid motion even with a smaller particle number.

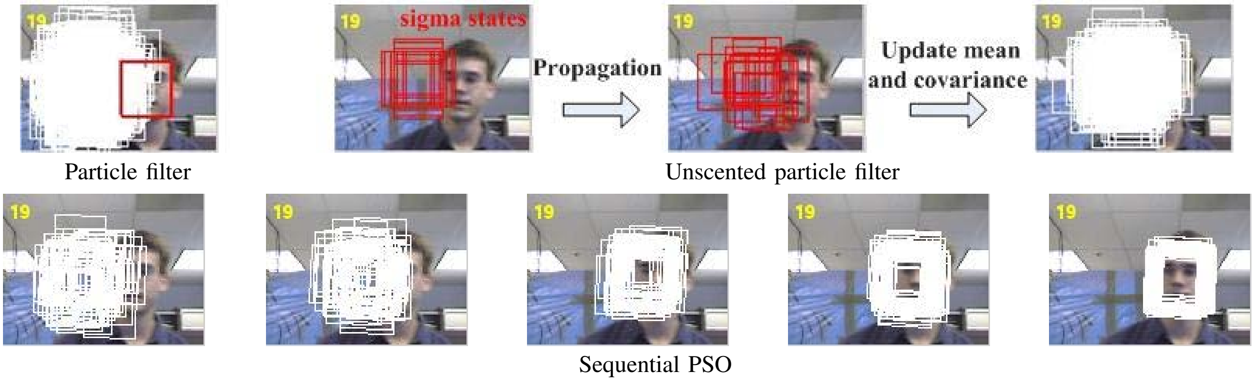


Fig. 10. Tracking procedure of the frame 19

C. Tracking Results of Different Scenes

In order to further evaluate the performance of the proposed tracking framework, it is tested on three video sequences with different environments. The first video sequence contains a man walking across a lawn with a cluttered background, large appearance and illumination changes. In the second video sequence, a pedestrian walks with a large pose change (bows down to reach the ground and stands back up later). Both of these two video sequences are taken from moving cameras outdoors. The third video sequence is a figure skating match, which contains a figure skater with a drastic motion.

From Fig.11(a), we can see that the online updating scheme easily absorbs the appearance and illumination changes, and our tracking framework provides an effective solution to follow the walking man in the cluttered background, because the sequential PSO framework is very effective at finding the global optimum. Fig.11(b) shows the result of tracking the walking pedestrian, demonstrating the effectiveness of our framework in tracking the large pose changes. A tracking result of the figure skater with agile motions is shown in Fig.11(c), which demonstrates that our algorithm has the ability to track the object where large movements exist between two successive frames.

VII. Conclusion

A new sequential particle swarm optimization framework for visual tracking has been proposed in this paper. The sequential information required by the tracking process is incorporated into the modified PSO to make this swarm technique properly suited for tracking. In addition, we have

(a) clutter background, large appearance and illumination changes

(b) large pose changes

(c) drastic motion

Fig. 11. More experimental results

reformulated the SPSO framework in a Bayesian way, and found that it is essentially a multi-layer importance sampling based particle filter. Furthermore, this framework has been naturally extended to multi-object tracking as multi-modal optimization. In experiments, the sequential PSO based tracker is compared very favorably with the particle filter and the unscented particle filter, both in terms of accuracy and efficiency, demonstrating that the sequential PSO is a promising framework for visual tracking.

In summary, the sequential PSO provides a more reasonable mechanism and an more effective way to tackle the dynamic optimization problems than sequential Monte Carlo methods. So it has many other potential applications in computer vision, including image registration, template matching and dynamic background modeling.

REFERENCES

[1] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4 :real-time surveillance of people and their activities", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 809–830, August 2000.

[2] R. Collins, A. Lipton, H. Fujiyoshi, and T. Kanade, "Algorithms for Cooperative Multisensor Surveillance", *Proc. IEEE*, vol. 89, no. 10, pp. 1456-1477, 2001.

[3] W.M. Hu, T.N. Tan, L. Wang, and S.J. Maybank, "A Survey on Visual Surveillance of Object Motion and Behaviors", *IEEE Trans. on Systems, Man and Cybernetics, Part C: Applications and Reviews*, vol. 34, no. 3, pp. 334-352, 2004.

[4] B.Stenger, A.Thayananthan, P.H.S.Torr, and R.Cipolla, "Model-based hand tracking using a hierarchical Bayesian filter", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 28, No. 9, pp. 1372- 1384, 2006.

[5] Y. Wu, J. Lin and T. S. Huang, "Analyzing and Capturing Articulated Hand Motion in Image Sequences", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, pp. 1910-1922, Dec, 2005.

[6] S. Intille, J. Davis, and A. Bobick, “Real-Time Closed-World Tracking”, *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 697-703, 1997.

[7] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer, “Multi-Camera Multi-Person Tracking for Easy Living”, *Proc. IEEE Intl Workshop Visual Surveillance*, pp. 3-10, 2000.

[8] J.Kato, T.Watanabe, S.Joga, Y.Liu, and H.Hase, “An HMM/MRF-based stochastic framework for robust vehicle tracking”, *IEEE Trans, Intelligent Transportation Systems*, vol. 5, no. 3, pp. 142 - 154, 2004.

[9] A.D. Bue, D. Comaniciu, V. Ramesh, and C. Regazzoni, “Smart Cameras with Real-Time Video Object Generation”, *Proc. IEEE Intl Conf. Image Processing*, vol. III, pp. 429-432, 2002.

[10] B. Menser, M. Brunig, “Face detection and tracking for video coding applications”, *Asilomar Conf. on Signals, Systems, and Computers*, pp.49-53, 2000.

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

[12] R. E. Kalman, “A new approach to linear filtering and prediction problems,”, *Transactions of the Am. Soc. of Mechanical Eng., D, Journal of Basic Engineering*, vol. 82, pp. 35-45, 1960.

[13] B. Anderson, and J. Moore, “Optimal filtering”, Englewood Cliffs, Prentice Hall, NewJersy, 1979.

[14] D. Freitas, M. A. Niranjan, A. H. Gee and A. Doucet, ”Sequential Monte Carlo Methods to Train Neural Network Models”, *Neural Computation*, 12(4): 955-993, 2000

[15] S. J. Julier, and J. K. Uhlmann, “A general method for approximating nonlinear transformations of probability distributions”, *Technical report*, RRG, Dept. of Engineering Science, University of Oxford

[16] A. Doucet, S. Godsill, and C. Andrieu, “On sequential Monte Carlo sampling methods for Bayesian filtering”, *Statistics and Computing*, vol. 10, no. 3, pp. 197C208, 2000.

[17] M. Isard and A. Blake, “Condensation: conditional density propagation for visual tracking”, *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5-28, 1998.

[18] R. Merwe, A. Doucet, N. Freitas, and E. Wan, “The unscented particle filter”, *Technical Report CUED/F-INFENG/TR 380*, Cambridge University Engineering Department, August 2000.

[19] Y. Rui, and Y. Chen, “Better proposal distributions: Object tracking using unscented particle filter”, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 786-793, 2001.

[20] B. Han, Y. Zhu, D. Comaniciu, and L. Davis, “Kernel-based Bayesian filtering for object tracking”, *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 227-234, 2005.

[21] M. Isard, and A. Blake, “ICondensation: Unifying low-level and high-level tracking in a stochastic framework”, *Proceeding of European Conference on Computer Vision*, vol. pp. 893-908, 1998.

[22] Y. Wu, and T. Huang, “Robust Visual Tracking by Integrating Multiple Cues Based on Co-Inference Learning”, *International Journal of Computer Vision*, vol.58, no.1, pp. 55-71, 2004.

[23] M. K. Pitt, N. Shephard, “Filtering Via Simulation: Auxiliary Particle Filters”, *Journal of the American Statistical Association*, 94(446): 590-599, 1999.

[24] S.J. McKenna, and H. Nait-Charif, “Tracking human motion using auxiliary particle filters and iterated likelihood weighting”, *Image and Vision Computing*, vol. 25, no. 6, pp. 852-862, 2007.

[25] J. Kennedy, and R. C. Eberhart, “Particle swarm optimization”, *Proceedings of IEEE International Conference on Neural Networks*, Piscataway, pp. 1942-1948, 1995.

[26] M. Clerc, and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space", *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58-73, 2002.

[27] M. P. Wachowiak, R. Smolikova, and Y. Zheng, "An approach to multimodal biomedical image registration utilizing particle swarm optimization", *IEEE Trans. on Evolutionary Computation*, vol. 8, no. 3, pp. 289-301, 2004.

[28] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: simpler, maybe better", *IEEE Trans. on Evolutionary Computation*, vol. 8, no. 3, pp. 204-210, 2004.

[29] K. E. Parsopoulos and M. N. Vrahatis, "On the Computation of All Global Minimizers Through Particle Swarm Optimization", *IEEE Trans. on Evolutionary Computation*, vol. 8, no. 3, pp. 211-224, 2004.

[30] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients", *IEEE Trans. on Evolutionary Computation*, vol. 8, no. 3, pp. 240-255, 2004.

[31] S. Janson and M. Middendorf, "A Hierarchical Particle Swarm Optimizer and Its Adaptive Variant", *IEEE Trans. on Systems, Man, And Cybernetics-Part B: Cybernetics*, vol. 35, no. 6, 2005.

[32] V. Kadirkamanathan, K. Selvarajah, and P. J. Fleming, "Stability analysis of the particle dynamics in particle swarm optimizer", vol. 10, no. 3, pp. 245-255, 2006

[33] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi, "Robust online appearance models for visual tracking", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1296-1311, 2003.

[34] S. Zhou, R. Chellappa, and B. Moghaddam, "Visual Tracking and Recongnition Using Appearance-adaptive Models in Particles Filters", *IEEE Transaction on Image Processing*, 13(11):1491-1506, 2004.