

Sequential Particle Swarm Optimization for Visual Tracking

Xiaoqin Zhang¹, Weiming Hu¹, Steve Maybank², Xi Li¹, Mingliang Zhu¹

¹National Laboratory of Pattern Recognition, Institute of Automation, Beijing, China

{xqzhang, wmhu, lixi, mlzhu}@nlpr.ia.ac.cn

²School of Computer Science and Information Systems, Birkbeck College, London, UK

sjmaybank@dcs.bbk.ac.uk

Abstract

Visual tracking usually involves an optimization process for estimating the motion of an object from measured images in a video sequence. In this paper, a new evolutionary approach, PSO (particle swarm optimization), is adopted for visual tracking. Since the tracking process is a dynamic optimization problem which is simultaneously influenced by the object state and the time, we propose a sequential particle swarm optimization framework by incorporating the temporal continuity information into the traditional PSO algorithm. In addition, the parameters in PSO are changed adaptively according to the fitness values of particles and the predicted motion of the tracked object, leading to a favourable performance in tracking applications. Furthermore, we show theoretically that, in a Bayesian inference view, the sequential PSO framework is in essence a multi-layer importance sampling based particle filter. 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, especially when the object has an arbitrary motion or undergoes large appearance changes.

1. Introduction

Visual tracking has emerged as a central problem in many applications such as surveillance, vision-based control, human-computer interfaces, intelligent transportation, and augmented reality. Recent years have witnessed great advances in the literature, e.g. the snakes model [1], template matching [2], mean shift [3], condensation [4], appearance models [5], probabilistic data association [6] and so on.

Most of the existing tracking algorithms can be formed as an optimization process, which are typically tackled using either deterministic methods [1, 2, 3, 7, 8, 9] or stochastic methods [4, 11, 12, 13, 14, 15, 16]. Deterministic methods usually involve a gradient descent search to minimize a cost function. The snakes model introduced by Kass et

al. [1] is a good example. The aim is to obtain a tight contour enclosing the object by minimizing an energy function. In [2], the cost function is defined as the sum of squared differences between the observation candidate and a fixed template. Then the motion parameters are found by minimizing the cost function through a gradient descent search. Mean shift, which firstly appeared in [10] as an approach for estimating the gradient of a density function, is applied by Comaniciu [3] to visual tracking, in which the cost function between two color histograms is minimized through the mean shift iterations. In general, deterministic methods are usually computationally efficient but they easily become trapped in local minima. In contrast, stochastic methods introduce some stochastic factors into the searching process in order to have a higher probability of reaching the global optimum of the cost function. For example, in [11], object tracking is viewed as an online MAP (maximum a posterior) problem, which is solved by randomly generating a large number of particles to find the maximum of the posterior distribution. Bray et al. [12] use the stochastic meta-descent strategy to adapt the step size of the gradient descent search, and thus avoid local minima of the optimization process in articulated structure tracking. Leung and Gong [13] incorporate random subsampling into mean shift tracking to boost its efficiency and robustness for low-resolution video sequences. Compared with the deterministic counterparts, stochastic methods are usually more robust, but they suffer a large computational load, especially in high-dimensional state space. Although considerable work has already been done above, a more effective optimization method is still intensively needed for robust visual tracking.

Recently PSO (particle swarm optimization) [17, 18, 19, 20], a new population based stochastic optimization technique, has received more and more attentions because of its considerable success in solving nonlinear, non-differentiable, multimodal optimization problems. Unlike other particle based stochastic optimization techniques such as genetic algorithms, the particles in PSO interact locally with one another and with their environment in analogy

with the 'cognitive' and 'social' aspects of animal populations, found in fish schooling, birds flocking, and insects swarming. Starting from a diffuse population, now called a swarm, individuals, now termed particles, move about the search space and eventually cluster in the regions where the optima are located. The advantages of this mechanism are, on one hand, the robustness and sophistication of the group behavior, and on the other hand, the simplicity and low cost of the computation associated with each particle.

In view of the forgoing discussion, we propose a sequential PSO based tracking framework. To the best of our knowledge, the proposed framework is new in the tracking literature. The main contributions of this paper are as follows. The sequential information in the tracking process is effectively incorporated into PSO method to form a robust tracking framework, in which any appearance models can be used. Meanwhile, we show theoretically that, in a Bayesian inference view, the PSO iterations are essentially a swarm-intelligence guided multi-layer importance sampling strategy which incorporates the new observations into a sampling stage, and thus avoids the sample impoverishment problem suffered by the particle filter.

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. Section 4 presents the proposed tracking algorithm in the sequential PSO framework. Experimental results are shown in Section 5, and Section 6 is devoted to conclusion.

2. Particle Swarm Optimization

Particle swarm optimization, originally developed by Kennedy and Eberhart in 1995 [17], 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 the observation model $f(x^{i,0})$, and has a relevant velocity $v^{i,0}$ which directs the movement of the particle. In each iteration, the i th 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 n th iteration (as shown in Fig. 1),

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

$$x^{i,n+1} = x^{i,n} + v^{i,n+1} \quad (2)$$

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 (1), the three different parts

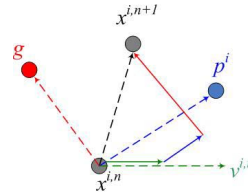


Figure 1. The n th iteration of particle i

represent *inertial velocity*, *cognitive effect* and *social effect* respectively.

After the n th iteration, the fitness value of each particle is evaluated by a predefined observation model as follows.

$$f(x^{i,n+1}) = p(o^{i,n+1} | x^{i,n+1}) \quad (3)$$

where $o^{i,n+1}$ is the observation corresponding to the state $x^{i,n+1}$. Then the individual best and global best of particles are updated in the following equations:

$$p^i = \begin{cases} x^{i,n+1}, & \text{if } f(x^{i,n+1}) > f(p^i) \\ p^i, & \text{else} \end{cases} \quad (4)$$

$$g = \arg \max_{p^i} f(p^i) \quad (5)$$

In this way, the particles search for the optima (here, assuming optimization means maximizing) through the above iterations until convergence. In 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.

3. Sequential Particle Swarm Optimization

3.1. 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.

Essentially, visual tracking is the successive localization of a specific region in a video sequence. 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, in tracking applications, the data is typically a time sequence, and hence the task is essentially a dynamic

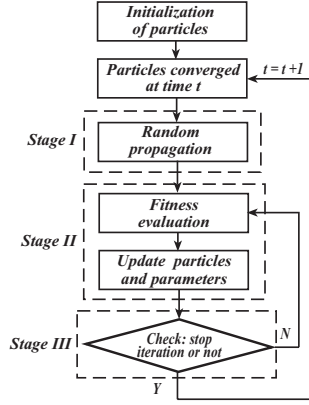


Figure 2. Overview of the sequential PSO algorithm

optimization problem which distinguishes it from traditional optimization problems. In this case, 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.

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.

3.2. 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.2. First, the individual best of particles from the previous optimization round are randomly propagated to enhance their diversities. Then, the 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 framework: random propagation, adaptive PSO and convergence criterion, which are described in the following sections.

3.2.1 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.

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 $\{p_t^i\}_{i=1}^N$ con-

verged at time t , the re-diversification strategy is carried out as follows.

$$x_{t+1}^{i,0} \sim \mathcal{N}(p_t^i, \Sigma) \quad (6)$$

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 .

$$v_t^{pred} = g_{t-1} - g_{t-2} \quad (7)$$

Meanwhile, the velocity $v_{t+1}^{i,0}$ is sampled from the uniform distribution $\mathcal{U}(0, v_t^{pred})$.

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 3.2.3). 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.

3.2.2 Adaptive 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 v_t^{max} , fostering the danger of swarm explosion and divergence especially in high-dimensional state space. To overcome this deficiency, we propose a modified PSO by self-tuning its parameters, where the acceleration parameters φ_1, φ_2 can be set as follows,

$$\varphi_1 = 2f(p^i)/(f(p^i) + f(g)) \quad (8)$$

$$\varphi_2 = 2f(g)/(f(p^i) + f(g)) \quad (9)$$

Compared with the canonical PSO in [17] which constantly sets these parameters to 2, our strategy is more reasonable. Meanwhile, the equations (8)(9) demonstrate that the preference for the 'cognitive' part or 'social' part is determined by their fitness values.

In tracking applications, the maximum velocity v^{max} provides a reasonable bound in order to cover the particle's maximum motion and prevent the particle from arbitrary moving. Traditionally, the maximum velocity v^{max} is set to a predefined constant. However, it is not reasonable when the object has an arbitrary motion. Therefore, we propose a novel scheme for selecting v_t^{max} based on predicted velocity v_t^{pred} of the optimum.

$$v_t^{max} = 1.2 * v_t^{pred} \quad (10)$$

So \mathcal{X} is set to

$$\mathcal{X} = \begin{cases} \|v_t^{max}\| / \|v_t^{i,n+1}\|, & \text{if } \|v_t^{i,n+1}\| > \|v_t^{max}\| \\ 1, & \text{else} \end{cases} \quad (11)$$

In this way, the maximum velocity v^{max} is heuristically selected by utilizing the motion information in the previous tracking process, and thus provides a reasonable limitation to the moving of particles and a certain capability to absorb their acceleration.

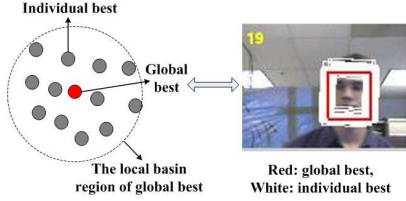


Figure 3. The convergence criterion of the sequential PSO algorithm

3.2.3 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.3, 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.

4. A Bayesian Inference Interpretation of Sequential PSO

We investigate the sequential PSO in a Bayesian inference view, and find that sequential PSO is a unified framework which combines the multi-layer importance sampling and particle filter. The multi-layer importance sampling stage incorporates the newest observations into importance sampling process to approximate the 'optimal' proposal distribution $p(x_t|x_{t-1}, o_t)$ [21].

To make this paper self-contained, we first briefly review the standard particle filter and its major limitation, which are described in more detail in [22]. We then present multi-layer importance sampling carried out by PSO iterations.

4.1. Standard Particle Filter

Particle filter [22] is an online Bayesian inference process for estimating the unknown state x_t at time t from a sequential observations $o_{1:t}$ perturbed by noises. A dynamic state-space form employed in the Bayesian inference framework is shown as follows,

$$\text{state transition model } x_t = f_t(x_{t-1}, \epsilon_t) \leftrightarrow p(x_t|x_{t-1}) \quad (12)$$

$$\text{observation model } o_t = h_t(x_t, \nu_t) \leftrightarrow p(o_t|x_t) \quad (13)$$

where x_t, o_t represent system state and observation, ϵ_t, ν_t are the system noise and observation noise. $f_t(\cdot, \cdot)$ and $h_t(\cdot, \cdot)$ are the state transition and observation models, which are determined by probability distributions $p(x_t|x_{t-1})$ and $p(o_t|x_t)$ respectively. The key idea of particle filter is to approximate the posterior probability distribution $p(x_t|o_{1:t})$ by a set of weighted samples $\{x_t^i, w_t^i\}_{i=1}^N$,

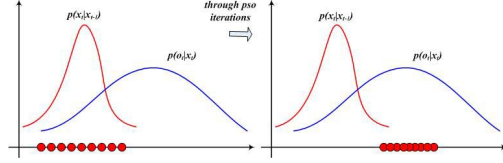


Figure 4. An illustration of importance sampling (left: sample from $p(x_t|x_{t-1})$, right: after PSO iterations)

which are sampled from a proposal distribution $q(\cdot)$, i.e. $x_t^i \sim q(x_t|x_{t-1}^i, o_{1:t})$, ($i = 1, \dots, N$), and then each particle's weight is set to

$$w_t^i \propto \frac{p(o_t|x_t^i)p(x_t^i|x_{t-1}^i)}{q(x_t^i|x_{t-1}^i, o_{1:t})} \quad (14)$$

Finally, the posterior probability distribution is approximated as $p(x_t|o_{1:t}) = \sum_{i=1}^N w_t^i \delta(x_t - x_t^i)$, where $\delta(\cdot)$ is the Dirac function.

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, it is unreasonable when $p(x_t|x_{t-1})$ lies in the tail of $p(o_t|x_t)$ (as shown in Fig.4). In fact, Doucet et al. [21] show that the 'optimal' proposal distribution is $p(x_t|x_{t-1}^i, o_t)$. So the question is, how to incorporate the current observation o_t into the transition model to form an effective proposal distribution in reasonable computation cost.

4.2. Multi-layer Importance Sampling and Incorporation of Measurement

From the above description, we can see that the sequential PSO is a combination of the PSO iterations and the particle filtering procedure. Unlike the traditional particle filter algorithm which directly samples the particles from state transition distribution, the PSO iterations employed in our framework is essentially a multi-layer importance sampling stage which progressively updates the sampled particles based on the newest observations.

Specially, the initial particles in sequential PSO method are firstly sampled from the transition distribution as follows.

$$x_{t+1}^{i,0} \sim \mathcal{N}(p_t^i, \Sigma) \quad (15)$$

In each PSO iteration, the particles are updated according to the newest observations. The detail of the multi-layer importance sampling strategy is presented in Algorithm 1.

As shown in Fig.4, the particles directly sampled from transition model are situated in the tail of the observation likelihood. In contrast, through the PSO iterations, 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.

Algorithm 1 Multi-layer Importance Sampling

1. Initialization $X_{t+1} = \{x_{t+1}^{i,0}\}_{i=1}^N$
 2. **for** $n = 0 : T$ **do**
 3. Carry out the PSO iteration based on Equ.(1)(2)

$$v_{t+1}^{i,n+1} = \mathcal{X}(v_{t+1}^{i,n} + \varphi_1 u_1 (p_{t+1}^i - x_{t+1}^{i,n}) + \varphi_2 u_2 (g_{t+1} - x_{t+1}^{i,n}))$$

$$x_{t+1}^{i,n+1} = x_{t+1}^{i,n} + v_{t+1}^{i,n+1}$$
 4. Evaluation of fitness values

$$f(x_{t+1}^{i,n+1}) = p(o_{t+1}^{i,n+1} | x_{t+1}^{i,n+1})$$
 5. Incorporation of the current observations

$$p_{t+1}^i = \begin{cases} x_{t+1}^{i,n+1}, & \text{if } f(x_{t+1}^{i,n+1}) > f(p_{t+1}^i) \\ p_{t+1}^i, & \text{else} \end{cases}$$

$$g_{t+1} = \arg \max_{p_{t+1}^i} f(p_{t+1}^i)$$
 5. Update the parameters.
 6. Check the convergence criterion: if satisfied, **break**;
 7. **end for**
 8. Output the particles set $X_{t+1} = \{x_{t+1}^{i,n}\}_{i=1}^N$
-

5. Proposed Tracking Algorithm

In this section, we introduce the proposed tracking algorithm and demonstrate how the aforementioned sequential PSO framework is adopted for tracking. Our algorithm localizes the tracked object in each image frame using a rectangular window, and the motion of a tracked 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. Moreover, the fitness value of each particle is evaluated by a spatial constraint MOG (mixture of Gaussian) 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 sequential PSO based framework.

5.1. Spatial Constraint MOG Based Appearance Model

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

1) *Appearance Model*: Similar to [5], [23], 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 independently and discards the spatial layout 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

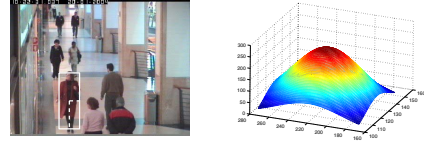


Figure 5. A 2-D gaussian spatial constraint MOG based appearance model

to the size of the target in the corresponding spatial directions, as illustrated in Fig. 5. Thus the fitness value of particles can be evaluated by the following appearance model,

$$f(x_t) = p(o_t | x_t) = \prod_{j=1}^d \left\{ N(x(j); x_c, \Sigma_c) * \sum_{l=s,w,f} \pi_{l,t}(j) N(o_t(j); \mu_{l,t}(j), \sigma_{l,t}^2(j)) \right\} \quad (16)$$

where $\{\pi_{l,t}, \mu_{l,t}, \sigma_{l,t}^2, l = s, w, f\}$ represent mixture probabilities, mixture centers and mixture variances of the S, W, F components respectively, o_t is the candidate region corresponding to state of particle x_t and d is the number of pixels inside o_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. Here, $N(x; \mu, \sigma^2)$ is a Gaussian density defined as follows.

$$N(x; \mu, \sigma^2) = (2\pi\sigma^2)^{-1/2} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} \quad (17)$$

This spatial constraint appearance model is based on the assumption that the closer the pixel to the center, the more important it is for the model representation. Fortunately, such an assumption is almost 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 previous 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, an online 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

$$m_{l,t}(j) \propto \pi_{l,t}(j) N(o_t(j); \mu_{l,t}(j), \sigma_{l,t}^2(j)) \quad (18)$$

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

Step2: The mixing probability of each component is estimated as

$$\pi_{l,t+1}(j) = \alpha m_{l,t}(j) + (1 - \alpha) \pi_{l,t}(j); l = s, w, f \quad (19)$$

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

$$M_{k,t+1}(j) = \alpha o_t^k(j) m_{s,t}(j) + (1 - \alpha) M_{k,t}(j); k = 1, 2 \quad (20)$$

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

$$\begin{aligned}\mu_{s,t+1}(j) &= \frac{M_{1,t+1}(j)}{\pi_{s,t+1}(j)}, \sigma_{s,t+1}^2 = \frac{M_{2,t+1}(j)}{\pi_{s,t+1}(j)} - \mu_{s,t+1}^2(j) \\ \mu_{w,t+1}(j) &= o_t(j), \sigma_{w,t+1}^2(j) = \sigma_{w,1}^2(j) \\ \mu_{f,t+1}(j) &= \mu_{f,1}(j), \sigma_{f,t+1}^2(j) = \sigma_{f,1}^2(j)\end{aligned}$$

5.2. 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.

Algorithm 2 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.

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

3. Update $\{p_{t+1}^i\}_{i=1}^N$ and g_{t+1} by the fitness values obtained above, and update the parameters.
4. Carry out the PSO iteration based on Equ.(1),(2);
5. Check the convergence criterion: if satisfied, continue, otherwise go to step 2;

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

6. Experiment Results

In our implementation, each candidate image corresponding to a particle is rectified to a 30×15 patch, and the feature is a 450-dimension vector of gray level values subjected to zero-mean-unit-variance normalization. All of the experiments are carried out on a CPU Pentium IV 3.2GHz PC with 512M memory.

6.1. Sequential PSO vs PF and UPF

First, we conduct a comparison experiment among the SPSO (sequential PSO) based tracking algorithm, a standard PF (particle filter) and its variation-UPF (unscented particle filter) [24] on a video with manually labeled groundtruth. Then, a theoretical investigation is presented to show why SPSO has advantages over the other two algorithms.

This video sequence¹ contains a human face moving to the left and right very quickly. Although the sequence is simple, it is effective to show the claimed advantages of the SPSO framework. In our implementation, the parameters in the particle filter and unscented particle filter are set

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

to $\{N = 600, \Sigma = \text{diag}(8^2, 8^2, 0.02^2, 0.02^2, 0.002^2, 0.002^2)\}$ corresponding to the number of particles and the covariance matrix of the transition distribution respectively. To give a convincing comparison, the sequential PSO algorithm is calibrated in the same metric, implementing with the same covariance matrix and with 60 particles in each iteration. As shown in Fig.6(a), the particle filter based tracker fails to track the object at frame 19, 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.6(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. 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 fails at frame 19 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 theoretical investigation shows the underlying reasons for the above experimental results. The undesired behavior of particle filter in Fig.6 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. 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.7), 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.7), 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.7 shows the multi-layer

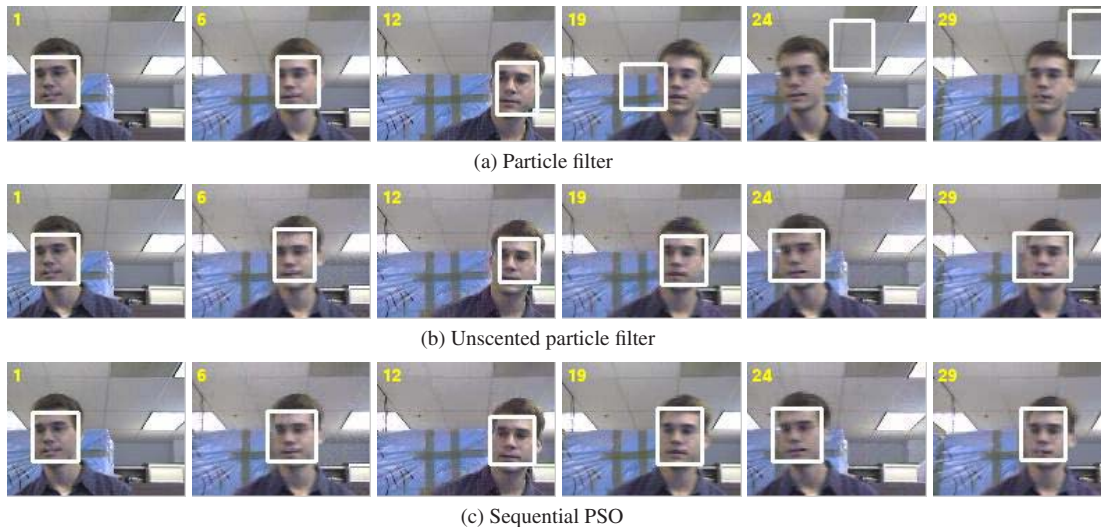


Figure 6. Tracking performances of a human face with rapid motion

Tracking Framework	Frames Tracked	MSE of Position (by pixels)
<i>PF</i>	18/31	17.069
<i>UPF</i>	31/31	6.975
<i>SPSO</i>	31/31	4.172

Table 1. Quantitative results of SPSO tracker and its comparison with PF tracker and UPF tracker

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.

6.2. 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.8(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.8(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.8(c), which demonstrates that our algorithm has the ability to track the object where large movements exist between two successive frames.

7. 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 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.

Acknowledgment

This work is partly supported by NSFC (Grant No. 60672040, 60705003) and the National 863 High-Tech R&D Program of China (Grant No. 2006AA01Z453).

References

- [1] M. Kass, A. Witkin and D. Terzopoulos, "Snakes: active contour models", *IJCV*, 1(4): 321-332, 1988. 1

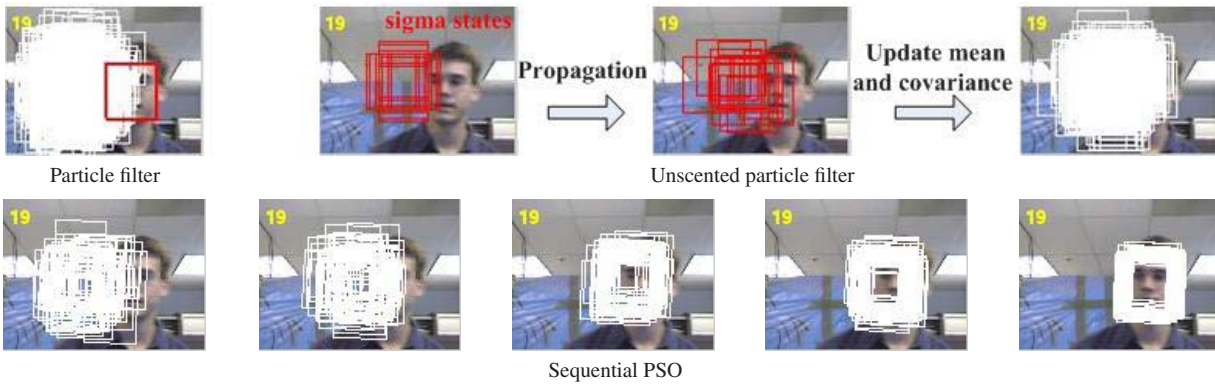


Figure 7. Tracking procedure of the frame 19

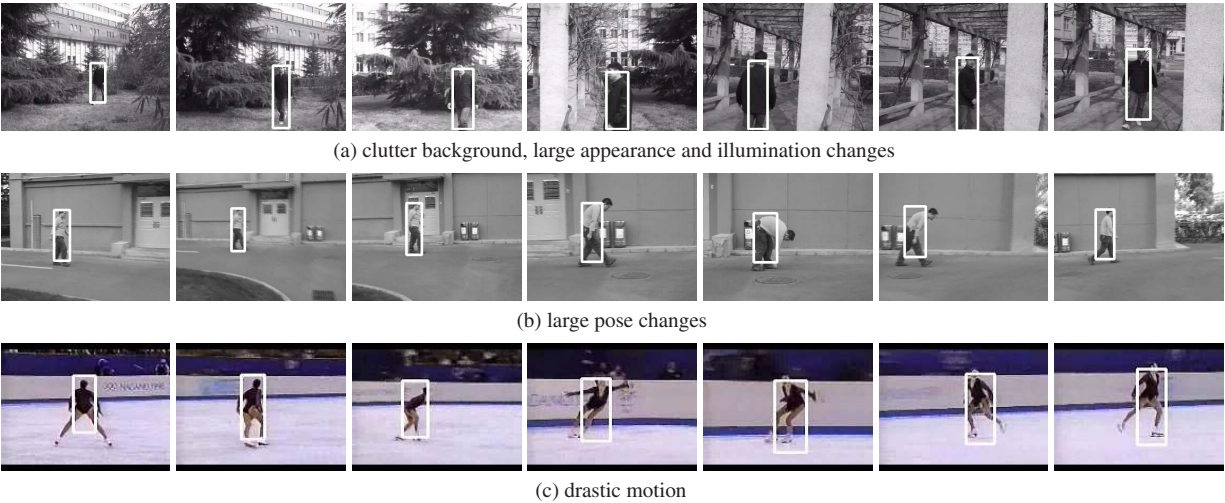


Figure 8. More experimental results

- [2] G. D. Hager and P. N. Hager, "Efficient region tracking with parametric models of geometry and illumination", *IEEE Trans. on PAMI.*, 20(10): 1025-1039, 1998. 1
- [3] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking", *IEEE Trans. on PAMI.*, 25(5): 234-240, 2003. 1
- [4] M. Isard and A. Blake, "Condensation: conditional density propagation for visual tracking", *IJCV.*, 29(1): 5-28, 1998. 1
- [5] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi, "Robust online appearance models for visual tracking", *IEEE Trans. on PAMI.*, 25(10): 1296-1311, 2003. 1, 5
- [6] C. Rasmussen and G. D. Hager, "Probabilistic Data Association Methods for Tracking Complex Visual Objects", *IEEE Trans. on PAMI.*, 23(6):560-576, 2001. 1
- [7] T. Yu and Y. Wu, "Differential Tracking based on Spatial-Appearance Model(SAM)", *Proc. CVPR'06*, pp.720-727, 2006. 1
- [8] Q. Zhao, S. Brennan and H. Tao, "Differential EMD Tracking", *Pro. ICCV'07*, 2007. 1
- [9] M. Dewan and G. D. Hager, "Toward Optimal Kernel-based Tracking", *Pro. CVPR'07*, pp. 618-625, 2007. 1
- [10] K. Fukunaga and L. Hostetler, "The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition", *IEEE Trans. on Information Theory*, 21(1): 32-40, 1975. 1
- [11] C. Yang, R. Duraiswami and L. Davis, "Fast Multiple Object Tracking via a Hierarchical Particle Filter", *Pro. ICCV'05*, pp. 212-219, 2005. 1
- [12] M. Bray, E. K. Meier, N. N. Schraudolph and L. V. Gool, "Fast Stochastic Optimization for Articulated Structure Tracking", *Image and Vision Computing*, 25(3): 352-364, 2007. 1
- [13] A. P. Leung, S. G. Gong, "Optimizing Distribution-based Matching by Random Subsampling", *Pro. CVPR'07*, pp. 1-8, 2007. 1
- [14] A. Cuzol, E. Memin, "A stochastic filter for fluid motion tracking", *Pro. ICCV'05*, pp. 396-402, 2005. 1
- [15] B. Song, A. K. Chowdhury, "Stochastic Adaptive Tracking In A Camera Network", *Pro. ICCV'07*, 2007. 1
- [16] X. Zhang, W. Hu, S. Maybank, and X. Li, "Graph Based Discriminative Learning for Robust and Efficient Object Tracking", *Pro. ICCV'07*, 2007. 1
- [17] J. Kennedy, and R. C. Eberhart, "Particle swarm optimization", in *Proc. IEEE Int'l Conf. on Neural Networks*, pp. 1942-1948, 1995. 1, 2, 3
- [18] M. Clerc, and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space", *IEEE Trans. on Evolutionary Computation*, 6(1): 58-73, 2002. 1
- [19] M. P. Wachowiak, R. Smolikova, and Y. Zheng, "An approach to multimodal biomedical image registration utilizing particle swarm optimization", *IEEE Trans. on Evolutionary Computation*, 8(3): 289-301, 2004. 1
- [20] D. Parrott and X. Li, "Locating and tracking multiple dynamic optima by a particle swarm model using speciation", *IEEE Trans. on Evolutionary Computation*, 10(4): 440-458, 2006. 1
- [21] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering", *Statistics and Computing*, 10(3): 197-208, 2000. 4
- [22] 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. 4
- [23] S. Zhou, R. Chellappa, B. Moghaddam, "Visual Tracking and Recognition Using Appearance-adaptive Models in Particles Filters", *IEEE Trans. on IP*, 13(11): 1434-1456, 2004. 5
- [24] 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. 6