2008 IEEE/RSJ International Conference on Intelligent Robots and Systems
Acropolis Convention Center
Nice, France, Sept, 22-26, 2008

Swarming Particles with Multi-feature Model for Free-Selected Object Tracking

Yuhua Zheng and Yan Meng, *Member, IEEE*

Abstract—This paper presents a new object tracking algorithm that embeds swarming particles into generic particle filter framework to achieve more robustness and flexibility. Firstly a group of particles associated with potential solutions are initialized in a high-dimensional space. Then particle swarm optimization (PSO) is used to drive particles flying. The object is tracked when the particles reach convergence. This PSO-based algorithm contains resample, similarity measure, and integration together such that the degeneracy problem of particle filter can be avoided. Furthermore, a multiple feature model is proposed for object description to enhance the tracking accuracy and efficiency. The proposed algorithm is independent with specific objects and can be used for any free-selected object tracking. Some experimental results demonstrate efficiency and robustness of the algorithm.

I. Introduction

Visual object tracking has drawn extensive attentions recently. The background model and motion detection are good for surveillance system with still backgrounds. However, in many scenarios like mobile robot navigation, both foreground and background are dynamically changed. And the appearance, position, and scale of interested object may vary with time. Therefore, it would be desirable to develop tracking algorithms which can be adaptive to complex scenes, robust with noisy images, and meanwhile capable of real-time execution.

Many algorithms apply probability-based estimations for object tracking, including Kalman filter (KF) [1] [2] and Particle Filter (PF) [3] [4] (also known as CONDENSATION [5]). Usually, there are two steps involved in the probability-based methods: predication and update, which are executed recursively to catch the object evolution. However, the conventional KF is only suitable for linear system with Gaussian noise, and the performance of PF would normally be affected by the degeneracy problem. Other methods adopt the kernel-searching idea where the tracking is taken by searching for the best match of the predefined object model [6-8]. Mean shift [9], as the most popular one, has been widely applied. It is an iterative process of shifting the predefined kernel gradually on the data surface by following the gradient.

Recently, Particle Swarm Optimization PSO [10][11] has been applied to many computer vision areas. In [12] each particle is treated as a detector with a specific scale. Then all particles scan over the IR image to find people whose size falling into one particle. Akbari et al. [13] employ both PSO and Kalman Filter in a hybrid framework. Each object is divided into non-overlapped blocks and each block is represented by a particle. Particles are guided by KF to do object tracking. A PSO-based algorithm was also proposed in [14], where particles fly over pixels directly in a predator-prey way to track the object. The PSO with mutation operator is proposed in [15] to enhance the particle filter, which avoids the impoverishment phenomenon by applying mutation to keep multiple modes of particle sets.

In this paper, we propose a new object tracking algorithm which embeds PSO into a particle filter, automatically including resample, similarity measure, and integration into one close loop. By using the high-dimensional solution space, the tracking window can be updated at position, scale, and appearance simultaneously. Furthermore, the exploration ability of PSO brings robustness to wiggling images and local maxima or minima traps. And the convergence of PSO accelerates the searching process and approaches a real-time performance with small swarm sizes less than 30.

The paper is organized as follows. Section II introduces the idea and the convergence of PSO. The connection between PSO and the particle filter is discussed in section III. Section IV explains the details of the PSO-based particle filter including object model and parameter settings. Experimental results are demonstrated and analyzed in Section V. Conclusion and further works are discussed in Section VI.

II. Particle Swarm Optimization

Proposed by Kennedy and Eberhart in 1995, PSO was inspired from the simulation of a simplified social model. Usually, potential solutions are presented as particles in a search space; and a fitness function is defined as the underlying mechanism to direct particles movements. The social metaphor that leads to PSO can be summarized as follows: the individuals that are part of a society hold an opinion that is part of a "belief space" (the search space) shared by neighboring individuals. Individuals may modify this "opinion state" based on three factors: the knowledge of

Yuhua Zheng is currently a Ph.D. student in the Department of Electrical and Computer Engineering, Stevens. Institute of Technology, Hoboken, NJ 07030 USA (email: yzheng1@stevens.edu).

Yan Meng is with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030 USA (phone: 201-216-5496; fax: 201-216-8246; e-mail: yan.meng@ stevens.edu).

the environment (inertia part); the individual's previous history of states (cognitive part); and the previous history of states of the individual's neighborhood (social part). Following certain rules of interaction, the individuals in the population adapt their scheme of belief to the ones that are more successful among their social network. Over time, a culture arises, in which the individuals hold opinions that are closely related.

Without loss of generality, the one-dimensional particle movement rules are expressed in equation (1) and (2) as:

$$v(t+1) = wv(t) + c_1(\hat{s} - x(t)) + c_2(\hat{n} - x(t)) \quad (1)$$

$$x(t+1) = x(t) + v(t+1) \quad (2)$$

Where $v(t)$ and $x(t)$ are the velocity and position of a particle at time t, respectively; $v(t+1)$ and $x(t+1)$ for time $t+1$, respectively. $\hat{s}$ is the best position that the particle has visited. $\hat{n}$ is the best position hold by its neighbors. w, c_1 and c_2 are weights for inertia, cognitive and social trends. Equation (1) can be further simplified into (3) by adopting notations in (4).

$$v(t+1) = w \cdot v(t) + c \cdot (\hat{p} - x(t)) \quad (3)$$

$$c = \frac{c_1 + c_2}{2} \qquad \hat{p} = \frac{(c_1 \times \hat{s} + c_2 \times \hat{n})}{(c_1 + c_2)} \quad (4)$$

By combining above equations, the PSO can be rewritten in a matrix form as:

$$\begin{bmatrix} x(t+1) \\ v(t+1) \end{bmatrix} = \begin{bmatrix} 1-c & w \\ -c & w \end{bmatrix} \cdot \begin{bmatrix} x(t) \\ v(t) \end{bmatrix} + \begin{bmatrix} c \\ c \end{bmatrix} \cdot \hat{p} \quad (5)$$

Then we apply the dynamic system theory and the eigenvalue analysis to equation (5). It appears that the equilibrium state will be achieved as equation (6) if conditions of equation (7) are satisfied.

$$\begin{bmatrix} x_{eq} \\ v_{eq} \end{bmatrix} = \begin{bmatrix} \hat{p} \\ 0 \end{bmatrix} \quad (6)$$

$$w < 1 \;\; c > 0 \quad 2w - c + 2 > 0 \quad (7)$$

The conclusion in (6) is intuitively correct. Firstly, particles are scattered over the space; then particle move around and finally are attracted by equilibrium points with velocities down to zero. The equilibrium points not only rely on their own visited paths but also on the social best.

III. THE PARTICLE FILTER

A. The Particle Filter Method

By modeling the tracking problem as estimating the posterior distribution, Particle filter recursively implements Bayesian filter by Monte Carlo simulations which approximates the real distribution by integrating a set of random samples with associated weights. Suppose $\{x_{0:k}^i, w_k^i\}_{i=1}^N$ denotes the set of support points with associated weights and the weights are normalized such that $\sum_{i=1}^N w_k^i = 1$. Thus, the posterior distribution at k can be approximated as.

$$p(x_{0:k} | z_{1:k}) \approx \sum_{i=1}^N w_k^i \delta(x_{0:k} - x_{0:k}^i) \quad (8)$$

Where $z_{1:k}$ presents measurement or observation up to time k and δ is the Dirac delta function. When $N \to \infty$, the approximation converges [16].

In equation (8), samples $\{x_{0:k}^i, w_k^i\}_{i=1}^N$ should be drawn from $p(x_{0:k} | z_{1:k})$, which is unknown and need to be estimated. Then by assuming the system propagation as one-step Markov process and introducing the importation density, the posterior distribution can be approximated as equation (9) with weights in (10). $p(z_k | x_k^i)$ is usually called likelihood density. More details of particle filter can be referred to [17].

$$p(x_k | z_{1:k}) \approx \sum_{i=1}^N w_k^i \delta(x_k - x_k^i) \quad (9)$$

$$w_k^i \propto w_{k-1}^i p(z_k | x_k^i) \quad (10)$$

B. The connection between PSO and PF

The particle filter discussed above is generally called the sequential importance sampling (SIS) algorithm, which has a common problem: the degeneracy phenomenon. Nearly most particles will only have negligible weights after a few iterations. In other words, considerable computation would be spent on updating particles that have almost zero contribution to the approximation of the posterior distribution. To solve this problem, many resample methods have been proposed to concentrate on important particles and eliminate those weak ones. After resample, weights are reset to $w_k^i = \frac{1}{N}$, and at next time step, weights of sampling points only depend on their likelihoods as:

$$w_{k+1}^i \propto \frac{1}{N} p(z_{k+1} | x_{k+1}^i) \propto p(z_{k+1} | x_{k+1}^i) \quad (11)$$

For most resample algorithms, the particles with high weights are statistically selected many times, which leads to many repeated points. This loss of diversity of particles is called the sample impoverishment problem.

Therefore, the PSO is introduced here as a dynamic sampling and evolving algorithm. Instead of generating sample points by certain functions, let particles keep in motion after initialization and adjust their locations according to the PSO rules. After iterations, particles will be attracted by high-likelihood points which have higher weights according to equation (11). The integration of these higher-weights particles by (9) can approximate the posterior distribution.

This PSO-based filter has the following characteristics:

--Flexibility. There is no need to predict or define a resample function. Once all particles are initialized, they will automatically search and distribute themselves to match the real density.

--Ability of exploration and exploitation. Particles are dynamically in motion. The inertia trend keeps particles to explore new areas while the cognitive and social trends try to keep them exploiting around the visited points.

--Simple. PSO rules as equations (1) and (2) are easy to implement. If adopting global networking with a single best, the algorithm can be executed of O(N), where N is the population of particles.

--Open framework. The proposed algorithm can be combined with different likelihood functions, initialization methods and so on.

IV. The PSO-based Particle Filter

A. The Algorithm Outline

Fig.1 shows the process of the proposed algorithm. Firstly the solution space is constructed and particles are initialized. Then Particles move around based on PSO principles, and search for points of high fitness values. After the evolution stops, the object profile and its fitness value can be updated and referred to next tracking circle adaptively.

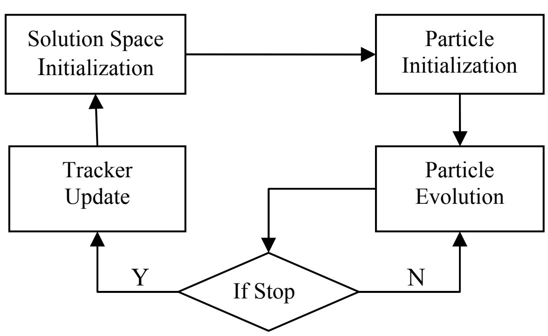

Fig.1: the flowchart of the PSO-PF for object tracking.

B. The Solution Space for Object Tracking

For visual object tracking, generally, the object is presented by a tracking window associated with its location and size. The object profile can be represented as

$$o_k^i = \{x_k^i, y_k^i, l_k^i, d_k^i\}, \tag{12}$$

Where (x_k^i, y_k^i) represents the central point of the tracking window; l_k^i, d_k^i represents the length and width of the tracking window, respectively. Thus from the view of PSO, a four-dimensional solution space is constructed. Every swarming particle of this space actually presents a tracking window. With movements of particles, tracking windows with varied locations and sizes are generated, tested, compared and eliminated.

C. Particle Initialization

Generally, the larger population means more accurate search. However, the computation cost increases as well. Because particles will dynamically move around, the PSO algorithm is capable of finding solutions with a small size of swarm. Based on experimental results, the population of 30 is good enough for most cases.

Considering the connection between two consecutive frames, it is reasonable to assume that the object motion is continuous. Thus particles are evenly distributed around the previous object along all dimensions in the initial space defined in (12). For tested experiments with images of 320 x 240 pixels, the locations of new particles can move up and down to 25 pixels from the center, and the sizes of the windows can be shrunk and extended up to 20 percent.

Therefore, after initialization, particles can be represented by the following equation at time k

$$P = \{p_k^i \big| o_k^i, w_k^i, f_k^i, v_k^i, s_k^i, n_k^i \quad i = 1,2,...,N\} \tag{13}$$

Each particle is associated with 6 parameters. The first three are for generic particle filter including an object profile o_k^i, a weight w_k^i, and a fitness value f_k^i. The others are for PSO rules, including a velocity v_k^i, a self memory s_k^i, and a neighbor best n_k^i. The velocity v_k^i is a randomly assigned vector, which has the same dimensions with the object profile o_k^i.

D. Particle Evolution

According to PSO principles, a particle's movements depend on its neighbor network and the parameter weights of PSO. For object tracking, a global society network is expected to work better than the local neighborhood, which means all members in the group share only one global best. Because object tracking generally is a one-to-one projection from frame to frame, a single global best is capable of clustering faster. For multiple object tracking, every object can be associated with one independent PSO-PF individually.

The convergence of a particle is guaranteed if the conditions in equation (7) are hold. In the view of convergence, the inertia weight will only affect the convergence time and a particle's trajectory. The cognitive and the social weights can be equal if a particle trusts its own experience as much as the whole group. In tested experiments, the parameter set of $\{w, c_1, c_2\} = \{0.5,1,1\}$ shows good results.

E. Termination Conditions

From initialization to convergence, computation loops taken by PSO may vary for different video sequences. The convergence can be recognized by identifying velocities of all particles becoming zero. Or the tracking can be stopped if the global best cannot be improved any more. However, to ensure the real-time performance, the maximum of computation iterations are used as the termination condition. For most tested experiments, a good result can be approached within 10 iterations.

F. Object Update

When the searching procedure stops, every particle holds a tracker o_k^i and its associated fitness value f_k^i, which is also the likelihood. Their weights w_k^i can be calculated by equation (11). By integrating equation (9), the updated object can be represented as (14). Then the new object profile can be used to initialize particles for next image.

$$O_k = \frac{1}{N}\sum_{i=1}^{N} f_k^i o_k^i \tag{14}$$

G. Multi-Feature Model as Fitness Function

The likelihood function or fitness function is critical for the PSO-based particle filter, which influences the individual best, the neighbor best, and the contribution weight for the final integration. The fitness function of a good feature model will enhance the accuracy and the robustness of PSO.

To handle the uncertainty caused by dynamic environments, a multi-feature object model is applied for constructing the fitness function. Firstly, both colors and local statistical features are used to achieve more robustness. Secondly, the accumulative histogram is proposed to catch appearance changes over time to achieve more accuracy.

1) Color Feature: Firstly, images are analyzed in HSV color space. Then hue values over all pixels will be collected to generate a histogram. Instead of using a single frame, all object appearance over image sequences will be adopted for building an accumulative histogram. The basic idea is to separate stable color components from noisy ones; and give them more weights, just like increasing the ratio of signal over noise.

Let $c_k(j)$ represents the credit for color j with $j \in [0,360]$ at time k. It is set to be one if color j is included by the object appearance at this moment, otherwise, it is zero.

$$c_k(j) = \begin{cases} 1 & color\ j\ included \\ 0 & otherwise \end{cases} \tag{15}$$

All color credits are added over time, and the accumulative credit of color j is defined as:

$$AC_k(j) = \frac{\sum_{t=1}^{k} c_t(j)}{k} \tag{16}$$

The more frequently a color appears the greater credit it has. Furthermore, this credit can be normalized as:

$$AC_k(j) = \frac{AC_k(j)}{\sum_{j=0}^{360} AC_k(j)} \tag{17}$$

With the accumulative color credit, the distance between the object histogram and a particle histogram is defined as

$$D(H_k^i, H_{k-1}) = 1 - \sum_{j=0}^{360} AC_k(j)\sqrt{n_k^i(j)\cdot n_{k-1}(j)} \tag{18}$$

Where H_k^i represents the histogram of particle i at time k. H_{k-1} represents the object histogram. $n_k^i(j)$ and $n_{k-1}(j)$ are normalized pixel numbers of color j for the particle and the object, respectively. $AC_k(j)$ is the normalized credit. Equation (18) is actually a weighted Bhattacharyya Coefficient calculation.

2) Local Statistic Features: Although color is a powerful feature, it will be affected heavily by illumination changes like lightening condition, shadow or even camera characters. In order to strength the object model, some local statistic features are also added. Many local features have been proposed for object description; including Scale-invariant feature transform (SIFT), and histograms of oriented gradients (HOG) [18] [19]. However, in order to meet the real-time requirement, Haar histograms are used here [20]. A set of templates are applied to filter image patches, and results will reflect the spatial information related with edges, textures and structures. And rapid computation methods are also available to execute these templates very quickly. After Haar histograms are constructed, the distance can also be computed in the similar way with color histograms.

3) Feature fusion: Let dc_k^i notates the distance of color histograms of particle i at time k given by equation (18). Similarly dh_k^i is used to represent the distance of Haar histograms. Since both histograms are independent with each other, the fitness function of the multi-feature model can be defined as (19), where α is the adaptive weight to adjust between color and Haar feature.

$$f_k^i = \alpha \cdot dc_k^i + (1-\alpha)\cdot dh_k^i \tag{19}$$

V. Experimental Results

The proposed PSO-based particle filter (PSO-PF) is implemented by C++ and runs on Pentium4 PC with Windows platform. Several video clips are applied to test the efficiency and performance of the algorithm.

Fig.2 provides the procedure of how particles move under PSO principles and eventually converges in the solution space. To simplify the display, the four-dimensional solution space is reduced into a 3D space. The horizontal plane represents the image plane. The vertical axis represents different sizes of potential windows. Particles are represented as dots in the space. This example helps to demonstrate how fast the convergence of PSO could be. A group of twenty-seven particles flying in 320 x 240 images, where only five iterations are needed to achieve the convergence.

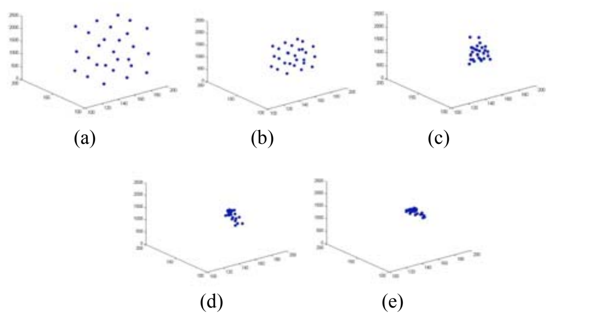

Fig.2. The convergence of PSO in solution space. Each point presents a particle and a global network is applied here. At the beginning, particles are evenly distributed, shown in (a). Then they start to move, shown as (b), (c) and (d). After 5 loops, particles cluster around the found global best as (e).

To evaluate the robustness of the PSO-PF tracking method under various environments, another experiment is conducted in an outdoor environment, as shown in Fig. 3. The upper body of a reporter is taken as the object. The proposed algorithm can catch the object when the reporter is moving with a dynamic background. In the hue images, the pattern of the object is not isolated clearly from the background, which makes the tracking more difficult. The proposed method is always looking for the best match in terms of the fitness function. In Fig. 3(a1) and (a2), the selected pattern includes a small dark region of the head and another lighter part of the body. This pattern is kept and tracked well in Fig. 3(b1) and (b2). In Fig. 3(c1) and (c2), the head part is lost in the background. However, the proposed algorithm is still capable of tracking the object by focusing on the other valid pattern, which is the upper body. So the tracking window moves down, locks on the reporter, and keeps tracking as shown in Fig. 3(d1) and (d2). This example shows the adaptive ability of the proposed algorithm, which ensures the robust tracking performance even when the object pattern is partially lost.

To evaluate the robustness of the proposed PSO-PF method, another experiment with occlusions has been conducted. To compare the performance with other methods, a general particle filter method and a general mean-shift method have also been implemented. As shown in Fig. 4, one student's face is defined as the tracking object. As can be seen from Fig. 4(a1) to (b1), the tracked object is getting closer and starts to be occluded by another student. When the object reappears, it attracts the tracking window back as shown in Fig. 4(c1). The object tracking is recovered and continued in Fig. 4(d1). Here the proposed PSO-PF algorithm shows its higher robustness under occlusions.

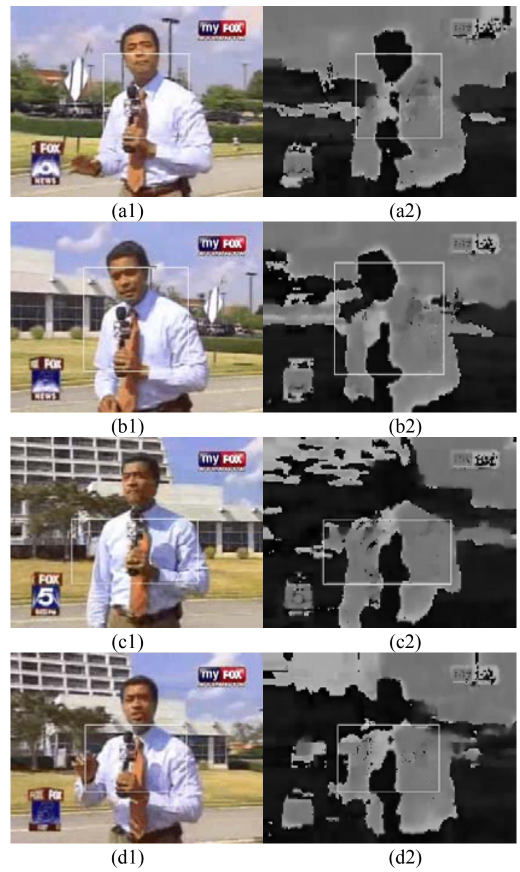

Fig.3. An outdoor tracking video clip. During the tracking process, the proposed algorithm is trying to find the best match of the object pattern.

Compared with the general particle filter method, as shown in Fig. 4(a1), (a2) and Fig. 4(b1), (b2), both PSO-PF and the general PF are capable of tracking the object without occlusion. However, when the occlusion occurs, the general PF is attracted by the cluttered background and cannot recover from the occlusion, as shown in Fig. 4 (c2) and (d2). When the face in track is occluded by another face, the tracking window using the general PF method drifts to another face, instead of the original tracking face. This is reasonable because another face has similar appearance with the face in track. However, when the face in track reappears, the general particle filter cannot explore widely enough to catch up the reappeared tracking object. So it still sticks to the other face and loses the interested object. On the other side, the PSO-PF algorithm is robust with occlusion and capable of re-catching the object, as shown in Fig. 4 (c1) and (d1). The reason for this robustness is because when the object reappears, the particles start searching around in a wider area with different window sizes and locations and can easily detect the tracked object. Since the face in track has the highest fitness value compared to another face, it will attract more particles, which leads to the object being tracked correctly.

Then PSO-PF is also compared with the mean shift method. The 'camshift' program from OpenCV is adopted here as the mean-shift program. The proposed PSO-PF algorithm keeps tracking the object from Fig. 4(b1) to (d1), while the mean-shift method gradually shifts off from the object and loses it eventually as shown in Fig. 4(b3) to (d3). The exploring capability of particles supplies more flexible adjustment in a high-dimensional space, which leads to more accurate size of the tracking window.

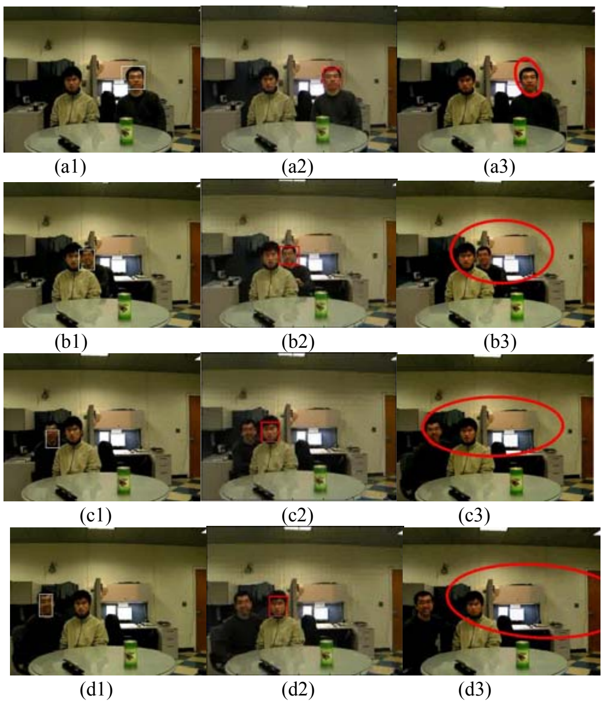

Fig.4. An indoor tracking video experiments using PSO-PF, general PF and mean shift methods. First column (a1)(b1)(c1)(d1) shows the results of the proposed PSO-PF method, the second column (a2)(b2)(c2)(d2) for a general PF method, and the last column (a3)(b3)(c3)(d3) for the mean shift method. From (a1) to (d1), the PSO-PF is capable of tracking the object with short occlusion. However the general PF lost the object when the object is occluded by cluttered backgrounds, as shown in (a2) and (d2). The mean shift lost the object at the very early stage and keeps drifting away in the whole process.

In the experiments, the PSO-PF demonstrates its adaptive capability and robustness for the object tracking under dynamic indoor and outdoor environments. This virtue comes from the flexibility and the sharing mechanism of the PSO algorithm, which allows individuals to explore new areas as well as keeps the society tight. The initial wide distribution of particles naturally makes this algorithm to handle jitters between image frames. Finally, this convergence is simple and quick to let the algorithm achieve real-time performance.

VI. CONCLUSION

In this paper, a PSO-based particle filter is proposed for the object tracking under dynamic environments. The basic mechanism of the PSO algorithm is introduced. The fusion of PSO and particle filter is explained in details. To apply PSO-based particle filter for object tracking, the construction of solution space, parameter settings and fitness function based on a multi-feature model are discussed. The experimental results show the converging process, the efficiency, and robustness of the proposed method under dynamic environments.

REFERENCES

[1] E. Cuevas, D. Zaldivar, and R. Rojas, "Kalman filter for vision tracking", *Technical Report B 05-12*, August, 2005.

[2] Y. Huang, T.S. Huang, and H. Niemann, "Segmentation-based Object Tracking Using Image Warping and Kalman Filtering", *IEEE ICIP*, Vol. 3, pp. 601-604, 2002.

[3] Ch. Yang, R. Duraiswami and L. Davis, "Fast Multiple Object Tracking via a Hierarchical Particle Filter", *ICCV*, 2005.

[4] J. Uk Cho, etc. "A Real-Time Object Tracking System Using a Particle Filter", *International Conference on Intelligent Robotics and Systems*, pp. 2822-2827, Beijing, China, 2006.

[5] M. Isard, *Visual Motion Analysis by Probabilistic Propagation of Conditional Density*, D.Phil. Thesis, Oxford University, 1998.

[6] C. Beleznai, B. Fruhstuck, and H. Bischof, "Human Tracking By Mode Seeking," in *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis* (2005),pp.1-6, Sept. 2005.

[7] B. Han, etc., "Incremental Density Approximation and Kernel-Based Bayesian Filtering for Object Tracking", *IEEE CVPR*, 2004.

[8] W. Qu and D. Schonfeld, "Robust Kernel-Based Tracking Using Optimal Control", *IEEE, ICIP*, 2006.

[9] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking", *IEEE Trans. Pattern Anal. Mach. Intel*., 25(5): 546-577, 2003.

[10] J. Kennedy, R. C. Eberhart, and Y. Shi, *Swarm intelligence*, San Francisco, Morgan Kaufmann Publishers, 2001.

[11] R. C. Eberhart and Y. Shi, "Particle Swarm Optimization: Developments, Applications, and Resources," *in Proc. Congress on Evolutionary Computation 2001 IEEE service center*, Piscataway, NJ., Seoul, Korea., 2001.

[12] Y. Owechko, S. Medasani, and N. Srinivasa, "Classifier Swarms for Human Detection in Infrared Imagery," *in 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04)*, Vol. 8, pp. 121, 2004.

[13] R. Akbari, M. D. Jazi, and M. Palhang, "A Hybrid Method for Robust Multiple Objects Tracking in Cluttered Background", Vol. 1, 24-28, *ICTTA '06*.

[14] L. Anton-Canalis, M. Hernandez-Tejera, and E. Sanchez-Nielsen etc, "Particle Swarms as Video Sequence Inhabitants For Object Tracking in Computer Vision", Vol. 2, pp. 604-609, *ISDA '06*.

[15] Qicong Wang, Li Xie, Jilin Liu, and Zhiyu Xiang, "Enhancing Particle Swarm Optimization Based Particle Filter Tracker", ICIC 2006, pp. 1216-1221

[16] A. Doucet, "On sequential simulation-based methods for Bayesian filtering", Eng. Dept., Cambridge Univ., Cambridge, U.K., *Tech. Rep. CUED/F-INFENG/TR 310*, 1998.

[17] S. Arulampalam, etc. "A tutorial on Particle filters for on-line non-linear/non-Gaussian Bayesian tracking", *IEEE Transactions on Signal Processing*, Vol. 50(2), pages 174-188, February 2002.

[18] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection". *In Proc. CVPR*, volume 1, pp.886-893, 2005.

[19] F. Han, etc. "A two-stage approach to people and vehicle detection with HOG-based SVM", *The 2006 Performance Metrics for Intelligent Systems Workshop*, pp. 133-140.

[20] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features". *CVPR 2001*, Vol. 1.