Swarm Intelligence Based Dynamic Object Tracking

Yuhua Zheng and Yan Meng, *IEEE, Member*

***Abstract*—This paper presents a new object tracking algorithm by using the particle swarm optimization (PSO), which is a bio-inspired population-based searching algorithm. Firstly the potential solutions of the problem are projected into a state space called solution space where every point in the space presents a potential solution. Then a group of particles are initialized and start searching in this solution space. The swarm particles search for the best solution within this solution space using the Particle Swarm Optimization (PSO) algorithm. An accumulative histogram of the object appearance is applied to build up the fitness function for the interested object pattern. Eventually the swarming particles driven by the fitness function converge to the optimal solution. Experimental results demonstrate that the proposed PSO method is efficient and robust in visual object tracking under dynamic environments.**

I. INTRODUCTION

VISUAL object tracking, as a key problem in computer vision, has wide applications including surveillance system, public security system, service robotics, human-machine interfaces and so on. Unlike the pure object detection or classification, visual tracking problems mainly focus on tracking mobile object like vehicles or walking people. However, in many scenarios where the camera is moving, both foreground and background are dynamically changing due to the motion of camera, ever changing illumination condition, or occlusions. And all these factors could bring more challenges to produce flexible and robust tracking algorithms. The appearance, position, and the scale of the object may vary with the background changes. Therefore, tracking algorithms under such situations should be adaptive with complex scenes, robust with noisy images, and capable of real-time execution.

Visual object tracking has attracted extensive attention in past decades. First main category of the tracking methods, including Kalman filter (KF) [1] [2] [3] [4] [5] and Particle Filter (PF) [6] [7] [8] [9] (also known as CONDENSATION [10]), applies dynamic system principles into tracking. Generally two steps including the predication and the update are executed recursively to catch the object evolution. Another popular category adopts the template-matching idea where the tracking is taken by searching for the match of the predefined object model. Many object models and matching strategies are proposed for various applications [11] [12] [13] [14] [15] [16] [17] [18]. Among these methods, mean shift [15], as a well-known kernel-based local searching algorithm, has been widely applied for object tracking. It is an iterative process of shifting the predefined kernel gradually on the data surface by following the gradient.

In this paper, a new swarm intelligence based optimization algorithm, Particle Swarm Optimization (PSO) [19] [20], is applied for object tracking to achieve more efficient and robust matching process, especially under dynamic environments. Some works have been proposed using PSO-based methods on image processing and object tracking. In [21], the authors used PSO to detect people in IR images, where every particle was treated as a detector with a specific scale. Then all particles scanned over the image to find people whose size falling into one particle. Akbari et al. [22] employed both the PSO algorithm and Kalman Filter in a hybrid framework of region and object tracking. 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 proposed in [23] to drive particles flying over image pixels directly, where object tracking emerged from interaction between particles and their environment. Similarly the algorithm called Swarmtrack in [24] is proposed by adopting a prey-predator scheme for object tracking.

Particle swarm optimization (PSO) is a bio-inspired searching algorithm, which introduces the communication mechanism of birds and fishes to accelerate the convergence in the potential solution space, so that the optimal solution can be found in a short time. The proposed algorithm treats the object tracking as a searching problem in a high-dimension solution space; then packs object features into the fitness function to drive PSO evolution. The basic idea is that a swarm of particles fly around the image and try to find the best-fit tracking window. When some particles successfully detect the interested objects, they will share this information with their neighbors. Each particle makes its own decision not only based on its neighbors, but also on its own cognition, which provides the flexibility and ability of exploring new areas. This

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).

978-1-4244-1823-7/08/$25.00©2008 IEEE

decision-making procedure can efficiently prevent the local optimum effect. With the convergence of particles, the tracking problem can be approached. To handle the uncertainty caused by dynamic environments, a normalized-accumulative histogram is proposed to generate the fitness function of the particles. This histogram tries to catch the illumination changes along object motion to provide more robust descriptions of image contents. This accumulative histogram based fitness function of the particles can help the PSO algorithm to construct a more accurate solution space and conduct more efficient searching.

Compared with other available object tracking methods, one virtue of the proposed algorithm is the accommodation of scenario changes. By using the multidimensional solution space, the tracking window is being automatically updated on position, scale and appearance simultaneously. Secondly, the exploration ability of particles ensures the algorithm to be robust to wiggling images and local maxima or minima traps. Furthermore, the convergence of the algorithm accelerates the searching process and approaches a real-time tracking performance by using small size of particles, for example, usually less than 30.

The paper is organized as follows. Section II introduces the idea and the convergence character behind PSO algorithm. Section III discusses the details of PSO-based algorithm for object tracking, including object model and parameter setting. Experimental results are demonstrated and analyzed in Section IV. Conclusion and further works in section V will be the last.

II. Particle Swarm Optimization

A. The Basic Idea of PSO

Proposed by Kennedy and Eberhart in 1995 [25] [26], PSO comes from the simulation of a simplified social model, which obviously has its root in artificial life in general, and in swarming theory in particular. As a population-based method, PSO approaches the convenient answer to a problem by driving multiple potential solutions evolving and converging. Usually, potential solutions are presented by particles in a virtual search space; and a fitness function is defined as the underlying mechanism to direct particles' movement. 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 trend of keeping its own way (inertia part)
- The individual's previous history of states (individual part)
- The previous history of states of the individual's neighborhood (social part)

An individual's neighborhood may be defined in several ways, configuring somehow the "social network" of the individuals. 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.

In the PSO algorithm, each individual is called a "particle", and is subject to a movement in a multidimensional space that represents the belief space. Particles have memory, thus retaining part of their previous states. There is no restriction for particles to share the same point in belief space, but in any case their individuality is preserved. Each particle's movement is the composition of an initial random velocity and two randomly weighted influences: individuality, the tendency to return to the particle's best previous position, and sociality, the tendency to move towards the neighborhood's best previous position.

Abstractly in the solution space, the velocity and position of the particle at any iteration is updated based on the following equations:

$$v_{id}^{t+1} = w \cdot v_{id}^{t} + c_1 \cdot \varphi_1 \cdot (p_{id}^{t} - x_{id}^{t}) + c_2 \cdot \varphi_2 \cdot (p_{gd}^{t} - x_{id}^{t}) \quad (1)$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \quad (2)$$

where v_{id}^{t} is the component in dimension d of the ith particle velocity in iteration t, and x_{id}^{t} is the component in dimension d of the ith particle position at iteration t. c_1, c_2 are constant weight factors. p_{id}^{t} is the best position achieved by particle i, and p_{gd}^{t} is the best position found by the neighbors of particle i. φ_1, φ_2 are random factors in the (0,1) interval, which have different values on each dimension of the state space. And w is the inertia weight. The PSO requires the tuning of some parameters: the cognitive and sociality weights c_1, c_2, and the inertia factor w.

According to (1), each particle adjusts its velocity by combing three forces: keeping the velocity of last moment, moving to the best position from its own memory, and moving to the best position found by its neighbors. Different parameters in (1) provide varied balance among those three factors. Then defined by (2) a particle moves in the search space according to the combined velocity to achieve a new position, which presents a new potential solution.

B. Convergence of PSO

The PSO behaviors under different conditions have been discussed extensively [27]. Here only a brief analysis is given referred to other works. From (1) and (2), it appears that each dimension is updated independently. All dimensions are only connected through the objective function, also known as the fitness function. Thus without loss of generality, the one-dimensional particle movements can be used for analyzing purpose. So equations (1) and (2) can be rewritten as:

$$v(t+1) = wv(t) + c_1(\hat{s} - x(t)) + c_2(\hat{n} - x(t)) \tag{3}$$
$$x(t+1) = x(t) + v(t+1) \tag{4}$$

where $v(t)$ and $x(t)$ are the velocity and position of the particle at time t, respectively; $v(t+1)$ and $x(t+1)$ for time $t+1$, respectively. $\hat{s}$ is the best position that this 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. Here the random factors φ_1, φ_2 are taken out so that the (3) and (4) actually present a deterministic version of the PSO algorithm, which then can be qualitative discussed. Make notations as:

$$c = \frac{c_1 + c_2}{2} \tag{5}$$
$$\hat{p} = \frac{(c_1 \times \hat{s} + c_2 \times \hat{n})}{(c_1 + c_2)} \tag{6}$$

Then equation (3) is simplified as:

$$v(t+1) = w \cdot v(t) + c \cdot (\hat{p} - x(t)) \tag{7}$$

Combining equations (4) (5) (6) and (7), the PSO can be expressed in matrix form as:

$$\begin{bmatrix} x(t+1) \\ v(t+1) \end{bmatrix} = \begin{bmatrix} 1-c & w \\ -c & w \end{bmatrix} \cdot \begin{bmatrix} x(t) \\ v(t) \end{bmatrix} + \begin{bmatrix} c \\ c \end{bmatrix} \cdot \hat{p} \tag{8}$$

Apply the dynamic system theory and the eigenvalue analysis to equation (8). It appears that the equilibrium state will be achieved as equation (9) if conditions of equation (10) are hold.

$$\begin{bmatrix} x_{eq} \\ v_{eq} \end{bmatrix} = \begin{bmatrix} \hat{p} \\ 0 \end{bmatrix}$$
$$w < 1 \quad c > 0 \quad 2w - c + 2 > 0 \tag{9}$$

The conclusion in (9) is intuitively correct. At the beginning particles are scattered over the space; then move around and finally are attracted by equilibrium points, which means the algorithm converges. The equilibrium points rely on not only their own visited paths but also the social best, as shown in equation (6). Once approaching convergence, velocities of particles become zeros.

III. THE PSO-BASED TRACKING APPROACH

A. General Idea

As a searching algorithm, PSO can be applied for object tracking to find out the best match of the predefined object in current scene. Firstly according to object model, the potential solution space is generated. This multi-dimensional solution space usually represents the interested parameters of the tracking window; and different point in the space stands for different values. For object tracking, the location and the size of the tracking window might be the most concerned factors. And each point in the solution space is also associated with a fitness value which indicates how good the current point is. The good fitness function should be able to distinguish the object from noisy background easily and accelerate the PSO searching.

Once the solution space is generated, particles are initialized to cover the possible solution region; then move around by following the rules in (1) and (2). Being driven by the fitness function, the optima in the solution space would attract more and more particles to converge. After convergence, the new tracking window can be presented according to these optimal points. Then based on the appearance of the new tracked object, the values of feature model, the accumulative histogram here with equation (10) and (11), will be updated for next frame.

B. Solution Space of the PSO Algorithm

To identify an object in an image, usually rectangle windows are utilized. Four parameters will be identified to describe the rectangle windows, including 2D location of the central point, width and height of the rectangle, as shown in Fig.1. These parameters can be used to build up a four-dimensional search space. In such a space, each particle actually presents a search window with specific location and size as:

$$P = \{p_i \mid p_i(x_i, y_i, l_i, w_i), i = 1,2,...,N\}$$

where (x_i, y_i) represent the central point of the rectangle related to particle i; l_i and w_i represents the length and width related to particle i, respectively; and N is the population of swarm particles. Each individual particle has different values of these parameters. In other words, they are distributed in a four-dimensional search space.

Theoretically, the particles should be distributed all over the four-dimensional solution space. However, the search would take much longer if the solution space is very large. Here, we apply the motion-based constraints to diminish the search area. A straightforward constraint is the movement continuity of the tracked object since it is reasonable to assume that the object motion is continuous under most tracking situations. In other words, the tracking window of a new frame should be close or adjacent to the window of the previous one on both location and size.

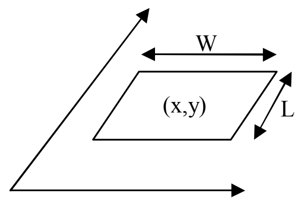


Fig.1: The four parameters associated with a particle window.

After being initialized, particles move around within the solution space, share information and try to converge to the optima. According to the convergence characteristic that is discussed before, particles' trajectories depend on both the fitness function and the control weights. The fitness function actually decides the distribution of the solution space; and the control weights specifically constrain particle motions.

C. Setting PSO Parameters

Due to the mechanism of sharing information, the PSO algorithm is capable of finding solutions with a small size of particles. Generally, a group of 10 to 30 particles would be enough for most problems. As we discussed before, the initialization of the particles are based on the assumption of the movement continuity of the tracked object. So the particles on the new frame are evenly distributed around the previous window. For images with 320 x 240, the locations of the new particles can move up and down up to 25 pixels from the center, and the sizes of the windows can be shrunk and extended up to 20 percent.

From the initialization to the final convergence, the PSO algorithm needs several iterations to catch the object. The stop conditions of the iterations include the maximum number of iterations is reached, the fitness value is within a predefined threshold value, or no more improvement can be observed. These conditions can be used separately or combined together. For most tested experiments, a good result can be approached after about 6 or 7 iterations, which makes it possible to achieve a real-time performance.

For object tracking, a global society network is expected to work better than the local neighborhood. Because global network means all members in the group share only one global best which conducts a faster clustering. The inertia factor is set at an initial value at the beginning to explore new solutions; then it would be reduced to ensure convergence. The weight of cognitive factor decreases slowly to prevent the particles being trapped into a local maximum.

When the search is finished, it is difficult to ensure that all particles converge to the exactly same point in the solution space. Sometimes, some particles may be trapped on some local maxima. However, most particles are clustered closely around the global optima. To get a reasonable solution, all the central points of the qualified particles will be calculated for the new tracking window. A particle is qualified if its fitness value is beyond the preset threshold which actually ensures the associated window is quiet close to the optimum.

D. Fitness Function

Using PSO, the feature model decides the fitness value of a point in the solution space, which tells how good this point is. A good feature model can accelerate and enhance the searching of particles. In this paper, color is adopted and an accumulative histogram is applied to build the fitness function.

First, the images need to be transformed from RGB to HSV color space. Then the hue values over all pixels will be collected to generate a histogram. The basic idea is to separate stable components of hue values from the noisy ones, and give more weights on the stable bars. Let $c(i,t)$ represents the credit for color i with $i \in [0,360]$ at time t, which is set to be one if color i shows as a part of the object, otherwise zero.

$$c(i,t) = \begin{cases} 1 & \textit{if the color with index i appears at time t} \\ 0 & \textit{otherwise} \end{cases} \tag{10}$$

Then this color credit is applied for the next image at time t+1, where the value of color i in the histogram can be defined as:

$$H_i(t+1) = \Big(1 + \frac{\sum_{j=1}^{t} c(i,j)}{t}\Big) v_i \tag{11}$$

where v_i is the total number of pixels whose hue value equals to i; and the part in parenthesis is called time factor. Introducing the time factor actually accumulates the appearance of a specific color. The more frequently a color appears, the greater weight it has.

When the PSO-based searching algorithm is running, each particle at every moment covers a region that is associated with a histogram. The best matched candidates for tracking can be obtained by comparing their histograms with the target histogram. Therefore, a method to measure the distance between two histograms is required.

The Bhattacharyya Coefficient [28] is used to measure the similarity between these two histograms as:

$$BC(H(t), \hat{H}(t-1)) = \sum_{i \in n} \sqrt{H_i(t), \hat{H}_i(t-1)} \tag{12}$$

where $H(t)$ represents the histogram of a particle, $\hat{H}(t-1)$ represents the histogram of the target at time t-1, and n denotes the range. $H_i(t)$ and $\hat{H}_i(t-1)$ are components obtained from (11).

By using (12), the distance between two histograms can be defined as:

$$D(H(t),\hat{H}(t-1)) = \sqrt{1 - BC(H(t),\hat{H}(t-1))} \qquad (13)$$

This distance is invariant to the scale of the target, while the popular used histogram intersection is scale variant [29] [30]. The smaller this distance is, the greater the fitness value is, and the better match the particle has with the target object.

E. Algorithm Summary

Considering the problem of object tracking as a searching process for the best match region of the object model, PSO is applied to achieve the efficiency and real-time performance. This algorithm can be summarized as followings:

PSO-based object tracking:

Given object description $p_{b'}(x_{b'}, y_{b'}, l_{b'}, w_{b'}, \theta_{b'})$, object histogram $\hat{H}(t-1)$, and new image frame $I(t)$.

Step1: pre processing the image frame to reduce noise.

Step2: initialize particles around the previous object as $P = \{p_i \mid p_i(x_i, y_i, l_i, w_i), i = 1,2,...,N\}$.

For every particle i:

{

Step3: update its location using (3) and (4).

Step4: calculate fitness value of new locations by (13); update the local and global best $\hat{x}$ and $\hat{s}$ for next iteration.

}

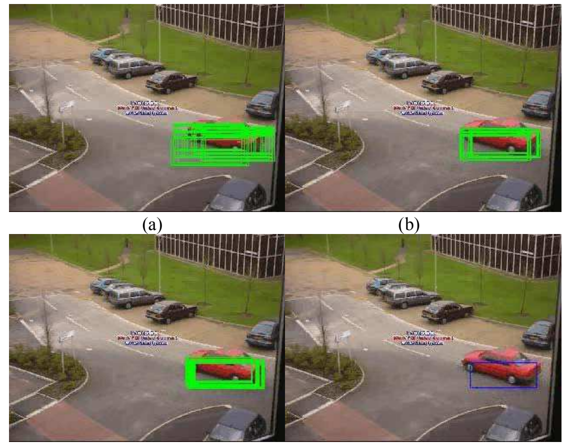
Step5: after visiting all particles, if stop conditions are met, go to step6; otherwise go to step3.

Step6: update object tracking window by calculating the central point of the qualified particles; update object histogram by using (11).

IV. Experimental Results

Based on the proposed PSO tracking algorithm, an automatic system is constructed to detect and track object. Given a video sequence, a detector consisting of cascade classifiers based on Haar-like features [31] [32], is used to recognize the interested object from the backgrounds. Once the object is detected, PSO starts to track it in the following frames until the object moves out of the scene. And in the following experiments, the system is written by C++ and runs on Pentium4 PC with Windows platform.

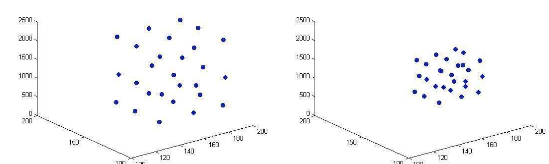
Fig.2 shows the evolvement and convergence process of PSO. The video clip is of 25 fps and 320 x 240 for each frame. Firstly in Fig. 2(a), particles are initialized evenly as a group of windows with varied sizes scattered widely. Then they start moving and converging as shown in Fig. 2(b) and (c). During the process, the diversity of particles is decreasing, which means they are approaching to the global best. When the convergence is reached as Fig. 2(d), the new tracking window is generated as well.



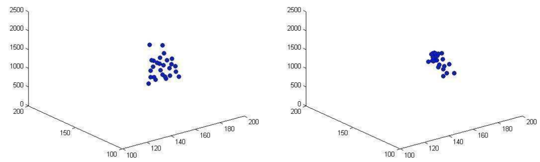
(c) (d)

Fig.2 the evolvement and convergence of PSO

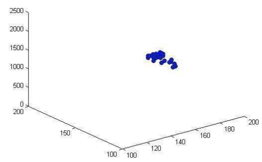
Fig.3 provides the procedure of how the PSO algorithm moves and eventually converges in the solution space. To simplify the display, the four-dimensional solution space is reduced into a 3D space as shown. The horizontal plane stands for the image to indicate center locations of potential windows associated with particles. The vertical axis presents different sizes of potential windows. Particles are drawn as dots in the space and their positions represent the associated windows. Just like what is shown in Fig. 2, at the beginning, particles are evenly distributed as Fig. 3(a); then with attraction of the optimum, they come closely and quickly cluster in Fig. 3(b), (c) and (d); Eventually all particles shrink into a very narrow region based on which the new tracking window is created, as shown in Fig. 3(e). 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, only five iterations are needed to converge to the searching result.



(a) (b)



(c) (d)



(e)

Fig.3 the convergence of PSO in solution space

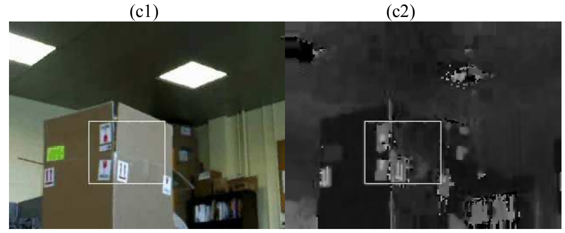
A. Experimental Results of Tracking

Fig.4 shows some snapshots of a tracking video clip using the proposed algorithm in an indoor office environment. Pictures are taken at 24th, 49th, 74th, and 103rd frames. The left column is tracking results with true data and the right one is hue layers. From row a to row c, the student stands up and walks across the room. During this process, the angle of his face changes from the front to the side as well as its size. The background and the light condition are dynamic. However, his face is always being tracked. From the hue layers, it is obvious that the tracking window holds the very similar pattern for various frames and even in c2 with a very small area of interest (AOI). In picture d, when the object is blocked by the box, the pattern disappears. However the particles still tries to find the pattern matched with the object. So that they converge to a region of the box which has the similar hue pattern as shown in Fig. 4(d2). This example demonstrates how the proposed algorithm adapts to catch the changes of the object and try to find the best result as much as possible under a dynamic environment.

(a1) (a2)

(b1) (b2)

(c1) (c2)



(d1) (d2)

Fig.4. An indoor tracking video clip

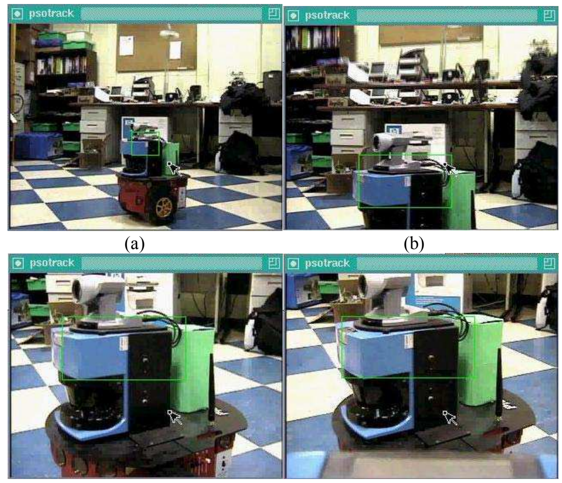
Another series of snapshots is shown in Fig.5 in an outdoor environment. The upper body of a reporter is taken as the object. Images are cut from Internet and their scenes are noisier. However, the proposed algorithm can still catch the object in general when the reporter is moving with a dynamic background. 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. 5(a2), the selected pattern includes a small dark region that belongs to the head of the reporter. In Fig. 5(c2), this head part is out of the tracked object; however other dark regions from the body come to make up the whole pattern of the object. Only the color is used for the fitness function here. If more features and stronger pattern are introduced, the tracking will be improved

(a1) (a2)

(b1) (b2)
(c1) (c2)
(d1) (d2)

Fig.5. An outdoor tracking video clip

Then the proposed detection and tracking algorithm is running and evaluated on a mobile robot with a pan-tilt-zoom camera. The robot runs a Linux-based platform with a Renesas SH7144-based microcontroller. The robot wanders around and real-time pictures are sending back to another computer called the control server via wireless network. The user selects any interested object remotely from the server and the robot will track and follow the object autonomously. In such an environment, due to the delay of the wireless network, there exist jitter movement of the robot and the camera. Therefore, the image sequences become noisy and wiggling, which brings more challenges to the tracking problem. In Fig. 6(a) an arbitrary region is selected as the object, which includes a small blue region, a small black region, and a small region of background with mixed colors. When the mobile robot moves closer to the target, the pictures will be zoomed automatically and the object is locked well, as shown in Fig. 6(b). Then, as shown in Fig. 6(c) and (d), the robot tilts the camera to keep the interested object in the center of the scene which brings many changes to the images. However the proposed algorithm is still capable of tracking the object with this dynamic change. In this example, the PSO tracking method catches up with images changes and provides the real-time tracking performance with 320 x 240 images and 25fps.



(a) (b)
(c) (d)

Fig.6. A real-time tracking task by a mobile robot equipped with a PTZ camera

B. Discussion

In the above experiments, the proposed algorithm 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. From the hue images showed above, it is observed that the object patterns are well tracked and locked. Even when the pattern is lost, the proposed method still tries hard to find the closest match, like Fig. 4(d) showing. The initial wide distribution of particles naturally makes this algorithm to handle jumps between image frames like Fig. 6. Finally, this convergence is simple and quick to let the algorithm achieve real-time performance.

The accuracy of this method still depends on other factors of problems. In our experiments, only color information is used and it is not always strong enough. Like Fig.4, the color pattern is lost, which makes the PSO algorithm missed the target. Introducing more feature clues to construct more powerful fitness function will enhance the performance of the algorithm, and the PSO algorithm is open to benefit from these underlying methods.

V. Conclusion and Future Work

In this paper, a PSO-based searching algorithm is proposed for the object tracking under dynamic environments. The basic mechanism of the PSO algorithm is introduced, and the proof of the convergence of the PSO algorithm is provided. Referring to the object tracking, an accumulative histogram is applied to construct the fitness function, which drives the particles to search around within the high-dimensional solution space. In addition, a cascade classifiers algorithm is applied to make automatic object detection. The experimental results show the converging process and the efficiency and robustness of the proposed method under dynamic environments.

This is a preliminary experiment of using the PSO-based algorithm for the object tracking, and there are still some open issues need to investigate in the future. Firstly, although some simple analyst is made, including swarm size and parameter setting, numerical discussion would definitely help to understand and use PSO in a more advanced level. Secondly, as mentioned, the better fitness function with more advanced object models will improve the algorithm. Furthermore, the PSO-based method has the potential to cooperate with other algorithms. For example, adding the learning module could help for the initialization and the parameter adjustment. The probability estimation can also direct the initialization of particles more efficiently.

REFERENCES

[1] G. Welch and G. Bishop, "An Introduction to Kalman Filter", *SIGGRAPH 2001*, Course 8, 2001.

[2] Erik Cuevas, Daniel Zaldivar, and Raul Rojas, "Kalman filter for vision tracking", Technical Report B 05-12, August, 2005.

[3] P.S. Maybeck, "Stochastic Models Estimation and Control", Vol 2. New York, Academic Press, 1982.

[4] Yu Huang, Thomas S. Huang, and Heinrich Niemann, "Segmentation-based Object Tracking Using Image Warping and Kalman Filtering", *IEEE ICIP*, Vol. 3, pp. 601-604, 2002.

[5] Jacinto Nascimento and Jorge S. Marques, "Robust Shape Tracking In The Presence of Cluttered Background", *IEEE transaction on multimedia*, Vol. 6, pp. 852-861, 2004.

[6] S. Maskell and N. Gordon, "A Tutorial on Particle Filters for On-line Nonlinear/Non-Gaussian Bayes'Law Tracking", 2001.

[7] Changjiang Yang, Ramani Duraiswami and Larry Davis, "Fast Multiple Object Tracking via a Hierarchical Particle Filter", *ICCV*, 2005.

[8] M. Jaward, L. Mihaylova, N. Canagarajah and D. Bull, "Multiple Object Tracking Using Particle Filters". IEEE AC 2006.

[9] Jung 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.

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

[11] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. "Pfinder: Real-Time Tracking of the Human Body", *Technical Report 353*, MIT Media Lab Perceptual Computer Section, 1995.

[12] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *IEEE Conf. on Computer Vision and Pattern Recognition 2001*, Vol. 1, pp. 511-518, 2001.

[13] I. Haritaoglu, D. Harwood, and L. S. Davis, "Hydra: Multiple People Detection and Tracking Using Silhouettes," in *10th International Conference on Image Analysis and Processing (ICIAP'99*), pp. 280-285, 1999.

[14] S. Birchfield, "Elliptical head tracking using intensity gradients and color histograms", *IEEE CVPR*, pp. 232-237, 1998.

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

[16] 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.

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

[18] Wei Qu and Dan Schonfeld, "Robust Kernel-Based Tracking Using Optimal Control", *IEEE, ICIP*, 2006.

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

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

[21] 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.

[22] 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*.

[23] L. Anton-Canalis, M. Hernandez-Tejera, and E. Sanchez-Nielsen etc, "Particle Swarms as Video Sequence Inhabitants For Object Tracking in Computer Vision", *Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications (ISDA'06*), Vol. 2, pp. 604-609, ISDA '06.

[24] L. Anton-Canalís, E. Sanchez-Nielsen and M. Hernández-Tejera, "Swarmtrack: A Particle Swarm Approach to Visual Tracking", *VISAPP'06*, Setúbal, Portugal, Feb. 25-28 2006. Vol.2. pp 221-228.

[25] R. C. Eberhart and J. Kennedy. "A New Optimizer Using Particles Swarm Theory", in *Proc. Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan., IEEE Service Center, Piscataway, NJ, 39-43,1995.

[26] J. Kennedy and R.C. Eberhart, "Particle Swarm Optimization", in *Proceedings of IEEE International Conference on Neural Networks*, 1942-1948, 1995.

[27] Ioan Cristian Trelea "The Particle Swarm Optimization Algorithm: Convergence Analysis and Parameter Selection", Information Processing Letter 85 pp. 317-325, 2003.

[28] T. Kailath, "The Divergence and Bhattacharyya Dis-tance Measures in Signal Selection," *IEEE Trans. Com-mun*. Tech., COM-15:52-60, 1967.

[29] D. Comaniciu, V. Ramesh, and P. Meer, "Real-Time Tracking of Non-Rigid Objects using Mean Shift", *CVPR*, Vol. 2, pp. 142-149, 2000.

[30] M. J. Swain, D.H. Ballard, "Color Indexing," *Intern. J.Comp. Vis*., 7(1):11-32, 1991.

[31] A. Kuranov, R. Lienhart, and V. Pisarevsky, "An Empirical Analysis of Boosting Algorithms for Rapid Objects With an Extended Set of Haar-like Features*," Intel Technical Report MRL-TR-July02-01*, 2002.

[32] G. Bradski, A. Kaehler, and V. Pisarevsky, "Learning-Based Computer Vision with Intel's Open Source Computer Vision Library," *Intel Technical Report,* Volume 09, Issue 02, 2005.