BAYES-BASED OBJECT TRACKING BOOSTED BY PARTICLE SWARM OPTIMIZATION

Yuhua Zheng and Yan Meng
Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030, USA

Keywords: Vision, object detection, tracking, particle swarm optimization, Bayes law.

Abstract: This paper presents a novel Bayes-based object tracking framework boosted by a particle swarm optimization (PSO) algorithm, which is a population based searching algorithm. Basically two searching steps are conducted in this method. First, the object model is projected into a high-dimensional feature space, and a PSO algorithm is applied to search over this high-dimensional space and converge to some global optima, which are well-matched candidates in terms of object features. Second, a Bayes-based filter is used to identify the one with the highest possibility among these candidates under the constraint of object motion estimation. The proposed algorithm considers not only the object features but also the object motion estimation to speed up the searching procedure. Experimental results demonstrate that the proposed method is efficient and robust in object tracking.

1 INTRODUCTION

Object detection and tracking in images is an active research area which has attracted extensive attentions from multi-disciplinary fields, and it has wide applications in many fields like service robots, surveillance systems, public security systems, and virtual reality interfaces. Detection and tracking of moving object like car and walking people are more concerned, especially flexible and robust tracking algorithms under dynamic environments, where lightening condition may change and occlusions may happen.

Up to now, the underlying mathematical models of most tracking methods are Bayes' law estimation and Hidden Markov Model (HMM). The most popular approaches to predict discrete probability distribution are Kalman filter (G. Welch and G. Bishop, 2001), condensation (M. Isard, 1998), particle filter (S. Maskell and N. Gordon, 2001) and mean shift (D. Comaniciu, and P. Meer, 2002). Kalman filter has the same idea with HMM, while Kalman filter deals with discrete variables. Some researchers proposed different control and noise models into the recursion function for image processing, however those assumptions are dependent on varied applications and need to be tuned carefully. Condensation methods mainly focus on how to sample probabilities and likelihoods. When these methods are applied to multiple objects, a dominant peak is established if an object has large likelihood values more frequently, which may depress and lose other objects. The performance of particle filter based methods is limited by dimensionality of state space, which may be feasible in the cases with fewer targets, but may be intractable with a large amount of targets. Generally speaking, the mean-shift algorithm is efficient for object tracking. However the searching window may drift away from the object under dynamic conditions. For example, if the kernel is lost from the tracked target in one frame under some emergent situations, such as illumination condition change, it would be difficult for the tracker to recover itself from this unpredicted event.

Usually for object tracking, an analysis window based on the expectation of objects features is built and scan over the image to find out areas of interest (AOI). However, most conventional analysis-window based trackers are influenced by the shape and size of the window, which may vary from one frame to another. It is difficult to find the appropriate window for each frame, especially under dynamic environments where the content of the images may be dramatically changed.

There are various features can be used for object detection and tracking, such as color, shape, texture,

Zheng Y. and Meng Y. (2007).
BAYES-BASED OBJECT TRACKING BOOSTED BY PARTICLE SWARM OPTIMIZATION.
In *Proceedings of the Fourth International Conference on Informatics in Control, Automation and Robotics*, pages 101-108
DOI: 10.5220/0001646901010108

gesture, contour, and motion. Some successful methods take advantage of knowledge of objects, such as shape or structures. However, the shape-based methods cannot handle the cases with occlusions efficiently. Appearance histogram is applied as tracking cue in this paper due to its independency with objects' shape and structure.

A Bayes-based object tracking approach using a particle swarm optimization (PSO) algorithm is employed to search for an optimal window in a super feature space based on appearance histogram instead of image plane directly. The PSO algorithm (J. Kennedy, R. C. Eberhart, and Y. Shi, 2001) was inspired by the social behavior of a flock of birds. In PSO algorithm, birds in a flock are symbolically represented as particles. These particles can be considered as simple agents "flying" through a problem space. A particle's location in the multi-dimensional problem space represents one solution for the problem. When a particle moves to a new location, a different solution is generated. This solution is evaluated by a fitness function that provides a quantitative value of the solution's utility.

The PSO algorithm is effective for optimization of a wide range of searching problems. In this problem, particles fly around the feature space, trying to find the best-fit tracking window parameters based on the fitness function of object features using appearance histogram. When some particles successfully detect the objects, they will share that information with their neighbors, and their neighbors may follow the directions to reach objects very quickly. 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 decision-making procedure can efficiently prevent the local optimum, which may cause the searching window drift.

By using PSO, the problem of identifying tracking window is translated from one-to-one estimation into one-to-many searching, which brings more flexibility. Since this searching procedure is conducted only in the object feature space, to improve searching results, a Bayes law filter is constructed based on the motion constraints of tracked objects to identify the most possible solution. Generally it is reasonable to assume that objects move consecutively in successive frames. The Bayes law filter tries to keep inertia of the object motion. Compared with conventional window-tracking algorithms, the proposed method can be executed automatically, and moving objects can be detected and tracked in a more flexible and robust way.

This paper is organized as follows. Section 2 simply reviews some related work in object detection and tracking. Section 3 introduces the PSO algorithm. The Bayes-based adaptive-window approach boosted by the PSO algorithm is described in Section 4. Experimental results are discussed and analyzed in Section 5. Conclusion and further work are given in section 6.

2 RELATED WORKS

There are many systems proposed in the past few decades for object detection and tracking. Zhang et al. (Zhang et al., 2006) proposed a robust method to detect moving objects at distance using a mobile camera. Through the utilization of the focus of expansion (FOE) and its associated residual map, the proposed method is able to detect and separate independently moving objects (IMOs) from the "moving" background caused by the camera motion. Leykin and Hammoud (Leykin and Hammoud, 2006) used a combined input from RGB and thermal cameras to build background model and tracker for pedestrians. This method showed robustness for outdoor environments. Olson and Brill (T. Olson and F. Brill, 1997) built a general purpose system for moving object detection and event recognition, where objects were detected and tracked by both first-order prediction and nearest neighbor matching.

The work which is most related to our method is (Yuri Owechko, Swarup Medasani, and Narayan Srinivasa, 2004), where the authors treated every particle as a classifier with different parameters. Those classifiers swarm in the solution space to converge to the optimal analysis window. However this is a simple application of PSO for people detection only. Reza Akbari etc. (Reza Akbari, Mohammad Davarpanah Jazi, and Maziar Palhang, 2006) employed both PSO algorithm and Kalman filter in a hybrid framework of region and object tracking, where vehicles were tracked in a cluttered background. A PSO algorithm was proposed in (Luis Anton-Canalis, Mario Hernandez-Tejera, and Elena Sanchez-Nielsen etc., 2006) to drive particles flying over image pixels directly, where object tracking emerged from interaction between particles and their environment.

3 PARTICLE SWARM OPTIMIZATION

PSO algorithm is an efficient optimization method proposed by Kennedy and Eberhart in 1995 (R. Eberhart and J. Kennedy, 1995) (J. Kennedy and R.C. Eberhart,1995) from the simulation of a simplified social model, which obviously has its root in artificial life in general, and in bird flocking, fish schooling and swarming theory in particular. On the other hand, it is also a method of evolutionary computation, related with both genetic algorithm and evolutionary programming.

The PSO algorithm is population-based: a set of potential solutions evolves to approach a convenient solution for a problem. Being an optimization method, the aim is to find the global optimum of a real-valued fitness function defined in a given search space. Rather than just being a social simulation, PSO can be treated as a powerful new search algorithm, capable of optimizing a wide range of N-dimensional problems.

The social metaphor that leads to this algorithm 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 the environment (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 the 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.

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 *i*th particle velocity in iteration *t*, x_{id}^{t} is the component in dimension d of the *i*th particle position in iteration *t*, c_1, c_2 are constant weight factors, p_{id}^{t} is the best position achieved by particle *i*, 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, and w is the inertia weight. The PSO requires tuning of some parameters: the individual and sociality weights c_1, c_2 , and the inertia factor w .

The mechanism of PSO implicitly assumes that in most real world situations, the optima have better residence around them. Experimentally during the search, regions with high fitness values attract more particles and make particles concentrated after a few iterations. So this type of search is faster and more effective than traditional scanning and gradient methods. On the other hand, PSO is simpler than genetic algorithm since all particles employ the same mechanism during evolutions. Although basic PSO is designed for only single optimum, there are many works have been done to process more complex issues (Kennedy, J. &R.Eberhart, 1997).

4 THE APPROACH

4.1 General Idea

Basically, object tracking can be considered as a probability-based classification and estimation, which searches for the best match of the target model. Usually searching algorithms rely on two factors: searching space and searching window. In terms of the searching space, the more features the object has, the larger the searching space will be. To expedite the search, we can either bound the searching space with some constraints, or develop an efficient searching algorithm. Considering the searching window, adaptive windows have been extensively utilized due to its robustness.

In this paper, we propose a framework which combines a PSO-based searching algorithm and a Bayes-based probability algorithm to achieve the efficiency and robustness of the tracking systems. Basically, a PSO-based searching algorithm identifies the changes in the scene, and the probability-based algorithm estimates the best candidate of the object with the highest possibility. More specifically, the PSO algorithm takes fast scouting in a high-dimensional feature space and finds out some object candidates. Then Bayes law filter decides which one is the best match.

4.2 Object Detection

Usually object detection and recognition depend on the features of the object, such as color, texture, and shape. As indicated in (J. R. Jain and A. K. Jain, 1981), most changes in video content are typically due to the motion of objects in the depicted scene relative to the imaging plane, and a small amount of motion can result in large differences in the values of the samples in a picture, especially near the edges of objects. Often, predicting an area of the current picture from a region of the previous picture that is displayed by a few samples in spatial location can significantly reduce the need for a refining difference approximation. We call this special displacement motion vectors.

Since only the moving objects are considered to be tracked in this paper, the object detection turns into motion detection where a simple background subtraction method is applied. When the detection starts, the first several frames are looked as the background. In the following frames, the moving targets can be easily detected by a motion detection algorithm using background subtraction. During this procedure, the histogram model of background is built and updated by averaging every coming frame to achieve higher robustness. The motion vector $V_i, i = 1,2,..., N$ can be obtained, where V_i represents motion vectors of particle *i*, and *N* represents the total number of particles. Once a valid object is identified, the tracking algorithm kicks in.

4.3 PSO-based Searching Algorithm

From the view of searching, the PSO algorithm is a distributed convergence method. The key is to take advantage of sharing information between the particles as well as their own past experiences to accelerate the convergence. The PSO algorithm would provide an optimal or near-optimal solution using appropriate fitness functions without the complete knowledge of the searching space.

To identify an object in an image, usually rectangle windows are utilized in most cases. 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 Figure 1. These parameters can build up a four-dimensional search space.

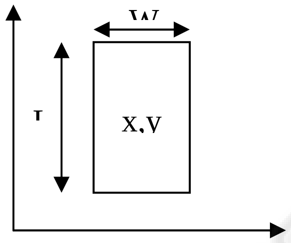


Figure 1: The four parameters associated with a particle window.

So in such a space, each particle presents a search window with specific values of parameters, which can be defined as:

$$P = \{ p_i \mid p_i(x_i, y_i, l_i, w_i), i = 1,2,..., N\} \quad (3)$$

Where x_i and 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*; 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.

Generally a four-dimensional feature space is very large, which makes search algorithms to be computation extensive. Some motion-based constraints can be applied to limit the search area to a smaller region where particles are initialized and move around. A straightforward constraint is using the continuity of movement since it is reasonable to assume that motion is continuous under most tracking situations. In other words, the tracking window of a new frame should be adjacent to its previous one. In this way, the initialization of particles could be accelerated. Suppose $p_{b'}(x_{b'}, y_{b'}, l_{b'}, w_{b'}, \theta_{b'})$ is the best particle (i.e., tracking window) in last frame, the initialized particles $p_i(x_i, y_i, l_i, w_i, \theta_i)$, where *i* = 1,2,…,*N*, in the new frame should be around $p_{b'}$ with some offsets in each dimension. In our experiments, locations are shifted up to 15 pixels, and sizes are shrunk and extended up to 20 percent. Therefore, by dispersing particles in a relatively smaller region instead of the whole space, searching procedure can be definitely accelerated.

Then particles move around, communicate and share information among the society, follow the better directions of their neighbors, and converge to the optima. This process is automatic and independent on knowledge of image contents. After a number of iterations, particles cluster around one or several optimal points in the space, which correspond to some regions with varied locations and sizes. These regions are candidates for the Bayes filter, which will be discussed in later section.

4.4 Fitness Function

The behaviors of particles are guided by the associated fitness function, which defines the search criteria underlying the PSO searching algorithm. In terms of object tracking, fitness function can be defined as a function of features of the tracked object. Lots of features are used for objects detection and tracking, including color, texture, shape and motion, which can be employed independently or several features can be combined together. In this paper, the appearance histogram is applied to construct the fitness function.

First, images are transformed from RGB format into HSV format, and the later one is more natural for people's eyes. Then, the values of hue are abstracted to build the histogram. Such histogram refers to the gradation of color within the visible spectrum. When a PSO-based searching algorithm is applied, each particle at every moment is associated with a histogram. The best matched one can be obtained by comparing these histograms with the target histogram. Therefore, a special criterion is required to measure the similarity between the searched window and the target window, which means a method to measure the distance between two histograms is required.

In statistics (T. Kailath, 1967), the Bhattacharyya Coefficient measures the similarity of two discrete probability distributions. It is a divergence-type measure that can be seen as the scalar product of the two vectors having components as the square root of the probability of the points $x \in X$. It thereby lends itself to a geometric interpretation: the Bhattacharyya Coefficient is the cosine of the angle enclosed between these two vectors. Therefore, the Bhattacharyya Coefficient is used to measure the similarity between these two histograms, which is defined as:

$$BC(H_i, H_g) = \sum_{x \in X} \sqrt{H_i(x) H_g(x)} \tag{4}$$

Where H_i represents the histogram of particle i, H_g represents the histogram of the target, and X denotes the distribution domain, which is the range of hue values from 0 to 255. $H_i(x)$ and $H_g(x)$ are pixel numbers with a specific hue value x for the particle and target, respectively.

By using (4), the distance between two histograms can be defined as (D. Comaniciu, V. Ramesh, and P. Meer, 2004):

$$D(H_i, H_g) = \sqrt{1 - BC(H_i, H_g)}\ . \tag{5}$$

This distance is invariant to the scale of the target, while the popular used histogram intersection is scale variant (M.J. Swain, D.H. Ballard, 1991). The smaller this distance is, the better the particle is matched with the target object. Thus given the target histogram, the fitness function for particle i is inversely proportional to the distance between H_i and H_g :

$$F(p_i, g) = 1 / D(H_i, H_g) \tag{6}$$

The higher the fitness value, the more similar the corresponding area is with the target.

4.5 Bayes-Based Filter

For each frame a motion vector V can be calculated according to a motion trajectory of the tracking window. The motion vector is zero in the first frame. And for others, it is the shift from the previous position to the current one.

Given the previous tracking window associated with the target histogram and the motion vector $\{H_g, V_g\}$, where V_g represents the motion vector of target. The PSO-based searching algorithm returns a set of candidate windows, which can be represented by $\{H_i, V_i \mid i = 1,2,...,m\}$, where H_i represents histograms of particle i, V_i represents motion vectors of particle i, and m is the number of the selected candidate windows. All of these candidate windows are good enough in terms of appearance features and their fitness values are higher than a preset threshold.

According to Bayes law, the problem can be described as:

$$p(H_i, V_i \mid H_g, V_g) = \frac{p(H_g, V_g \mid H_i, V_i)\, p(H_i, V_i)}{p(H_g, V_g)} \tag{7}$$

$p(H_i, V_i \mid H_g, V_g)$ represents the condition probability of a particle with $\{H_i, V_i\}$ given $\{H_g, V_g\}$. $p(H_g, V_g)$ represents the probability of the target window, which is same for all particles. $p(H_g, V_g \mid H_i, V_i)$ represents the back projection from candidates to the previous tracking window. Since all of particles can point back to the target window in different ways, it is hard to tell which particle is the most possible one without any predefined knowledge of the image environment. In this paper, we simply assume that all $p(H_g, V_g \mid H_i, V_i)$, i = 1,2,…,m, are equal. However this assumption may not hold in some practical applications, for instance a mobile vision system where the previous motion trajectory of the mobile platform would provide more information for the back projection, which will be investigated in our future work.

Considering that the PSO-based searching algorithm returns all of candidates which are good enough in appearance histogram, it is reasonable to ignore the histogram here and simplify (7) as:

$$p(V_i \mid V_g) = cp(V_i)\ , \tag{8}$$

where c is a positive constant factor, and $p(V_i)$ represents the probability of a particle on the motion trajectory. According to the inertia of motion, $p(V_i)$ depends on the distance between V_i and V_g . The closer two vectors are, the higher the possibility of the corresponding particle, which makes (8) as the following equation:

$$p(V_i \mid V_g) = cp(V_i) = \frac{k}{D(V_i, V_g)} \tag{9}$$

where k is a positive factor. If two vectors are shifted to the same original point, the distance between two vectors turns into the distance between two points, where Euclidean distance can be calculated.

5 EXPERIMENTAL RESULTS

To evaluate the proposed algorithm, some video clips from PETS database are applied in this paper. The program is written in C++ using OPENCV library, running on a Pentium4 desktop. Most data come from a surveillance system with a stationary camera.

Figure 2 shows the process of identifying moving objects by motion detection, where pictures from left to right are true data, foregrounds, and backgrounds, respectively. If there is no moving object, as shown in Figure 2(a), the background is the same with true image and the foreground is empty since no object is detected. With some general preprocessing, the noise can be depressed and the model of background can be enhanced. When a car drives in, it is detected and recognized as an object. As shown in Figure 2(b), a car shape appears in the foreground while the background keeps the same with the true image. For most testing data with static background, motion detection can detect moving objects quickly. For those testing data under dynamic environment, some pre-knowledge of objects, such as moving behaviors, would help to improve the detection performance.

Figure 3 shows the procedure of the proposed PSO algorithm searching for candidate windows. A number of particles are distributed around the target according to the tracking window of previous frame in Figure 3(a). Due to the uncertainty of the object movement, initially, these windows are set up as different sizes and locations near the detected object using motion detection. Then particles start to move around and eventually converge to some optimal points under PSO rules. Figure 3(b) shows these optimal points, which are good candidates of tracking windows. As shown in Figure 3(b), it is obviously that these candidate windows are much closer to the car compared with those initial windows in Figure 3(a), which demonstrates the efficiency of the PSO-based searching algorithm. Then Bayers filter is applied to select the best match from those good candidates, as shown in Figure 3(c). Usually, the PSO-based searching algorithm converges quickly. In our experiments, initially 20 windows are generated, then after 10 to 15 time steps, those windows cluster to the object.

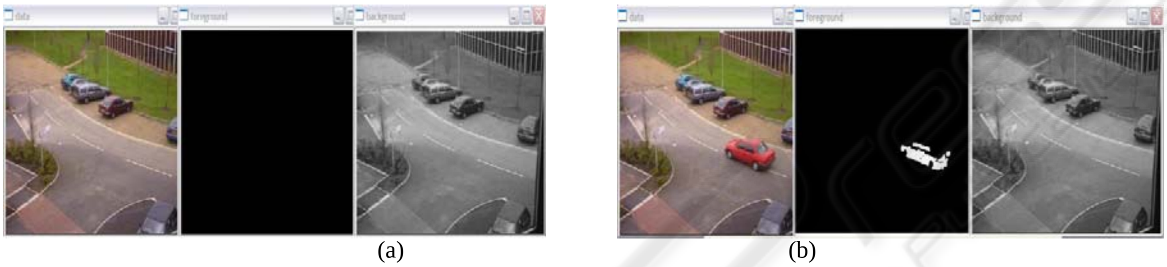
To evaluate the robustness of the proposed tracking method under occlusion, another experiment is carried out as shown in Figure 4. First, a white car drives in and is detected as the target by a blue rectangle window as shown in Figure 4(a). Then, the white car traverses the scene and is occluded by a block of texts in the image, as shown in Figure 4(b) and (c). During the occlusion, the tracking window changes with scenes, but still tracks the car. As shown in Figure 4(b), when the car starts moving into the block, the tracking has almost the same size with the one in Figure 4(a). Under the influence of the block, the tracking window shifts a little and shrinks. But the object is still locked. When the car moves away as shown in Figure 4(d), the window becomes smaller until disappeared. It can be seen that the tracker can still lock the object under occlusion.

The above experiments demonstrate the proposed algorithm is efficient and robust. However under some complex situations, such as dynamic background, more robust motion detection is required. For some noisy videos, the tracking window may be lost due to frame skips. A recovery algorithm may need to increase the system reliability.

6 CONCLUSION

In this paper, a robust adaptive-window based tracking algorithm is proposed to automatically detect and track moving objects. First, a motion detection algorithm is applied to detect the moving object. Then a PSO-based searching algorithm comes to search for good candidates of adaptive tracking windows with parameters on the new fame. Last, Bayes-based filter is employed to identify the best-matched tracking window under the motion constraints of the tracked object. The experimental results demonstrate that the proposed algorithm is robust and efficient in some popular used video data

There are still several issues remained and need to be improved and extended in the future work. The first one is to investigate new object detection approaches under dynamic environment where the background of the image and illumination conditions may be dramatically changed and the motion detection and histogram-based method applied in this paper will not be reliable any more. The second one is to concrete the Bayes filter using some predefined knowledge of the tracked targets.



(a) (b)

Figure 2: Motion detection to recognize objects, (a) to (b) from left to right.

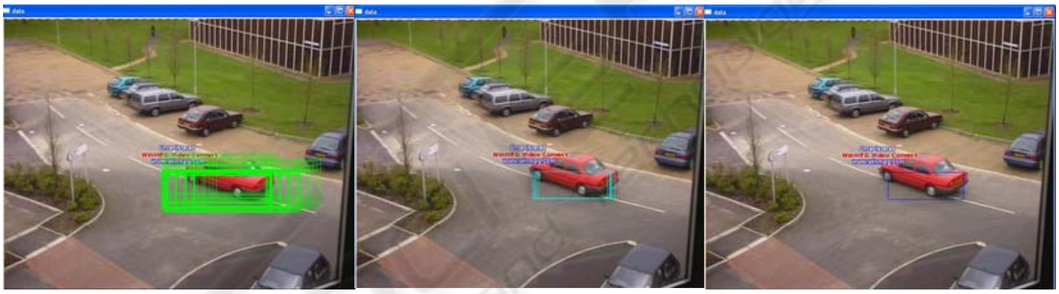
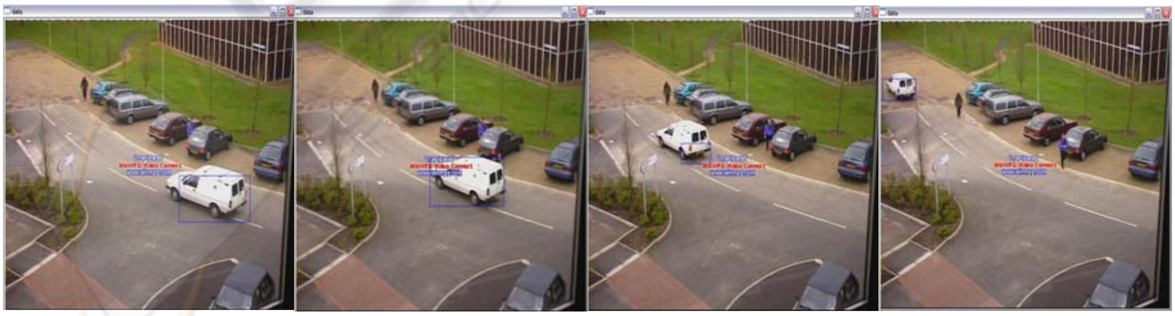


Figure 3: Tracking procedure using PSO-based searching.



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

Figure 4: Tracking under occlusion.

REFERENCES

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

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

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

D. Comaniciu, and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Trans. Pattern Analysis Machine Intel.*, Vol. 24, No. 5, pp. 603-619, 2002.

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

T. Olson and F. Brill. "Moving Object Detection and Event Recognition Algorithms for Smart Cameras*", in Proc. DARPA Image Understanding Workshop,* pp. 159-175, 1997.

Yuri Owechko, Swarup Medasani, and Narayan 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.

Reza Akbari, Mohammad Davarpanah Jazi, and Maziar Palhang "A Hybrid Method for Robust Multiple Objects Tracking in Cluttered Background", Vol. 1, 24-28, *ICTTA '06*.

Luis Anton-Canalis, Mario Hernandez-Tejera, and Elena Sanchez-Nielsen etc. "Particle Swarms as Video Sequence Inhabitants For Object Tracking in Computer Vision", Vol. 2, pp. 604-609, *ISDA '06*.

R. 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.

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

Kennedy, J. and R. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proceedings of the Conference on Systems, Man and Cybernetics*, pp. 4104-4109, Piscataway, New Jersey, IEEE Press, 1997.

J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding", *IEEE Trans. Commun*. Vol. COM-29, no.12, pp.1799-1808, Dec. 1981.

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

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

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

A. Leykin and R. I. Hammoud, "Robust Multi-Pedestrian Tracking in Thermal-Visible Surveillance Videos", *IEEE Computer Vision and Pattern Recognition Workshop, OTCBVS'06*, New York,USA, 17-22 June 2006 pp:136 – 141

Y. Zhang, S. J. Kiselewich, W. A. Bauson, and R. Hammoud, "Robust Moving Object Detection at Distance in the Visible Spectrum and Beyond Using a Moving Camera", *IEEE Computer Vision and Pattern Recognition Workshop (CVPRW), on Object Tracking and Classification Beyond the Visible Spectrum*, New York, USA, 17-22 June 2006, pp: 1331-137