A Mobile Camera Tracking System Using GbLN-PSO With An Adaptive Window

Zalili Musa and Rohani Abu Bakar
Faculty of Computer System and Software Engineering
University Malaysia Pahang
Kuantan, Malaysia
Email: zalili@ump.edu.my and rohani@ump.edu.my

Junzo Watada,
Graduate School of Information, Production & Systems
Waseda University
Fukuoka, Japan
Email: junzow@osb.att.ne.jp

***Abstract*—The availability of high quality and inexpensive video camera, as well as the increasing need for automated video analysis is leading towards a great deal of interest in numerous applications. However the video tracking systems is still having many open problems. Thus, some of research activities in a video tracking system are still being explored. Generally, most of the researchers are used a static camera in order to track an object motion. However, the use of a static camera system for detecting and tracking the motion of an object is only capable for capturing a limited view. Therefore, to overcome the above mentioned problem in a large view space, researcher may use several cameras to capture images. Thus, the cost will increases with the number of cameras. To overcome the cost increment a mobile camera is employed with the ability to track the wide field of view in an environment. Conversely, mobile camera technologies for tracking applications have faced several problems; simultaneous motion (when an object and camera are concurrently movable), distinguishing objects in occlusion, and dynamic changes in the background during data capture. In this study we propose a new method of Global best Local Neighborhood Oriented Particle Swarm Optimization (GbLN-PSO) to address these problems. The advantages of tracking using GbLN-PSO are demonstrated in experiments for intelligent human and vehicle tracking systems in comparison to a conventional method. The comparative study of the method is provided to evaluate its capabilities at the end of this paper.**

***Keywords*-Dynamic tracking; particle swarm optimization; an adaptive window; pattern matching; mobile camera.**

I. INTRODUCTION

The most important research in tracking systems is to find and develop new technologies for tracking applications. Currently, the latest trend in the systems is to use a mobile camera for searching a large view space. It is because of the use a static camera system to detect and track the motion of an object, as in [1][2][3], is only capable of capturing a limited view [4]. As a result, many cameras are used to capture an image in a large view space [4]. However, the cost increases with the number of cameras. To control the costs, a mobile camera is employed with the ability to track the wide field of view in an environment.

Nevertheless, tracking an object with a mobile camera should consider several issues, such as dynamic background and motion as well as an occlusion. Therefore, this study proposes a Global best Local Neighborhood Oriented Particle Swarm Optimization (GbLN-PSO) algorithm for detecting and tracking a static or moving object in a large view space. This paper involves the use of pattern matching for moving object detection. This algorithm will help the researchers to address the following problems:

1) Dynamic background-As the camera is moving while tracking an object, each scene might capture a different background. Therefore, it is difficult to distinguish the moving object from its background.
2) Dynamic motion-When tracking with a mobile camera, the location of an object may be dynamic although the object is static, or vice verse, the camera and object may move concurrently during the process of capturing the image. Therefore, it is difficult to track the object's location.
3) Occlusion among objects-This study considers two causes of occlusion; object motion and camera views. Principally, occlusion among objects occurs when two or more objects are moving in parallel or skew directions. In addition, the focus of the camera can also make objects appear to overlap one another. Therefore, a movable camera tracking system is expected to mitigate this problem.

This paper is presented in five different sections. Section 1, the introduction and problem statement on the study are discussed, and Section 2 will highlight some literature related to the study. Then, Section 3 will describe the techniques and algorithms of the tracking process in detail. Section 4 will present the outcome and simulation results, and the entire study will be concluded with discussion and opportunities for future work in Section 5.

II. RELATED WORKS

Currently, PSO is a very popular method that has been applied in various fields, such as camera tracking [5], tracking in signal processing [6], clustering [7] and optimization [8]. However, this method is seldom used in camera tracking systems. Only a small number of researchers have applied PSO for tracking human movement in indoor and outdoor environments with a cluttered background [5]. Some of these researchers are Zhang et al. and Kobayashi et al., who

978-0-7695-4562-2/11 $26.00 © 2011 IEEE
DOI 10.1109/CIMSim.2011.53

have proposed the use of PSO in detecting the motion of an object, such as a bouncing ball [9]. In addition, Wang et al. has proposed a new method, enhancing a particle swarm optimization based particle filter tracker to search for a human head [10]. Sulistijono and Kubota have proposed human head tracking for a partner robot using particle swarm optimization to reduce computational cost and time [11].

The Particle Swarm Optimization (PSO) algorithm was introduced by Kennedy and Eberhart in 1995 [12]. The idea of PSO was to simulate the flocking behavior of birds or the schooling behavior of fish, in which, from observations on behavior of birds and fish, they came up with the assumption that a group of birds randomly search for their food in a certain area. The PSO algorithm is similar to evolutionary computation techniques, such as Genetic Algorithms (GA). For instance, PSO randomly uses particle groups to seek the best solution to a problem, such as combinatorial optimization.

As preparation for the study, the PSO algorithm was divided into three steps: 1) the initialization of the particles, 2) the evaluation of the population and 3) the updating of the particles. In the first step of PSO, a distribution of particles is randomly generated in the problem space. Each particle is assigned a certain fitness value by evaluating the fitness function.

Next, each particle is evaluated in every iteration compared to the local best and global best values obtained among the particles in the population. The local best refers to the best solution that is the current best position among particles surrounding the local particle. For example, if the fitness value of some particle close to the current particle is better than the one of the current particle, then the fitness value of the local best is set to the fitness value of that particle. Concurrently, a global best is immediately updated when a new best position is found considering all of the particles in the problem space.

Lastly, the velocity and position of each particle are updated based on the following formulas:

$$V_{ix}^{t+1} = \omega V_{ix}^{t} + c_1 r_1 (P_{lx} - x_i^t) + c_2 r_2 (P_{gx} - x_i^t) \quad (1)$$

$$V_{iy}^{t+1} = \omega V_{iy}^{t} + c_1 r_1 (P_{ly} - y_i^t) + c_2 r_2 (P_{gy} - y_i^t) \quad (2)$$

$$x_i^{t+1} = V_{ix}^{t+1} + x_i^t \quad (3)$$

$$y_i^{t+1} = V_{iy}^{t+1} + y_i^t \quad (4)$$

where the acceleration values c_1 and c_2 are constant, r_1 and r_1 are uniformly distributed random numbers between 0 and 1, the current loop time is t, the total loop time is T and the inertia weight ω is decreases linearly from the maximum value ω_{max} to the minimum value ω_{min} during the loop time T, P_l is the local best and is P_g the global best value.

The final result is obtained within the optimization process, and it depends on the preset terminal condition, such as finding an optimal solution or reaching a maximum number of iterations. When the terminal condition is not satisfied, the algorithm is repeated.

However, the conventional PSO is unable to identify the optimize value of search space; which is one of the vital drawback. By nature, PSO is randomly distribute conventional particle locations. Somehow, the global optimum is distant from the target location. Thus, the probability of swarm to explore into the other search space is almost impossible. As a result, the swarm is trapped in local optima. To overcome this problem, the hybrid PSO mixes the use of the traditional velocity and position update rules of star, ring and Von Neuman topologies all together. Nonetheless, these method is weak and a new strategy is needed, especially if the particle trapped in local minimum in their first iteration [17].

Hence, the objective of this research to propose and generalize new variant of PSO algorithm, called Global-best Local Neighborhood oriented Particle Swarm Optimization (GbLN-PSO) algorithm to find a better solution without trapping in local minimum models and to overcome the decision conflict by the conventional PSO. In addition, the proposed method is expected to reduce the searching processing time. Finally, performance analysis will be undertaken to validate the strength of the proposed method over conventional PSO algorithm. It is expected that the findings of this research will give high impact in this field and later will be further employed as a mechanism to solve real world discrete optimization problems such as shortest path problem and other problems.

III. Target Tracking

In target tracking module, the GbLN-PSO algorithm has been applied in order to find the true location of object in the next frame. This module can be divided by ten (10) steps. The details process of each step are discussed as below:

1) Read Video Image: Once the object detection module is finished, reading the next video image will be started. As mentioned before, a target or visual object can move rapidly in any direction from each frame to the next. Therefore to overcome the problem, we need to predict the new target window, d.

2) Predict The New Window: In this research, the purpose of target window is to simplify the process of object location detection. Basically, the velocity and direction of object motion are changes and not fixed for every frame. Thus, using a target window in the object tracking process, the distribution location of a particle can be determined according to the location of target window. Beside that, if the distribution location of a particle is wide and distant from the object. It is difficult to identify the precise location of object and the target detection will be loosed. This situation will be effected to determine the location of an object.

Assuming, the center coordinate of object after object detection module are $C_{x,i}$ and $C_{y,i}$, respectively. Thus, the formula for prediction window can be calculated as:

$$W_{x,i} = C_{x,i} - a \tag{5}$$

$$W_{x,i+1} = C_{x,i} + a \tag{6}$$

$$W_{y,i} = C_{y,i} - b \tag{7}$$

$$W_{y,i+1} = C_{y,i} + b \tag{8}$$

where a and b are constant values.

3) Initialize a Position of a Particle: In the first stage of the GbLN-PSO process, $\eta = 0$, the positions of L particles and a fitness value should be initialized by the following formula:

$$P_l^{\eta}(px, py, f) = 0, \tag{9}$$

where $p(.,.,.)$ denotes a particle and $l = [1, ..., L]$ a particle number, f is a fitness value and px and py are coordinate value based on axes X and Y. However, for the next iteration $\eta > 0$, the random location of particle can be assigned as follows:

$$px = rand(1, L) \times m, \tag{10}$$

$$py = rand(1, L) \times n, \tag{11}$$

where m and n are the size of a target window according to the equations (5), (6), (7) and (8). Figure 1 shows the process of initialize position of particles for the target windows.

To ensure the new position of a particle does not exceed the limit, the following specific rules are implemented:

$$P_l^{\eta+1}(px) = rand(1) \times M : 1 > P_l^{\eta+1}(px) > m. \tag{12}$$

$$P_l^{\eta+1}(py) = rand(1) \times N : 1 > P_l^{\eta+1}(py) > n. \tag{13}$$

4) Assign The Fitness Value: For each particle, we need to find out the fitness values as:

$$f = \begin{cases} 0 : & \eta = 0 \\ \tau_{l,\eta} : & \eta > 0 \end{cases} \tag{14}$$

where η is the iteration number, and τ the value obtained from the pattern matching process.

Assume that the template image of an object has a $q \times r$ sized window. Therefore, the formula of the pattern matching process is

$$\tau^n = \frac{cr}{(\sum_{i=1}^{q} \sum_{j=1}^{r} sI_{i,j}^s)} \tag{15}$$

where i and j are integer numbers, $1 \leq i \leq q$ and $1 \leq j \leq r$. Meanwhile, s denotes the number of small windows, and sI is a pixel value of the image frame. Based on the formula above, cr is the feature correlation between the template image ϑI and the small window image of the video frame ϑT, denoted in the following

$$cr = \sum_{i=1}^{q} \sum_{j=1}^{r} (\vartheta I_{i,j}^s \times \vartheta T_{i,j}) \tag{16}$$

where ϑI and ϑT are defined as follows:

$$\vartheta T = sI_{i,j}^s - \frac{\sum_i^q \sum_j^r sI_{i,j}^s}{q \times r} \tag{17}$$

$$\vartheta I = sO_{i,j}^s - \frac{\sum_i^q \sum_j^r sO_{i,j}^s}{q \times r} \tag{18}$$

where sO is a pixel value of the template image. According to the equations (15), (16), (17) and (18) above, the general algorithm of the pattern matching process can be described as follows:

i. Get the feature values of the small image, ϑI using (18).
ii. Get the feature values of the template, ϑT using (17).
iii. Get the feature correlation between the template and the small image, cr using (16).
iv. Find the template matching values, τ using (15).
v. return τ.

In this study, pattern matching is a vital process in object tracking. The purpose of this process is to find a small bit of an image that matches the query region of that object pattern. Basically, this technique is derived from a mean formula to identify matching features between the two elements. Additionally, To identify similar features of the object in every frame, the current object image (template) is compared to the preceding image. Hence, in the pattern matching process, the latest features of the object are compared to the former features of the objects as seen in the previously captured images. Later, all of the comparison values are stored as a fitness value. To achieve the maximum value of fitness function, the local best position for every particle must be identified.

5) Find The Local Best Position: At the first iteration, $\eta = 1$, all of the components of particles will be assigned to local best positions, $pbest$, as follows:

$$pbest_l^{\eta} = P_l^{\eta}. \tag{19}$$

Then, for $\eta > 1$, $pbest$ is defined based on the following rule:

$$pbest_l^{\eta} = \begin{cases} P_l^{\eta} : & pbest_l^{\eta}(f) \leq P_l^{\eta}(f) \\ pbest_l^{\eta} : & pbest_l^{\eta}(f) > P_l^{\eta}(f) \end{cases} \tag{20}$$

6) Find The Global Best Position: The global best position, $gbest$ is the maximal value of the local best. Therefore, $gbest$ is defined as follows:

$$gbest^{\eta} = pbest_l^{\eta} : pbest_l^{\eta}(f) \equiv \max(pbest^{\eta}(f)) \tag{21}$$

7) Neighborhood Comparison: After the process of determining global best, the fitness value of $gbest$ will be compared to the template matching values of its local neighborhood. As a first step of this process, the location of $gbest$'s local neighborhood needs to identify in the following formula:

$$gn_g(px) = gbest^{\eta}(px) + Vx_g \tag{22}$$

Figure 1. Process of initialize position of particles for the target windows.

$$gn_g(py) = gbest^\eta(py) + Vy_g \tag{23}$$

where Vx and Vy are random values between -2 to 2. On the other hands, $gn_g(px)$ and $gn_g(py)$ are locations of $gbest$'s local neighborhood based on axes X and Y. Where g is a integer number.

According to (22) and (23), the template matching value for each $gbest$'s local neighborhood can be employed as (16). Thus, the comparison between fitness value of $gbest$ and template matching values of its local neighborhood, $gn_g(cr)$ can be shown as follows:

$$gbest^\eta = \begin{cases} gn_g(cr): & gbest^\eta(f) \le gn_g(cr) \\ gbest^\eta: & gbest^\eta(f) > gn_g(cr) \end{cases} \tag{24}$$

8) Update The Velocity and Position of Particle: The current position $P^\eta(px, py)$ and velocity of vx and vy are updated after each iteration via the following equations:

$$vx^{\eta+1} = \omega \times vx^\eta + c_1 r_1 (pbest_l^\eta(px) - P_l^\eta(px)) + c_2 r_2 \\ (gbest^\eta - P_l^\eta(px)). \tag{25}$$

$$vy^{\eta+1} = \omega \times vy^\eta + c_1 r_1 (pbest_l^\eta(py) - P_l^\eta(py)) + c_2 r_2 \\ (gbest^\eta - P_l^\eta(py)). \tag{26}$$

$$P_l^{\eta+1}(px) = P_l^\eta(px) + vx^{\eta+1}. \tag{27}$$

$$P_l^{\eta+1}(py) = P_l^\eta(py) + vy^{\eta+1}. \tag{28}$$

where ω is an inertia weight, c_1 and c_2 are constant values, and r_1 and r_2 are random values.

9) Target Location Detection: Once $\eta \equiv \max(\eta)$, the process of target location detection will be executed in the following equation:

$$C_{x,i} = P_l^\eta(px) \tag{29}$$

$$C_{y,i} = P_l^\eta(py) \tag{30}$$

10) Updated Template: As we mentioned in Section 1, the object and background of image are changes for every frame. Therefore, to overcome the problem of dynamic object and background, for every frame we need to change the template into the new template in the equation below:

$$sO_i = \begin{cases} sO_{i+1}: & ((distX < c_1) \& (distY < c_2)) \\ sO_i: & ((distX > c_1) \& (distY > c_2)) \end{cases} \tag{31}$$

where

$$distX = |C_{x,i} - C_{x,i-1}| \tag{32}$$

$$disY = |C_{y,i} - C_{y,i-1}| \tag{33}$$

In this case c_1 and c_2 are equivalent to 10.

IV. Result and Discussions

As mentioned in Section 1 in this chapter, the most important goals of the proposed method are to defeat three issues in employing mobile cameras for object tracking. To identify the extent of efficiency of the proposed method, the researchers compared this method with other two conventional methods: (1) Particle Swarm Optimization with Euclidean distance and (2) continuous detection and Kalman filter.

The researchers considered two main issues with using an active camera in object tracking: dynamic background and motion. The results shown in Table I above indicate that the proposed method successfully tracked the vehicle plate with 94.67% accuracy, compared to Particle Swarm Optimization with Euclidean distance with 87.11% and continuous detection and Kalman filter with only 80.89%. Likewise, in tracking human motion, the table shows that the proposed method has achieved the best percentage with 92.89%, followed by Particle Swarm Optimization with Euclidean distance with 83.78% and continuous detection and Kalman filter with only 79.78%. These findings imply that, compared to the other two methods, the proposed method is the most efficient in tracking motion objects.

In addition, the results in Table 2 above also imply that the Particle Swarm Optimization with Euclidean distance

Table I
COMPARISON OF RESULTS OBTAINED FROM THREE METHODS FOR TRACKING

Methods	Vehicle Plate(%)	Human (%)
GbLN-PSO	94.67	92.89
Particle Swarm Optimization with Euclidean distance	87.11	83.78
continuous detection and Kalman filter	80.89	79.78

and continuous detection and Kalman filter are unstable in tracking an object's movement as compared to the proposed method, especially in a crowded area. This is due to the use of mobile cameras in both methods. The mobile camera captures different background images in every frame and produces dynamic images of the motion of objects. Therefore, the views of an object apparently change from one frame to another in a video stream. This reason makes it difficult for both methods to recognize the pattern of the targeted object. The continuous detection and Kalman filter is most appropriate to be employed for objects in linear motion. Thus, this method hardly tracks objects in dynamic motion. Figure 2 shows several results produced by the proposed method in dynamic background and motion.

Additionally, compared to the Particle Swarm Optimization with Euclidean distance and continuous detection and Kalman filter method, the proposed method is more effective in handling the occlusion of objects. Figure 3 shows a comparison between the results of these three methods. Figures 3(a) and (b) demonstrate that the Particle Swarm Optimization with Euclidean distance and continuous detection and Kalman filter methods failed to detect the accurate location of the targeted object. However, the proposed method has successfully detected the precise location of objects even in occlusion conditions, as shown as Figure 3(c).

V. CONCLUSIONS

This study has proposed a novel method that has successfully addressed the three issues mentioned in Section 1 in the chapter; thus, this study has achieved its goal. The study examined several video data sets of humans and vehicle plates to evaluate the effectiveness of the proposed method. The results indicate that the proposed method has successfully tracked the locations of humans and vehicle plates. In particular, the study found that the proposed method has effectively addressed the occlusion of objects. In contrast, Particle Swarm Optimization with Euclidean distance and continuous detection and Kalman filter failed to track the locations of objects properly, especially in crowded places, and failed to address the occlusion of objects. This study has also established that using GbLN-PSO in template matching can reduce processing time by up to 55% and 98.8% compared to conventional PSO and template matching processes, respectively.

However, the proposed method still requires some improvements. This method is unable to recognize the location of a totally hidden object. Hence, this method needs to be combined with a prediction method as in [17]. As a final point, the researchers have proposed a new method to overcome the tracking application problem using a mobile camera. This chapter has emphasized the use of GbLN-PSO with an adaptive window to make the system a success.

REFERENCES

[1] M. Ebbecke, M.B.H. Ali, and A. Dengel, Real time object detection, tracking and classification in monocular image sequences of road traffic scenes, International Conference on Image Processing, Vol. 2, pp. 402-405, 1997

[2] I. Fernandez, M. Mazo, J.L. Lazaro, D. Pizarro, E. Santiso, P. Martin and C. Losada, Guidance of a mobile robot using an array of static cameras located in the environment, Vol. 23, pp. 305-324 , 2007.

[3] B. Lai, D. Y. Zhang, Z. Y. Yuan and J. H. Zhao, Crowd Segmentation from a Static Camera, Lecture Notes in Computer Science, Advanced Intelligent Computing Theories and Applications. With Aspects of Theoretical and Methodological Issues, Vol. 5226/2008, pp. 1134-1140, 2008.

[4] Z. B. Musa, and J. Watada, A Grid-Computing Based Multi-Camera Tracking System for Vehicle Plate Recognition, Kybernetika Journal, Vol. 42, No. 4, pp. 495-514, 2006.

[5] X. Zhang, W. H. Maybank, S. X. Li, M. Zhu, Sequential particle swarm optimization for visual tracking, IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-8, 2008.

[6] H. Zhu, S. Ngah, Y. Xu, Y. Tanabe, T. Baba, A Random Time-varying Particle Swarm Optimization for Local Positioning Systems, Vol. 8 No. 6 pp. 49-60, 2008.

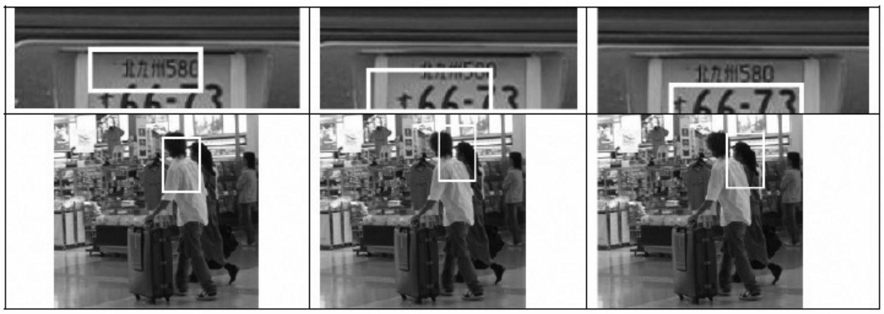
[7] S. Li, X-J. Wang, Y. Zhang, X-SPA: Spatial Characteristic PSO Clustering Algorithm with Efficient Estimation of the Number of Cluster, Fifth International Conference on Fuzzy Systems and Knowledge Discovery, Vol. 2, pp. 533-537, 2008.

[8] S. Shangxiong, A Particle Swarm Optimization (PSO) Algorithm Based on Multi-agent System, International Conference on Intelligent Computation Technology and Automation (ICICTA), Vol. 2, pp. 802-805, 2008

[9] T. Kobayashi, K. Nakagawa, J. Imae, G. Zhai, Real Time Object Tracking on Video Image Sequence Using Particle Swarm Optimization, International Conference on Control, Automation and Systems, pp. 1773-1778, 2007.

[10] Q. Wang, L. Xie, J. Liu and Z. Xiang, Enhancing Particle Swarm Optimization Based Particle Filter Tracker, Lecture Notes in Computer Science, Special Session on Intelligence Computation and Its Application, Vol. 4114/2006, pp. 1216-1221, 2006.

Figure 2. Results of proposed method for dynamic background and motion.



(a) (b) (c)

Figure 3. Comparison of overlapping objects tracked between (a) Particle Swarm Optimization with Euclidean distance (b) continuous detection and Kalman filter (c) GbLN-PSO with an adaptive window.

I. A. Sulistijono, and N. Kubota, Human Head Tracking Based on Particle Swarm Optimization and Genetic Algorithm, Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol.11, No.6 pp. 681-687, 2007

D. Murray and A. Basu, Motion Tracking with an Active Camera, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 16, No. 5, pp. 449-459, 1994.

M. Isard and A. Blake, Contour tracking by stochastic propagation of conditional density, Proc. of the 4th European Conference on Computer Vision, Vol. 1, pp. 343-356, 1996.

F. Pei, P. Cui and Y. Chen, Adaptive MCMC Particle Filter for Nonlinear and Non-Gaussian State Estimation, International Conference on Innovative Computing Information and Control,

L. Song, R. Zhang, Z. Liu and X. Chen, Object Tracking Based on Parzen Particle Filter Using Multiple Cues, Lecture Notes in Computer Science, Advances in Multimedia Information Processing-PCM 2007, Vol. 4810/2007, pp. 206-215,

Z. B. Musa and J. Watada, A Particle Filter Approach for Multi-Camera Tracking Systems in a Large View Space, International journal of innovative computing, information & control (IJICIC), Vol. 6, No. 6, pp. 2827-2836, 2009.

[17] S. A. Hamdan, Hybrid Particle Swarm Optimiser using multi-neighborhood topologies, INFOCOMP - Journal of Computer Science, Vol.7, No.1, pp.36-43, 2008.