A new Weight Adjusted Particle Swarm Optimization for Real-time Multiple Object Tracking

Guang Liu[1], Zhenghao Chen[1], Henry Wing Fung Yeung[1], Yuk Ying Chung[1]

[1]School of Information Technologies, University of Sydney, Sydney, NSW 2006, Australia
guang.liu@sydney.edu.au

Abstract
This paper proposes a novel Weight Adjusted Particle Swarm Optimization (WAPSO) to address the occlusion problem and computational cost in multiple object tracking. To this end, an update strategy of inertia weight of the particles is designed to maintain particle diversity and prevents pre-mature convergence. Meanwhile, the implementation of a mechanism that enlarges the search space upon the detection of occlusion enhances WAPSO's robustness to non-linear target motion. In addition, the choice of Root Sum Squared Errors as the fitness function further increases the speed of the proposed approach. The experimental results shown that in combination with the model feature that enables initialization of multiple independent swarms, the high-speed WAPSO algorithm can be applied to multiple non-linear object tracking for real-time applications.

Keywords: Object Tracking, Particle Swarm Optimization, Root Sum Squared Errors, Multiple Object Tracking

1 Introduction

Recently, object tracking has become a hot topic of research in the field of computer vision. It has diverse application areas including surveillance system, computer-human interaction and traffic monitoring.

Object tracking is defined as the estimation of the trajectory of a target in a sequence of video frames. To track an object successfully, it is necessary to build a model to represent the features of the target object and a mechanism to detect and track the object across multiple frames. Particle Swarm Optimization (PSO) has become a popular choice in object tracking in the recent decade due to its accuracy and fast convergence nature [4, 5, 6, 7, 8]. PSO, proposed by Eberhart and Kennedy, is an evolutionary computation method mimicking the behavior of natural swarms such as bird flocking and fish schooling [1, 2, 3]. During the optimization process, a swarm of particles is at first randomly generated in a search space located inside the frame. The particles are then updated based on their own experience as well as the experience of the whole swarm. Iteration of the process allows the particles to continuously evolve and thus quickly converge to the optimum. However, owing to its fast converging property, the traditional PSO has a few drawbacks, most notably pre-mature convergence and loss in particle diversity. PSO is also found to be ineffective in situations involving occlusion and unexpected disappearance of the target objects. Furthermore, the performance of PSO is severely constrained by the efficiency of its fitness function, yet most of the existing modifications made to the traditional model overlooked the need for a better alternative to the widely adopted histogram based approach.

In this paper, we propose a novel Weight Adjusted Particle Swarm Optimization algorithm (WAPSO) that facilitates convergence to global optimal while maintaining particle diversity. In addition, Root Sum Squared Errors (RSSE) fitness function is chosen as an alternative for the common histogram based approach. Section 2 presents a brief description on the traditional PSO algorithm. Section 3 introduces the WAPSO and the RSSE fitness function in detail. Section 4 presents the experimental results while Section 5 concludes the paper.

2 The Traditional PSO Algorithm

In PSO, a set of particles, defined by their x and y coordinates, are generated randomly across a search space with pre-defined width and height located within the image. Each particle is the center of a search window which is of the same size as the target object. All particles with their corresponding search windows are considered as potential candidates of the solution to the tracking problem. Their fitness are evaluated by a fitness function and subsequent updates will apply to renew their positions in the search space. After sufficient iterations, the particles will converge to the optimum and particle with the best fitness value will be chosen as the solution.

The traditional PSO comprises of two equations and one fitness function. The first equation (1) governs the change in velocity of each particle whereas the second equation (2) updates the position of each particle for a particular iteration.

$$v_n^{t+1} = \omega \cdot v_n^t + C_1 \cdot R_1 (Pbest_n^t - x_n^t) + C_2 \cdot R_2 \cdot (Gbest^t - x_n^t) \tag{1}$$

$$x_n^{t+1} = x_n^t + v_n^{t+1} \tag{2}$$

In equation (1) and (2), x_n^t and v_n^t denote respectively the position and velocity of the n^{th} particles in the t^{th} iteration. The position that gives the best fitness value among all past and present positions of the n^{th} particle is denoted as $Pbest_n^t$ while the best fitting position among all past and present positions of all the particles is given by $Gbest^t$. R_1 and R_2 are random variables with range 0 to 1. C_1 and C_2, which usually sum to 4, are defined respectively as the individual and social factor. ω is the inertia weights that describes the degree of path dependency of the particle.

3 The Proposed Approach

3.1 Weight Adjusted Particle Swarm Optimization (WAPSO)

WAPSO proposed in this paper aims to solve the problem of diversity loss and premature convergence which are the most notable shortcomings of the traditional PSO.

During tracking, it is necessary to maintain a diversified cohort of particles in order to fully utilize the whole search space. Diversity loss refers to a situation which particles process very similar properties. This situation leaves us with a limited choice of potential solutions, thus hinders the performance of the algorithm in locating the global optimum. In the case of diversity loss, we are expected to observe similar distance to global best in most of the particles, or in mathematical terms, a low deviation of distance (D_n). The deviation of the distance to global best is captured by

$$\sigma_D = \frac{Min\ (D_n, \overline{D})}{Max(D_n, \overline{D})} \tag{3}$$

which is a strictly decreasing function of the difference between D_n and $\overline{D}$ for all n particles. Capturing the deviation in distance alone may not be sufficient since equidistant particles to the global best can possibly process very different fitness value. Therefore, it is necessary to have another mechanism to measure the deviation of the fitness values (F_n) from mean $(\bar{F})$ for all n particles, given by

$$\sigma_F = \frac{Min\ (F_n, \bar{F})}{Max(F_n, \bar{F})} \tag{4}$$

which is a strictly decreasing function of the difference between F_n and $\bar{F}$.

In the traditional PSO, pre-mature convergence refers to the situation which the particles converge to a solution without thorough search in the given space. For example, if the particles located at a local optimum instead of the global optimum in the first iteration, all particles will converge pre-maturely to the local optimum. In the ideal case, particles should remain spread out to search for the global optimum during early iterations and coverage when the loop approaches the end. Therefore, we proposed a mechanism that manages the tendency to converge (τ), defined as

$$\tau = 1 - \frac{I_c}{I_t} \tag{5}$$

where I_c and I_t are the current and total number of iteration respectively.

The WAPSO algorithm introduces an update equation for the inertia weight ω_n which incorporates equations (3) to (5). It is given by

$$\omega_n = \tau \cdot (\omega_i + \alpha \cdot \sigma_D + \beta \cdot \sigma_F) \tag{6}$$

where ω_i is initial inertia weight. α and β are parameters with range 0 to 1. The proposed Equation (6) shows that ω_n is a decreasing function of iterations and particle diversity. When particle diversity decreases, the impact of inertia weight for particle n is increased, encouraging the particle to explore more area. Furthermore, the inertia weight ω_n is kept high at early iterations to avoid pre-mature convergence. As iteration number increases, the inertia weight will decrease, allowing for convergence of the particles.

In addition to Equation (6), a mechanism is implemented in the WAPSO aiming to detect occlusion and disappearance of the target object and relocate it once it reappears. This is accomplished by setting a fitness value threshold (F_T). If the fitness value of the global best particle falls below F_T, the search space will enlarge to cover the whole image, allowing for global search to recapture the target object.

Given the above-mentioned advantages of WAPSO, it can achieve more accurate results than the traditional PSO and is more robust to handle occlusion and disappearance problems. Fig. 2 below provides an overall description of the procedure for the proposed WAPSO.

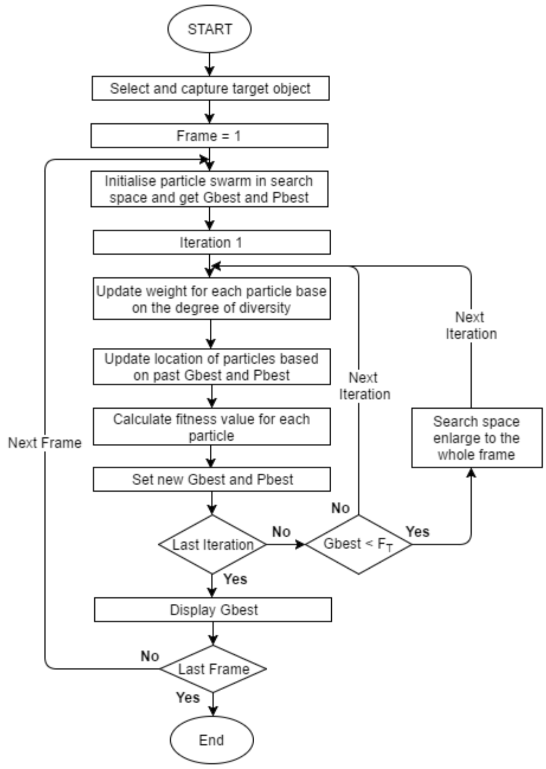


Fig. 2. Flowchart of the WAPSO Algorithm

3.2 Fitness function

A fitness function is required to evaluate the similarity of each candidate particles with the target object. Most previous research adopts a histogram comparison approach as a measure of similarity between the particle and the target object [4,5,6,7]. The chosen alternative in this paper given is by equations:

$$RSSE(p,t) = \sqrt{\sum_i \sum_j \left(p_{ij} - t_{ij}\right)^2} \tag{7}$$

$$Fitness = f\left(\frac{RSSE(p,t)}{M \cdot N}\right) \tag{8}$$

where $RSSE(p,t)$ is the Root Sum Squared Errors, p_{ij} denotes the pixel in the ith row and jth column of the image captured by the search window of a particle and t_{ij} denotes the pixel in the ith row and jth column of the target image. Each pixel is presented either in RGB or in HSV form. The RSSE value is divided by M and N, which represent the row and column of the input image respectively, and is then normalized to the range of 0 to 1. RSSE is chosen for its simplicity which substantially reduces computation and allows for real-time application.

3.3 Multiple Object Tracking

Tracking multiple objects is a challenging task because it usually requires higher computational cost comparing to tracking a single object. In addition, the observations of different target objects may overlap during occlusion which may result in low tracking performance. To overcome these difficulties, we introduce multiple swarms for multiple object tracking. By selecting multiple target objects to be tracked from the screen, multiple object models are established. Each of the models is associated with a particle swarm. The swarms then parallel process the target searching. The parallel feature of this method renders the searching speed while the occlusion problem can be well-handled as each swarm only focuses on its own target object independently.

4 Experiment results

The proposed framework was tested and compared with the traditional PSO and the CPSO proposed by Sha (2015) [6]. The CPSO is a PSO based tracking algorithm that shares similar trait to our proposed WAPSO. It serves as a benchmark to assess our proposed framework. The first testing video assessed the performance of WAPSO using a real world traffic environment. Tracking was performed on a car with neither occlusion nor disappearance from screen. The second video tested the ability of WAPSO in multiple tracking of non-linear targets using 3 balls with different colors. For fair evaluation of our proposed algorithm, we tested our algorithm with a set of fixed parameters described in Table 1. In addition, the number of particles per swarm and the number of iterations are both set to 10 for all algorithms.

Table 1. Parameters for Testing

Algorithm	Parameter	Value
PSO	ω in equation (1)	1
	C_1 and C_2 in equation (1)	2
CPSO	C_1 and C_2	2
WAPSO	ω_i in equation (6)	1
	α and β in equation (6)	0.5
	C_1 and C_2 in equation (1)	2
	Fitness Threshold (F_T)	0.8

The frames of interest are represented in Table 2 and Table 3. The error of tracking is calculated in each frame by the Euclidean distance between the gBest particle and the manually annotated target. The testing results are shown in Fig. 3, 4, 5 and 6. For better visualization, the error is set to 0 in case of occlusion and disappearance. The accumulated runtime for each algorithm has been calculated and documented in Fig. 7 and 8.

Table 2. Video – Car (Pixels: 1920*1080)

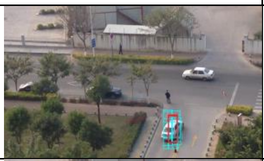
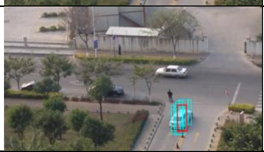
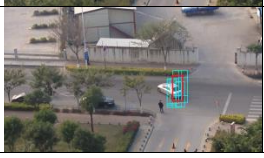
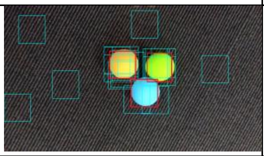
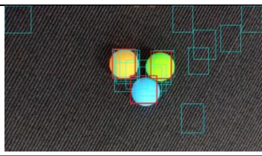
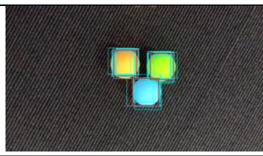
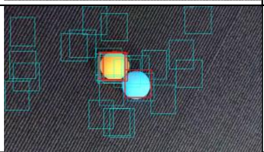
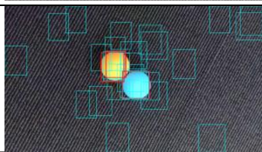
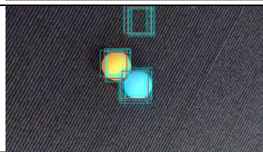
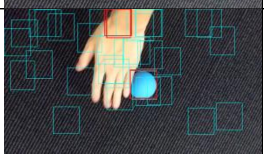
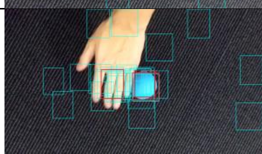
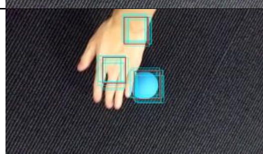
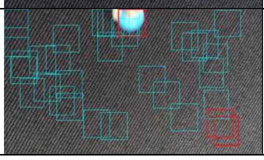
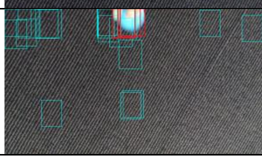
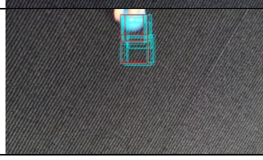
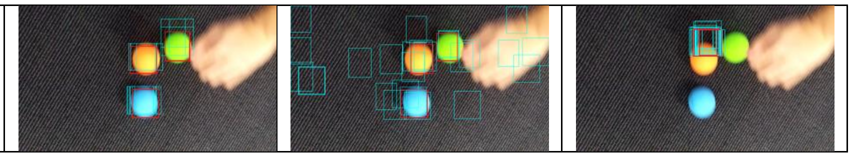
Frame	Test 1: WAPSO	Test2 : CPSO	Test 3: PSO
2			
19			
140			

Table 3. Video - Color Balls (Pixels: 1920*1080)

Frame	Test 1: WAPSO	Test2 : CPSO	Test 3: PSO
2			
35			
60			
116			



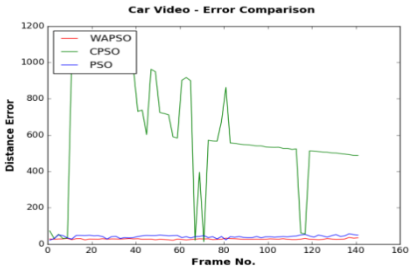


Fig. 3. Accuracy Comparison - Car

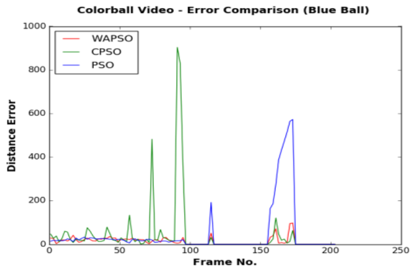


Fig. 4. Accuracy Comparison – Blue Ball

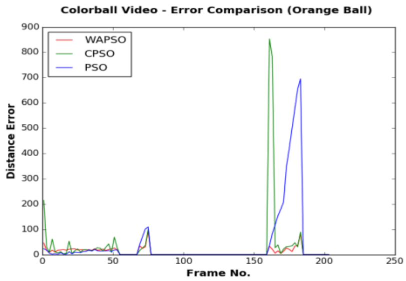


Fig. 5. Accuracy Comparison – Orange Ball

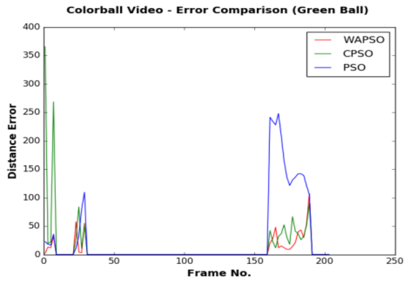


Fig. 6. Accuracy Comparison – Green Ball

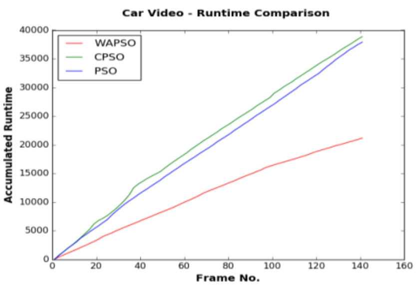


Fig. 7. Runtime Comparison - Car

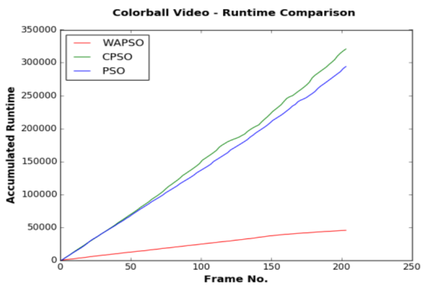


Fig. 8. Runtime Comparison – Color Balls

In the car video, both WAPSO and PSO can successfully track the target car without any loss throughout the whole video while CPSO lost the target from frame 19 onwards and only managed to recapture the target car in a few frames. This corresponds to the relatively mild fluctuation of errors for WAPSO and PSO and the sharp variation of errors for CPSO as shown in Fig. 3. Moreover, the error of the WAPSO fell below that of the PSO in almost all frames, suggesting that the accuracy has improved due to the increase of particle diversity and avoidance of pre-mature convergence. Fig. 7 shows further advantage of WAPSO which only required approximately half the runtime of its counterparts. This can mostly be attributed to a significant reduction in computation cost due to the adoption of the new fitness function.

In the color balls video, the 3 balls with different colors are removed from the screen one by one. This process is demonstrated in Table 3. Fig. 4 to 6 presents the tracking error of the three targets. The results show that WAPSO provided consistent tracking results on all three targets in all frames. CPSO scored satisfactory results with a few occasional peaks of errors whereas PSO failed to recapture the balls after their disappearance. The runtime taken for WAPSO to track all three targets was approximately a sixth of the other frameworks, allowing WAPSO to perform real-time multiple object tracking.

5 Conclusion

In this paper, we present a novel Weight Adjusted Particle Swarm Optimization (WAPSO) for multiple object tracking. The Root Sum Squared Errors is chosen to be an alternative to the widely used histogram based fitness function. Experimental results show that WAPSO has successfully overcome the drawbacks of the traditional PSO in rediscovering objects after occlusion and avoiding premature convergence. WAPSO outperforms both the traditional PSO and the CPSO in terms of speed and accuracy. Due to its substantial speed advantage, WAPSO is demonstrated to succeed in real-time tracking of multiple objects.

References

1. Eberhart R C, Kennedy J. A new optimizer using particle swarm theory[C]//Proceedings of the sixth international symposium on micro machine and human science. 1995, 1: 39-43
2. J. Kennedy and R.C. Eberhart (1997). A discrete binary version of the particle swarm algorithm, Systems, Man, and Cybernetics, 1997 Computational Cybernetics and Simulation, IEEE International Conference, 5(12-15), 4104-4108.
3. R.C. Eberhart and Y. Shi (2001). Particle Swarm Optimization: Developments, Application and Resources, Proceedings of the 2001 Congress on Evolutionary Computation, Seoul, South Korea, Vol. 1, 81-86.
4. Zheng Y, Meng Y. The PSO-based adaptive window for people tracking[C]//Computational Intelligence in Security and Defense Applications, 2007. CISDA 2007. IEEE Symposium on. IEEE, 2007: 23-29
5. Hsu C, Dai G T. Multiple object tracking using particle swarm optimization[J]. World Academy of Science, Engineering and Technology, 2012, 68: 41-44
6. Sha F, Bae C, Liu G, et al. A categorized particle swarm optimization for object tracking[C]//Evolutionary Computation (CEC), 2015 IEEE Congress on. IEEE, 2015: 2737-2744
7. Sha F, Bae C, Liu G, et al. A probability-dynamic Particle Swarm Optimization for object tracking[C]//Neural Networks (IJCNN), 2015 International Joint Conference on. IEEE, 2015: 1-7
8. Zhang L, Tang Y, Hua C, et al. A new particle swarm optimization algorithm with adaptive inertia weight based on Bayesian techniques[J]. Applied Soft Computing, 2015, 28: 138-149