Optimized Particle Filter for Tracking of a Moving Video Object

Basil Hamed[1], Ahmed Alostaz[2]

Department of Electrical Engineering, Islamic University of Gaza, Palestine

[1]bhamed@iugaza.edu

[2]aostaz@gmail.com

Abstract

Recently, the particle filter (PF) is used to track video object by estimating its position in image frame, but it suffers some problems: degeneracy phenomenon and sample impoverishment. In this paper, a new optimized bootstrap particle filter (OBPF) is introduced to solve the problem of particle filter on video object tracking for rigid motion. Where adaptive Chemostic step, bacteria use in adaptive bacteria foraging based on particle swarm optimization algorithm (ABF-PSO), is tackled to improve predict step in PF. Moreover, particle swarm optimizations (PSO) is used in resample step of OBPF. The comparison among OBPF, particle filter based on bacteria foraging optimization (PF-BFO) and particle filter based particle swarm optimization (PF-PSO) on video object tracking is presented in this paper by using Matlab program. The results show that OPF method presents outstanding performance versus PF-PSO and PF-BFO. It has the best velocity and the best accuracy, so it is suitable for real time tracking.

Keywords: Visual Object Tracking, Bootstrap Particle Filter, Particle Swarm Optimization, Adaptive Bacteria Foraging Based PSO.

1. Introduction

The applications of object tracking is heavily used such as computer vision, surveillance, and intelligent transportation. However, there are a number of challenges. Although many substantial researches have been done to tackles the challenges, developing a robust and efficient tracking algorithm still unsolved because of hardship in tracking problem.

Recently, Particle filters have been extensively used in tracking field. They proved to be a robust method of tracking due to their ability of solving non-Gaussian and non-linear problems [1]. Video object tracking through using PF suffers the same problems that PF do: degeneracy phenomenon and sample impoverishment.

The degeneracy phenomenon, where after a few iterations, all but one particle will have negligible weight, is an undesirable effect in particle filter. To reducing its effect, a great number of particles is used. However, this will increase the computational cost. Moreover, in order to reduce the effects of degeneracy, many particle filters introduce a re-sampling procedure whenever a significant degeneracy is observed. The basic idea of re-sampling is to eliminate particles that have small weights and to concentrate on particles with large weights. But the re-sampling step introduces other practical problems, such as the problem of sample impoverishment.

Sample impoverishment occurs when the likelihood is very narrow or the likelihood distribution function lies at the tail of prior distribution. This sample impoverishment can be solved through enlarging the sample set to cover the whole state space and to ensure estimation successfully. Therefore, the computation will be negatively affected

To solve both tracking problems, optimization algorithms are combined with particle filter. In this paper, the researchers use new technique, bootstrap particle filter based on ABF-PSO, to reduce impoverishment and degeneracy problem in video object tracking.

This paper is organized as follows: Section 2 presents literature review. Section 3 introduces the bootstrap particle filter algorithm. Section 4 shows generic PSO. Section 5 describes (ABF-PSO). Section 6 proposes the new optimized particle filter. Section7 shows how to use OBPF Algorithm based in video object tracking. In section8, the experiment results is shown. Finally, Section 9 includes the conclusions and future research.

2. Literature Review

Recent years have witnessed great advances in the literature of optimization tools for Visual tracking. Many approaches of tracking objects are proposed to reduce degeneracy and impoverishment the problem of PF for example:

Two powerful object tracking frameworks namely: PF and PSO for estimating parameters of non-linear and non-Gaussian models is presented in this paper[2], also an Enhanced Particle Filter (EPF) algorithm for object tracking is introduced to enhance the robustness of the tracking algorithm.

Improved particle filter based on genetic algorithm (GA) is proposed [3]. Genetic Monte Carlo into sampling process with the basic idea of solving particle degeneration by means of evolution thought is introduced. It is shown that the novel particle filtering framework can effectively eliminate particle degeneration and reduce its dependency on the particle validity.

The Mean-shift algorithm is added to the particle filter in object tracking [4]. They present a new object tracking algorithm based on particle filtering technique and the mean shift algorithm. In the paper two problems of the particle filter technique are encountered, the degeneracy phenomenon and the huge computational cost. To solve these problems, a new tracking algorithm uses the mean shift algorithm inside the particle filter

Combining particle filter with ant colony optimization is used in video object tracking [5]. A particle filtering algorithm based on ant colony optimization (ACO) was proposed to enhance the performance of particle filter with small sample set. ACO algorithm optimized the sample set before re-sampling step.

PF-BFO, in video object tracking is presented in [6]. The results show that the new particle filter is more accurate and stable tracking. It solves both degeneracy phenomenon and impoverishment problem. The new method has the advantages of both adding mean shift to PF and improving resample method of PF. However, applying new method in real time video tracking needs processor that has high velocity.

3. Bootstrap Particle Filter (BPF)

PF is a Bayes estimation algorithm based on Monte Carlo method[7]. It performs the posterior probability density function via a number of weighted particles and eliminates particles that have small weights and concentrates on particles with large weights using resample. There are a number of pragmatic PF techniques that have been introduced in the literature. In this work, Bootstrap Particle Filter with optimization algorithms and with Gaussian weight function is used.

The basic of bootstrap algorithm developed by Gordon, Salmon and Smith is one of the first practical implementations of the processor to the tracking problem. It is the most heavily applied of all PF techniques due to its simplicity. The bootstrap filter is a variation of PF where the dynamic model $p(xk|\ xk-1)$ is used as the importance distribution. This makes the implementation of the algorithm very easy. In order to construct the bootstrap PF, follow steps[8]:

1. Initialization: Generate the initial state ,$x_i(0)$

   Generate the process noise, $w_i(t)$

2. Prediction: give positions for particles according to the model and previous position.

$$x_i(k) = A(x_i(k-1), u(k-1), W_i(k-1)) \qquad (1)$$

3. Weights of particle (update state): generate the likelihood, $c(y(t)|xi(k))$ using the current particle and measurement

4. Normalize weight: sum of weight =1.

5. Resample particles: resample the set of particles retaining and replicating those of highest weight.

6. Estimation: estimate the best value using eq. (2).

$$\hat{x}_k = E[\,x_k|y_k\,] = \sum_{i=1}^{N} W_k^j * \overset{i}{x_{sk}} * \delta(\,x_{sk}^i - x_s^i) \qquad i = 1, \dots\dots\dots N_s \qquad (2)$$

7. Generate the new set,

$$\{x_i(k), W_i(t)|with\ W_i(t) = 1/N_p\}$$

Because of PF problems , optimization algorithms as PSO, genetic algorithm and BFO are combined with particle filter [9,10]. In this paper, a new optimized particle filter is proposed.

4. Particle Swarm Optimization (PSO)

The particle swarm optimization (PSO) method is an stochastic optimization technique based on natural behavior of birds flocking or fish schooling [11]. In detail, a PSO algorithm is initialized with a group of random as shown in the following steps [6][12]:

1. Assume the size of the swarm (number of particles) is *N*.

2. Generate the initial population of X randomly as X_1, $X_2 \dots X_N$ (positions of particle).

3. Find the velocities of particles. All particles will be moving to the optimal point with a velocity. Initially, all particle velocities are assumed to be zero.
4. In the ith iteration, find the following two important parameters used by a typical particle j :

   (a) The historical best value of $X_j(i)$ (coordinates of jth particle in the current iteration i), $,j$, with the highest value of the objective function, $f[X_j(i)]$ encountered by particle j in all the previous iterations.

   The historical best value of $X_j(i)$ (coordinates of all particles up to that iteration), G_{best}, with the highest value of the objective function $f\big[X_j(i)\big]$, encountered in all the previous iterations by any of the N particles.
   (b) Find the velocity of particle j in the ith iteration as follows:

$$V_j(i) = V_j(i-1) + c_1 r_1 * [(P_{best,i} - X_j(i-1)] + c_2 r_2 * [(G_{best} - X_j(i-1)] \quad (3)$$

(c) Find the position or coordinate of the jth particle in ith iteration as

$$X_j(i) = X_j(i-1) + V_j(i) \quad (4)$$

5. Check the convergence of the current solution. If the positions of all particles converge to the same set of values, the method is assumed to have converged. If the convergence criterion is not satisfied, step 4 is repeated by updating the iteration number as $i = i + 1$, and by computing the new values of $P_{best,i}$ and G_{best}, the iterative process is continued until all particles converge to the same optimum solution.

5. Adaptive Bacteria Foraging Based PSO

ABF_PSO algorithm combines both BFO and PSO [13]. The aim is to make PSO able to exchange social information and BFO able to find new solution by tumble, swim, reproduction, elimination and dispersal in the same algorithm[14]. The detail of algorithm is shown below:

1) Initialize input parameter:

S_p : Total number of bacteria in the population.
N_c : The number of chemotaxtis steps.
N_{re} : The number of reproduction steps.
N_s : The length of swim
N_{ed} : The number of elimination-dispersal events.
P_{ed} : Elimination-dispersal probability.
$C(i)$: The size of the step taken in the random direction.

2) Create random initial swarm bacteria $\theta^i\,(j,k,l)$ and initialize their fitness J_{health}, J_{best}, J_{gbest}

3) For i =1, 2 … N_{ed}
For j =1, 2 … N_{re}
For k =1, 2 … N_c
For l =1, 2 …S_p

Compute cost function $J(i,j,k,l)$

End for

Update J_{best}

For l =1, 2 …S_p

Perform adaptive chemotaxis step:

a) Calculate magnitude of step ($C(i)$) using eq.(5)

$$C(i) = w * norm(\Delta(i)) + c_1 * norm(\theta_{best} - \theta_i) + c_2 * norm(\theta_{gbest} - \theta_i) \quad (5)$$

Where Δ indicates a vector in the random direction whose elements lie in [-1, 1]

b) Perform tumble (indicate direction of step)

c) Swim using eq. (6), until improve fitness or arrive the length of swim.

$$\theta^i\,(j+1,k,l) = \theta^i\,(j,k,l) + c(i) * \frac{\Delta(i)}{\sqrt{\Delta(i)^T * \Delta(i)}} \quad (6)$$

d) Fitness is calculated using only object function.

End for
End for

Perform reproduction step:

- Calculate J_{health} using eq.(7)

$$J^i_{health} = \sum_{j=1}^{N_c+1} J(i,j,k,l) \quad (7)$$

- The S bacteria with the highest J_{health} values die and the remaining S bacteria with the best values split.
- Update J_{gbest}

End for

Perform eliminating _dispersal step for all bacteria $\theta^i\,(j,k,l)$ with probability $0 < Ped < 1$.

End for

By combining between two previous optimization theories with BPF, OPBF is introduced.

6. Optimized Bootstrap Particle Filter (OBPF):

Bootstrap particle filter is optimized by improving prediction step using equation from ABF_PSO and using PSO in resampling step.

Bootstrap filter predicts the new position of particle in one stage using this model $p(xk|\, xk-1)$ as important distribution. In the OBPF, there are two stages for prediction. First stage uses the same relation $p(xk|\, xk-1)$, where the new positions of particles depend on previous positions. The second stage uses a new model $p(xk|yk)$ to predict particle positions.

In new model, new position of particles depend on their current weight. Adaptive Chemostic step proposed in section 5 is used in this model, where particles move in random directions, but in known magnitude. New relation $xk \sim p(xk|yk)$ is implanted by using following step:

1. Weigh particles.
2. Update θ_{best} and θ_{gbest}.

3. Update particle position using eq. (5) and eq. (6)

Besides, PSO is used in resampling step. In order to construct the OBPF, follow next steps:

1. Initialization: Generate the initial state ,$x_i(0)$
   Generate the process noise, $w_i(t)$
2. Prediction I: give positions for particles according to the model and previous position as in eq. (8).

$$X_k = A * X_{k-1} + B * U_{t-1} \tag{8}$$

3. Prediction II : give positions for particles according to the weights of particle

$$xk \sim p(xk|yk)$$

4. Weights of particle (update state): generate the likelihood, using the current particle and measurement
5. Resample particles: resample the set of particles using PSO algorithm.
6. Estimation: estimate the best value using eq.(9)

$$\widehat{X} = \frac{\sum_{i=1}^{N_s} X_k^i}{N_s} \tag{9}$$

7 Video Object Tracking Based OBPF

Achieving video object tracking, OBPF is used as the following:

1. Applying OBPF algorithm to each frame of the video in sequence.

2. Using pixels as particles

3. Using the value and location of pixel in weight function as following:

reducing the difficulty of tracking as the result of changing in elimination, different types of cue are used in weighting particle such as using histogram color, histogram edge [15], texture [16] , histogram of spatial color [17].

In this paper, simple weighing cues are used to show the strength of combining particle filter with optimization algorithms. Moreover, two cues for weighing are used:

1) Histogram color of the particle -pixel- value H_i is compared with wanted histogram color pixel H_{target}:

$$D_H = \text{norm}\ (H_k^i - H_{target}) \tag{10}$$

$$W_H^i = \frac{1}{\sqrt{2\pi\sigma_{color}}} e^{\frac{-D_H^2}{2\sigma_{color}^2}} \tag{11}$$

2) The position of the particle X_i is compared with the best previous position X_{target} which equals previous location of the tracked video object.

$$D_p = \text{norm}\ (X_k^i - X_{target}) \tag{12}$$

Where: D_p^2 is distance in pixel.

$$W_P^i = e^{\frac{-D_P^2}{2\sigma_{postion}^2}} \tag{13}$$

The weight of particle is calculated according to equation:

$$W = W_H^i * W_P^i \tag{14}$$

Two cues are used in saved movie, but first cue is just used in real time video. In addition, to increase the robust of tracking, optimization algorithms are combined with PF.

8. Experiment Results:

The researchers use laptop whose specifications are Intel core i7 processor and 8GB DDR3 to make following tests.

We have done experiment test using OBPF method and compared our results with the results PF-PSO and PF-BFO methods [6].

Figure 1 presents the same video with different picture that used in [6]. The color of the child skin of the object is tracked in sample video. It is taken by digital camera that has low resolution (240*320) and it has high noise, high variance elimination and high similarity between colors as shown in Figure (1). These weak specifications are used to show strength of tracking algorithm.

Figure1: Sample Video for OBPF Test

In this paper two tests are taken; the first test is to measure the minimum particle that each pervious methods needs to follow tracking. The lowest number of particles mean we have the best method results. In real time tracking time is very critical, so fast tracking method also give the best result. In the second test, color standard deviation is used as a measurement tool which introduces less variance more accuracy

Firstly, OBPF and PF_PSO are used with one iteration for frame and PF_PFO is used with following parameter: Nr=1, NC=1and Ns=1 for frame. The results are shown in Table 1, Figure (2), Figure (3) and Figure (4).

Table (1) shows the PF_PSO is fast, but it is the weakest in tracking. Although PF_BFO has better tracking than PF_PSO, it is still has the slowest time. OBPF has the best result in tracking and time.

Table 1: Result of OBPF Test with 7 Standard Deviation

STANDARD DEVIATION	FILER TYBE	Min number of particles hold tracking	TIME (S)
7	PF_PSO	50	0.0045
	PF_BFO	35	0.0263
	OBPF	20	0.0018

Figure (2) (a) and (b) show that the PF_PSO can track the object in the frames 30 and 60 with 50 particles and color standard variation 7. Although PF-PSO with 45 particles can track the object in frame 30 as shown in Figure (2) (c), it can't track in frame 60 as shown in Figure (2) (d).

Figure (3) (a) and (b) show that the PF_BFO can track the object in the frames 30 and 60 with 35 particles and color standard variation 7. Although PF_BFO with 30 particles can track the object in frame 30 as shown in Figure (3) (c), it can't track in frame 60 as shown in Figure (3) (d).

Figure (4) (a) and (b) show that the OBPF can track the object in frame 30 and 60 with 20 particles.

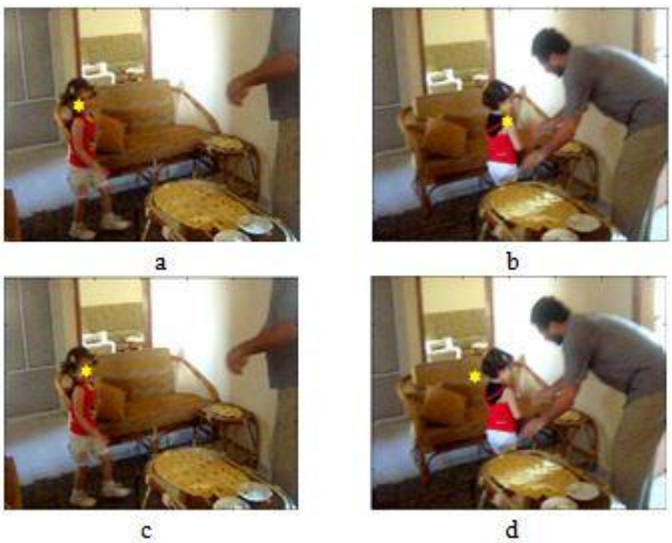

Figure 2: The Tesult Of First Test With PF_ PSO

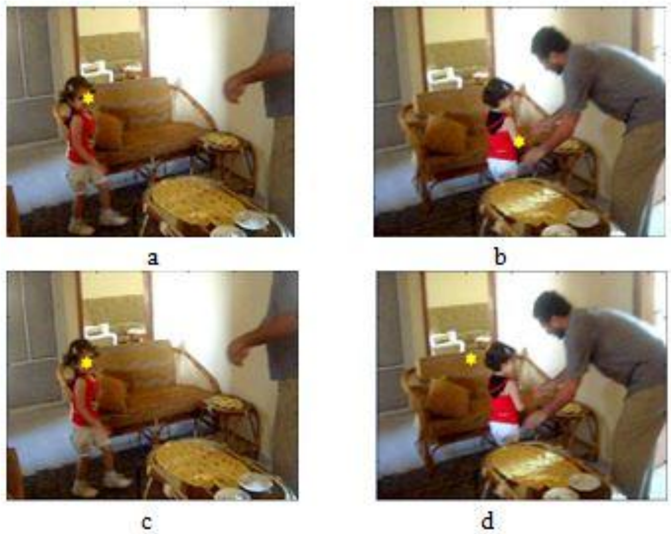

Figure 3: The Tesult Of First Test With PF_ BFO

Figure 4: The Tesult Of First Test With OBPF

Secondly, the effect of color standard deviation changing on the number of particle is measured to show the robust of techniques used in tracking. When Color standard deviation is reduced from 7 to 5, as the result shown in Table (2).

Comparison between Table (1) and Table (2) shows the following; When Color standard deviation is reduced, the number of particle in PF_PSO is increased by 13 particles to hold tracking. The number of particle in PF_BFO is increased by 10 particles and the number of particle in OBPS is increased by only 5 particles. That proofs the strength of OBPF in video object tracking.

Table 2: Result of OBPF Test with 5 Standard Deviation

STANDARD DEVIATION	FILER TYBE	Min number of particles hold tracking	TIME (S)
5	PF_PSO	63	0.0048
	PF_BFO	45	0.0341
	OBPF	25	0.0020

9. CONCLUSIONS

OPBF, unprecedented methods is provided in video object tracking. Comparsion among OBPF, PF_BFO and PF_PSO is made for tracking object in saved movie. OBPF outperfored other methods in term of accuracy, stability and speed. It solves both degeneracy phenomenon and impoverishment problem, where PF has by improving predict step in PF. OBPF has two predect steps. First predict step is similalr to predict step in BPF. Second predict step depends on adaptive Chemostic step.

REFERENCES

1. A. J. Haug, “A Tutorial on Bayesian Estimation and Tracking Techniques Applicable toNonlinear and Non-Gaussian Processes”, MITRE Technical Report MTR 05W0000004, 2005.
2. Howida A. Abd El-Halym, Imbaby I. Mahmoud, A. AbdelTawab and S. E.-D. Habib " Particle Filter versus Particle Swarm Optimization for Object Tracking" 13th International Conference on AEROSPACE SCIENCES & AVIATION TECHNOLOGY 2009, Cairo, Egypt.
3. Long Ye , Jingling Wang , Chuanzhen Li , Hui Wang and Qin Zhang, “A Novel Particle Filtering Framework Using Genetic Monte Carlo Sampling” , International Conference on Management and Service Science, Sept 20-22 2009.
4. Bai and Weiming Liu, “Improved Object Tracking with Particle Filter and Mean Shift” , Proceedings of the IEEE, International Conference on Automation and Logistics , August 18 - 21, 2007, Jinan, China
5. Zhou Hao Xuejie Zhang Pengfei Yu Haiyan Li, “Video Object Tracing Based on Particle Filter with Ant Colony Optimization”, International Conference on Advanced Computer Control (ICACC), March 27-29 2010
6. Mohammed Alhanjouri, Ahmed Al Ostaz" BFO vs. BSO for Video Object Tracking Using Particle Filter (PF)" Journal of Emerging Trends in Computing and Information Sciences 4(2), 8 (2013).
7. Chen, Z.: Bayesian filtering: From Kalman filters to particle filters, and beyond. Statistics **182**(1), 1-69 (2003).
8. Candy, J.V.: Bayesian signal processing: Classical, modern and particle filtering methods, vol. 54. John Wiley & Sons, (2011)
9. Li, M., Pang, B., He, Y., Nian, F.: Particle Filter Improved by Genetic Algorithm and Particle Swarm Optimization Algorithm. Journal of Software (1796217X) **8**(3) (2013).
10. Zhang, G., Cheng, Y., Yang, F., Pan, Q.: Particle filter based on PSO. In: Intelligent Computation Technology and Automation (ICICTA), 2008 International Conference on 2008, pp. 121-124. IEEE
11. N. Nedjah, Luiza de Macedo Mourelle, "Swarm Intelligent Systems", Springer, 2006.
12. Rao, S.S., Rao, S.: Engineering optimization: theory and practice 4th ed. John Wiley & Sons, (2009)
13. Ahmed Alostaz, Alhanjouri, M.: A new adaptive BFO based ON PSO for learning neural network. i-manager ’s Journal o Computer Science **1**(3), 8 (2013).
14. Passino, K.M.: Biomimicry for optimization, control, and automation. Springer, (2005)
15. Liu, K.-Y., Li, S.-Q., Tang, L., Wang, L., Liu, W.: Fast face tracking using parallel particle filter algorithm. In: Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on 2009, pp. 1302-1305. IEEE
16. Chen, R., Zhang, Z., Lu, H., Cui, H., Yan, Y.: Particle-filter-based object tracking with color and texture information fusion. In: Sixth International Symposium on Multispectral Image Processing and Pattern Recognition 2009, pp. 74952F-74952F-74958. International Society for Optics and Photonics
17. Ma, J., Han, C., Chen, Y.: Efficient visual tracking using particle filter. In: Information Fusion, 2007 10th International Conference on 2007, pp. 1-6. IEEE