International Journal of Engineering

Journal Homepage: www.ije.ir

An Efficient Target Tracking Algorithm Based on Particle Filter and Genetic Algorithm

S. Sadegh Moghadasi*[a], N. Faraji[b]

[a] *Artificial Intelligence Group, Faculty of Computer Engineering and Information Technology, Islamic Azad University, Qazvin, Iran*
[b] *Electrical Engineering Department, Imam Khomeini International University, Qazvin, Iran*

PAPER INFO

Paper history:
Received 08 January 2019
Received in revised form 14 March 2019
Accepted 02 May 2019

Keywords:
Object Tracking
Particle Filter
Genetic Algorithm
Sample Impoverishment
Resampling

ABSTRACT

In this paper, we propose an efficient hybrid Particle Filter (PF) algorithm for video tracking by employing a genetic algorithm to solve the sample impoverishment problem. In the presented method, the object to be tracked is selected by a rectangular window inside which a few numbers of particles are scattered. The particles' weights are calculated based on the similarity between feature vectors of the scattered particles and that of the central particle. Before the resampling stage of PF algorithm, particles with the highest weights are evolved using a genetic algorithm. The evolved particles' coordinates are transferred to the next frame by a random walk model, and the rectangle involving new particles is specified. Moreover, we utilize the idea of partitioning (selecting parts of target in the first frame with a distinct color/texture) and reducing image size to decrease the number of particles. The partitioning idea also helps our method in resolving the occlusion problem. Simulation results demonstrate the outperformance of the suggested approach comparing with other methods in terms of precision and tracking time when it encounters with the challenges such as full and partial occlusions, illumination and scale variations, fast motions, and color similarity between the object and background.

***doi**: 10.5829/ije.2019.32.07a.03*

1. INTRODUCTION

The visual tracking process is the estimation of time variant positions of a moving object in a video sequence based on some measurement (observation) information [1-9]. Object tracking systems are applicable in various fields such as surveillance systems [3,4,10], human-computer interactions [3,4], driving assistance [4] and etc. In these tracking systems, there are a lot of problems to be surmounted of which we can note partial and full occlusion [3,5,7], illumination variation [3,4,9], dynamic background [7], irregular movement [7], complex scene [7], sudden and fast motion [3], target scale change and rotational errors.

Utilizing adaptive filters is a dominant solution for visual tracking problems. In the case of linear models and Gaussian noises, Kalman filtering approaches have been employed to track moving objects in video sequences [1,6,10].

Particle filtering (PF) is the other significant technique for tracking moving objects [3,5], which is originated from Monte Carlo integration method and is mainly used in Bayesian probability estimation in nonlinear stochastic systems [5,7,11]. In the PF-based visual tracking systems, as there are only a few particles to display the probability density function (PDF) of the object's correct state, the degeneracy of particles occurs in practice [3-6]. Researchers have proposed a number of methods, such as resampling techniques and usage of appropriate proposal density to solve the degeneracy problem. However, sub-optimal sampling mechanisms used in the resampling techniques lead to the Sample Impoverishment (SI) problem [6]. Sample Impoverishment problem significantly affects the capability of particle filter to describe the correct state of the moving object [7].

Recently, evolutionary algorithms have been widely used to solve Sample Impoverishment problem caused by the resampling step [6,7]. Employing evolutionary

*Corresponding Author Email: *smy.moghadasi@gmail.com* (S. Sadegh Moghadasi)

algorithms prior to the resampling step leads to particles optimization in particle filter algorithms [7], reducing the computational load and increasing the precision and speed of tracking. Han et al. [8] have used particle filter with the Immune Genetic Algorithm (IGA-PF) to prevent Sample Impoverishment problem. Zhao and Li [9] have presented a particle filter method based on Particle Swarm Optimization resampling (PSO-PF) for visual tracking. Walia and Kapoor [6] have proposed a particle filter method to track targets in video sequences and used Improved Cuckoo Search to avoid sample impoverishment (ICS-PF). In this method, new particles (100 particles) were produced before the resampling step using levy flight. The tracking results show that ICS-PF method is able to handle target scale changes and rotational errors in object tracking and that is more efficient than the generic particle filter and the PSO-PF method [12].

Zhao et al. [11] have presented an improved particle filter based on Genetic Resampling (GRPF) In this algorithm, crossover and mutation operators are adopted to replace the strategy of resampling. The proposed algorithm improves the particles diversity, solves the Sample Impoverishment problem and demonstrates a better performance compared to the particle filter.

In this paper, we propose a novel hybrid approach called Reduced Particle Filter Genetic Algorithm (RPFGA). Both RPFGA and GRPF [11] methods rely on genetic resampling where crossover and mutation operators perform the resampling process. It is worth mentioning that GRPF is presented with 1000 particles, while our proposed RPFGA method is presented with 20 particles. Moreover, performance of the GRPF hybrid method [11] has been tested on mathematical nonlinear models, while our proposed RPFGA hybrid method is applied to image sequences. Mathematical non-linear models do not encounter challenges of a real-world application as exist in visual tracking methods. In image sequences, our proposed RPFGA tracker faces a variety of challenges, such as illumination variation, full and partial occlusions, scale variation, target and background colour variations, and sudden and fast motion. Therefore, the task of object tracking is highly difficult compared to the mathematical model-based tracking. To sum up, the proposed RPFGA method with fewer particles outperforms the GRPF [11] method in terms of both tracking precision and computational complexity in spite of the challenges mentioned above.

Comparing the proposed RPFGA hybrid method with IGA-PF, resampling is performed using mutation and crossover operators. Both IGA-PF and the proposed RPFGA methods are applied to the image sequences with 100 and 20 particles, respectively. In fact, the main advantage of the RPFGA method compared to the IGA-PF method is that using the proposed idea of partitioning our algorithm rapidly tracks the target with fewer particles. In the partitioning idea, the user (observer) specifies parts of the target in the first frame of a video sequence, precisely [6,8,9]. Utilizing this approach in the RPFGA method, we are able to track some parts of the target that have distinct colors from the other moving objects in the background and to overcome the occlusion problem.

In the previous works [6,8,9,11], the number of particles and tracking time have been major problems to realize real-time systems. In this paper, these problems are resolved by employing both partitioning idea and reducing image dimensions that lead to reducing the number of particles dispersed in the rigid rectangle surrounding the target. To sum up, the innovations involved in this study are 1) combining genetic algorithm with particle filter to resolve sample impoverishment problem in the resampling step, 2) reducing the image and target dimensions, and 3) partitioning the image and selecting some parts of the target in the first frame of a video sequence to increase tracking speed.

This paper is organized as follows. In section 2, the basic particle filter algorithm and its usage for object tracking are brought. Section 3 explains genetic algorithm principles and its cooperation with the PF algorithm to resolve sample impoverishment. The conducted simulations and experimental results are given in section 4. Finally, we concluded the paper in section 5.

2. PARTICLE FILTER ALGORITHM FOR VISUAL TRACKING

2. 1. The Generic Particle Filter Algorithm The basic policy of all particle filter methods is replacing integration by the sample mean to obtain the minimum variance estimation [11]. This goal is realized by importance sampling that is generating a set of random samples spreading in the state-space to approximate the posterior PDF [11]. The state-space dynamic model is defined by Equations (1) and (2):

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}, \boldsymbol{w}_{k-1}), \tag{1}$$

$$\boldsymbol{y}_k = h(\boldsymbol{x}_k, \boldsymbol{v}_k), \tag{2}$$

where $\boldsymbol{x}_k$ is the state vector at time instance k, $\boldsymbol{y}_k$is the measurement vector, and $\boldsymbol{w}_{k-1}$ and $\boldsymbol{v}_k$ are process and measurement noises, respectively. Also, f and h are known as nonlinear functions.

A set of N particles $\{\boldsymbol{x}_k^{(i)}, i = 1,2, \ldots, N\}$ is sampled from a known proposal density function $q(\boldsymbol{x}_k, \boldsymbol{y}_{1:k})$ that meets the requirements of a typical importance density function. The goal is to describe the posterior PDF $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$ that is approximated by Equation (3):

$$\hat{p}(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}) = \sum_{i=1}^{N} w_k^{(i)} \delta\left(\boldsymbol{x}_k - \boldsymbol{x}_k^{(i)}\right), \tag{3}$$

where $\delta(.)$ is the Dirac delta function, and $w_k^{(i)}$ are normalized importance weights expressed by Equation (4)

$$w_k^{(i)} = w_{k-1}^{(i)} \frac{p(y_k|x_k^{(i)})p(x_k^{(i)}|x_{k-1}^{(i)})}{q(x_k^{(i)}|x_{k-1}^{(i)},y_k)}, \quad (4)$$

In the case of using the optimal importance density function, the particle filter algorithm is very difficult to be realized. A simple and useful alternative method is choosing $p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k)$ as the suboptimal importance density function by Equation (5)

$$q\left(\boldsymbol{x}_k \middle| \boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{y}_k\right) = p\left(\boldsymbol{x}_k \middle| \boldsymbol{x}_{k-1}^{(i)}\right). \quad (5)$$

Through sampling from this importance density function, we have particles $\boldsymbol{x}_k^{(i)} \sim p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)})$. The particles' weights are calculated by $w_k^{(i)} = w_{k-1}^{(i)} p(\boldsymbol{y}_k|\boldsymbol{x}_k^{(i)})$ and are normalized by $\widetilde{w}_k^{(i)} = \frac{w_k^{(i)}}{\sum_{i=1}^{N} w_k^{(i)}}$.

The problem of sampling from the mentioned above suboptimal density function is increasing the variance of weights [13]. In fact, after a while, most of the particles will have normalized weights near to zero, and only one particle will have a great weight near to one. This problem is referred to as the degeneracy of particles [14,15]. Resampling is a method to solve the degeneracy problem by making the variance of weights get smaller. In the resampling step, samples with high weights are copied several times, and samples with low weights are removed [16]. Hence, the resampling step leads to another problem called Sample Impoverishment (SI). Due to the over-replication of high-weighted particles, the number of meaningful particles reduces, that leads to reducing the information capacity of the new particles set [7]. The number of effective particles $\widehat{N}_{eff}$, is calculated by Equation (6):

$$\widehat{N}_{eff} = \frac{1}{\sum_{i=1}^{N} (\widetilde{w}_k^{i})^2}. \quad (6)$$

In this paper, we present a novel method called Reduced Particle Filter Genetic Algorithm (RPFGA) to resolve the Sample Impoverishment problem.

2. 2.The Proposed Particle Filter For Visual Tracking

In this paper, first the user (observer) selects the target using a rigid rectangle in the first frame of an image sequence. Similar work carried out in literature [7,8]. Then, the center point of this rectangular window is calculated automatically and is considered as the center point of the target in the first frame. Finally, the particles are scattered in the rectangular window and tracking the target center is started.

After selecting the target and initializing the state vectors, the state vectors are updated using a state transition model. The motion of the moving object can be considered as a random walk model, a constant velocity model, a constant acceleration model, and similar models [17,18]. As the object movement between the consecutive frames is small, we employ a random walk model [7]. Let $\boldsymbol{x}_{k-1}$ be the two-dimensional target state vector in the frame $k-1$. The target state vector at time instance k is calculated by Equation (7)

$$\boldsymbol{x}_k = \boldsymbol{x}_{k-1} + \boldsymbol{u}_k, \quad (7)$$

where $\boldsymbol{u}_k$ is the particle displacement vector obtained by $\boldsymbol{u}_k = [r_1 \times \Delta_x, r_2 \times \Delta_y]^T$, and r_1 and r_2 are two uniformly-distributed random numbers in the range -1 to 1. Also, Δ_x and Δ_y are the rigid rectangle's dimensions in the direction of *x* and *y* coordinates, respectively.

The probability density function of observation $p(\boldsymbol{y}_k|\boldsymbol{x}_k^{(i)})$ depends on the statistical description of the target [19,20]. In this paper, color and texture properties are considered as the features in the observation model. The probability of observing the target is calculated using the texture and color feature vectors of the central particle (target center) and the particles around the central particle in each frame. The feature vectors of all particles are compared with that of the central particle using the Bhattacharyya distance. Particles with less distance to the central particle get more weights.

The Bhattacharyya coefficient (the similarity between two feature vectors $\boldsymbol{p}$ and $\boldsymbol{q}$) is calculated by Equation (8) [21]:

$$\rho[\boldsymbol{p}, \boldsymbol{q}] = \sum_{u=1}^{\beta} \sqrt{p^{u} q^{u}}, \quad (8)$$

where β is the dimension of two vectors, and q^{u} and p^{u} represent the u-th element in the corresponding feature vectors. The Bhattacharyya distance is therefore obtained by Equation (9)

$$D(\boldsymbol{p}, \boldsymbol{q}) = \sqrt{1 - \rho[\boldsymbol{p}, \boldsymbol{q}]}. \quad (9)$$

Finally, the joint probability distribution (the weight of each particle) is calculated according to Equation (10) [8]:

$$w_k^{(i)} \propto p\left(\boldsymbol{y}_k|\boldsymbol{x}_k^{(i)}\right) = \frac{1}{\sqrt{2\pi}\,\delta} e^{\frac{-D^2(\boldsymbol{p},\boldsymbol{q})}{2\delta^2}}. \quad (10)$$

Equation (10) determines a Gaussian PDF with the standard deviation of δ where the value of δ is set experimentally. Finally, the weights are normalized after calculating all particles' weights. In case that the effective sample number (N_{eff}) is lower than a threshold ($0.7 \times N$), the resampling before the next frame is performed to prevent the degeneracy problem. This procedure continues until the last frame.

3. USING GENETIC ALGORITHM FOR RESAMPLING IN PARTICLE FILTER

3. 1. Genetic Algorithm Principals and Its Usage

Genetic Algorithm was first introduced by Holland in

1975. The genetic algorithm is a randomized search technique based on the principle of natural selection, inheritance, and population optimization using the elitist selection, crossover and mutation [11]. The genetic algorithm generates a population of solutions (parents) among which good solutions have a higher chance of regeneration (child production), whereas bad solutions are less likely to be regenerated [11]. Good parents are selected by a roulette wheel, which is created based on the fitness of individuals, for crossover, mutation, and elitism. In our work, the weight of each particle is considered as the fitness of an individual.

Using the elitism operator, the genetic algorithm directly transfers the parents with the best fitness values (weights) to the next generation. This operator increases the efficiency of the genetic algorithm to reach an optimal global solution by preventing the loss of good solutions. In fact, the worst offspring produced by parents of the current generation are replaced by elite parents of the current generation.

Parents with higher weights (fitness values) are more likely to be selected for crossover and mutation. Assuming the probability of crossover is $P_C = 0.9$, first a random number U is generated in the interval [0 1]. If $U < P_C$, two particles $\boldsymbol{x}_k^{(i)}$ and $\boldsymbol{x}_k^{(j)}$ are randomly selected for crossover, that is performed by Equation (11) [11]

$$\tilde{\boldsymbol{x}}_k^{(i)} = \alpha\boldsymbol{x}_k^{(i)} + (1-\alpha)\boldsymbol{x}_k^{(j)}, \quad \tilde{\boldsymbol{x}}_k^{(j)} = \alpha\boldsymbol{x}_k^{(j)} + (1-\alpha)\boldsymbol{x}_k^{(i)}, \tag{11}$$

where we set the value of α experimentally to 0.5. After crossover, the new individuals $\left\{\tilde{\boldsymbol{x}}_k^{(i)}, \tilde{\boldsymbol{x}}_k^{(j)}\right\}$ replace the old individuals $\left\{\boldsymbol{x}_k^{(i)}, \boldsymbol{x}_k^{(j)}\right\}$.

Assuming the probability of mutation is $P_m = 0.1$, a random number U is generated in the interval [0 1]. If $U < P_m$, a particle is randomly selected for mutation, and the mutated particle substitutes the old particle. In our work, the mutation operation is carried out according to Equation (12) [11]

$$\tilde{\boldsymbol{x}}_k^i = \boldsymbol{x}_k^i + \boldsymbol{\eta}, \tag{12}$$

where $\boldsymbol{\eta} \sim N(0, \sigma\mathbf{I})$, σ is set to 0.15 and $\mathbf{I}$ is the identity matrix. We represent each particle by a chromosome, whose genes are texture and color features. The feature vector to represent the i^{th} particle is shown as $Feat.Vec_i = [h_i, s_i, v_i, M_i, V_i, R_i]^T$, where h_i, s_i, and v_i denote the hue, saturation, and value of the HSV color space, respectively. Also, M_i is the mean, V_i is the variance and R_i is the range of gray-scale pixels in a 9×9 window around each particle. The parameter setting of the used genetic algorithm is shown in Table 1. In this paper, particles are optimized by a genetic algorithm before the resampling process. Thus, the number of effective particles increases, and consequently the correct target state is estimated more accurately [11].

3. 2. The Proposed Approaches In The Hybrid PF Tracker

In this paper, we apply two ideas, which are partitioning and reducing image size, to increase the speed of tracking as well as using a genetic algorithm to surmount sample impoverishment. Therefore, we call the proposed method Reduced Particle Filter Genetic Algorithm (RPFGA). By reducing the image size, the rigid rectangle involving the target gets smaller that it results in reducing the number of dispersed particles and processing time of each frame.

Using the idea of partitioning, the user (observer) specifies parts of the target precisely in the first frame of a video sequence [6,8,9]. Therefore, it is possible to track parts of the target that have distinct colors from the other moving objects in the background. In fact, partitioning shrinks the common area between the target and the other objects. Through reducing the number of particles, the partitioning idea leads to reducing the processing time of each frame and increasing speed of the proposed RPFGA algorithm.

It should be noted that the precise selection of the target in the first frame, which is required to be performed fast by the user, is very important. If the target is not selected precisely in the first frame, the starting point (target center) is not calculated correctly and RPFGA fails in target tracking in initial frames. In fact, in the proposed RPFGA, the user plays an important role for target selection in the first frame.

Setting proper values for the parameters of RPFGA method has also a great effect on the performance of the tracking algorithm. In the proposed RPFGA method, the number of particles (the population size of GA) is set to 20, and the number of iterations is set to 4. In the proposed hybrid approach, by setting the elitism rate to 0.3 the number of particles decreases that results into reducing implementation cost of genetic algorithm and fast converging to the global solution in spite of small population size and a low number of iterations.

The proposed RPFGA algorithm is explained in the following:

1. The user selects the target in the initial frame by a rectangular window. The target center position is

TABLE 1. The parameter setting of the employed GA

Initial population	Random generation
Fitness function	$p(\boldsymbol{y}_k \vert \boldsymbol{x}_k^{(i)}) = \frac{1}{\sqrt{2\pi}\,\delta} e^{\frac{-D^2(\boldsymbol{p},\boldsymbol{q})}{2\delta^2}}$
Representation	$[h_i, s_i, v_i, M_i, V_i, R_i]^T$
Parent selection	Roulette wheel
Replacement	(μ+ʎ)

calculated automatically and its feature vector is extracted. The initial particle set $\left\{\boldsymbol{x}_1^{(i)}, w_1^{(i)}\right\}_{i=1}^{N}$ is established by dispersing N particles around the central particle and their feature vectors are extracted. The initial particles' weights are considered equally as $w_1^{(i)} = 1/N$.

2. **For** k= 2 to "*the number of image sequences"*

- The particles' position vectors $\left\{\boldsymbol{x}_k^{(i)}\right\}_{i=1}^{N}$ are updated based on the state transition model and the target center is calculated. The particles' weights are calculated according to equation (10) and are normalized using equation (6).
- The number of effective samples N_{eff} is obtained using equation (7).
- **If** Neff $< \ N \times 0.7$

  The genetic algorithm is run for a number of four generations: use mutation ($P_m = 0.1$), heuristic crossover ($P_c = 0.9$), and elitism selection (with the rate of 0.3) to generate new individuals. The weights of particles (fitness functions) are recalculated at each generation.

  End **If**

End **For**

4. SIMULATIONS AND EXPERIMENTAL RESULTS

To evaluate the proposed RPFGA and the other tracking methods, the target's motion paths obtained by tracking approaches are compared with the real path extracted by an expert. The test data includes OneLeaveShopReenter2Cor.mpg, OneShopOneWait1cor.mpg,oneShopOneWait2cor.mpg, OneStopEnter1cor.mpg, and OneStopEnter2cor.mpg, that are subsets of CAVIAR [22] and two sequences from CHEHOVIN [23], and PETS'2000 [24] datasets. All experiments are conducted using a Corei3 processor with 4 GB RAM in MATLAB 2014 software. Table 2 shows the characteristics of each video sequence in the conducted experiments.

As the first experiment, we evaluate the effect of elitism rate in the tracking process of RPFGA method. The simulation is performed on 183 frames of the sequence OneShopOneWait1cor.mpg, and processing time of RPFGA method is compared with the PF tracker in Table 3. As shown in Table 3, the processing time of the proposed RPFGA method with 20 particles and elitism rate of 0.3 (6 elites) shows a notable reduction compared to the PF tracker with 300 and 500 particles. Also, the average error of RPFGA method with 20 particles and elitism rate of 0.3 is the second lowest one. To sum up, through setting a proper elitism rate the number of particles and tracking time can be reduced. In the next simulations, the elitism rate is set to 0.3 with 6 elites and 20 particles, as these parameters have led to the best results according to Table 3.

TABLE 2. The characteristics of the used video sequences

Video sequence	Num. of frames	Frame size	Challenges
OneShopOneWait1cor.mpg	183	288×384	Partial occlusion, illumination variation, scale variation
OneShopOneWait2cor.mpg OneStopEnter1cor.mpg OneStopEnter2cor.mpg	100	288×384	Partial occlusion, illumination variation, scale variation
OneLeaveShopReenter2cor.mpg	328	288×384	Full and partial occlusion, illumination and scale variations, color similarity between object and background
CHEHOVIN	113	180×320	Fast motions, color similarity between object and background
PETS'2000	320	288×384	Partial occlusion, color similarity between object and background

TABLE 3. The average Euclidean distance (average error) and processing time of RPFGA and PF methods on the sequence OneShopOneWait1cor.mpg in terms of different elitism rate and the number of particles

Tracking method	Num. of particles	Elitism rate (6)	Process. time per frame (s)	Average error
PF	100	-	2.2	37.7
PF	300	-	5.3	47.2
PF	500	-	10.9	152.02
RPFGA	20	0.3	**2.4**	2.7
RPFGA	30	0.2	3.5	3.3
RPFGA	40	0.15	4.4	3.8
RPFGA	50	0.1	5.3	**2.5**
RPFGA	60	0.1	6.1	33.02

The second sequence, One Leave Shop Reenter2cor.mpg, includes a video from a shopping center. The original sequence consists of 575 frames with the dimension of 288 × 384 pixels, captured by large lens from two different views. We used the idea of partitioning to track the target in this sequence. The attained tracking results are shown in Figure 1. As indicated in Figure 1(a), the PF method gradually loses the target, due to the problem of occlusion and great Sample Impoverishment. In Figure 1(b), employing the partitioning idea in the first frame, the occlusion problem is completely eliminated. The precision of tracking algorithms in terms of Euclidean distance are presented in Figure 2, and the detection results are brought in Table 4. The average errors reported in Table 4 are obtained by calculating the mean of Euclidean distance between the actual path and the results of RPFGA and PF methods.

The third sequence includes a scenario of movements of a male gymnast on the gymnastics mat from CHEHOVIN. In this sequence, the significant challenges are sudden and fast motions and similarity between the target's color and background. This sequence includes 567 frames with the image dimension of 180 × 320 pixels. We selected a subset of 113 frames in which sudden and fast motions are high and therefore the tracking process becomes more challenging.

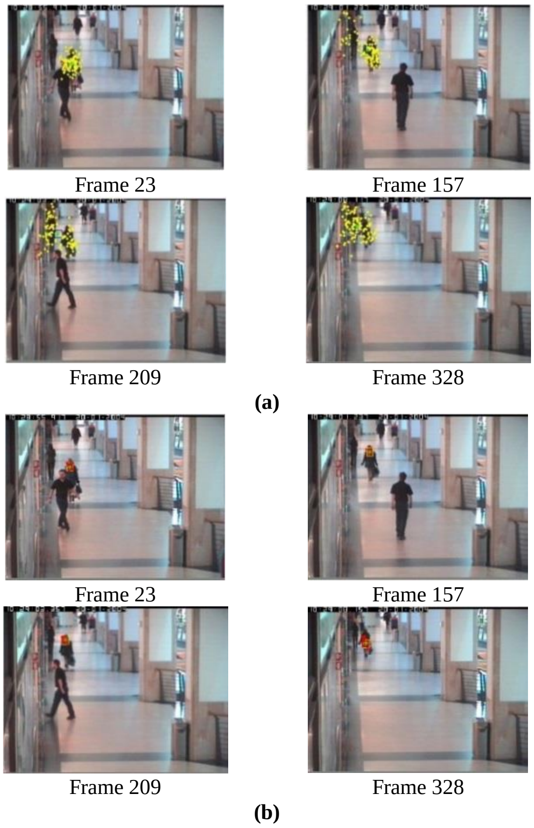

Figure 1. The obtained results on the second sequence, OneLeaveShopReenter2cor.mpg, (a) PF tracker; and (b) our proposed RPFGA tracker

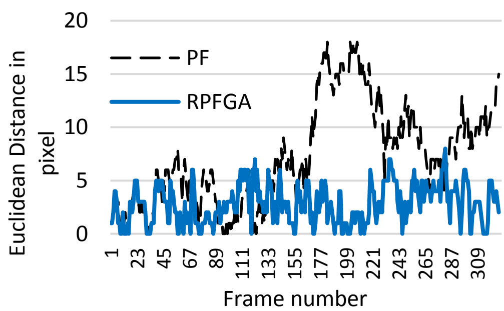

Figure 2. Comparing tracking performance of the proposed RPFGA method with 20 particles with a particle filter with 100 particles for OneLeaveShopReenter2cor.mpg sequence

TABLE 4. The obtained tracking results for 328 selected frames from OneLeaveShopReenter2cor.mpg

Tracking method	Frames per second	Num. of false detections	Num. of missed detections	Process. time per frame (s)	Average error
PF (100)	25	33	84	2.5	7.3
RPFGA (20)	25	0	0	1.6	2.7

As shown in Figure 3(a), due to sudden changes in the moving direction of the target and also color similarity between the target and background, the original PF algorithm lacks the target orientation and often jumps to some areas of the background. In contrast, the proposed RPFGA method (without partitioning) efficiently tracks the target along the sequence.

The fourth test sequence is selected from PETS'2000 whose tracking results are shown in Figure 4. PETS'2000 includes a scenario of a human movement recorded by a fixed camera with the challenges such as partial occlusion, and complex background. This sequence is of 412 frames, with the actual image size of 758 × 576 pixels. However, in the conducted experiments, we reduced the image dimensions to 384×288 pixels. In this sequence, the person in the frames from 80 to 100 is under partial cover with a light pole. In Figure 4(a), the PF tracker is able to detect the target in the frames from 80 to 100 where the object is under partial cover, but after a while, the target is not detected due to the complex background. In Figure 4(b), our proposed RPFGA method (without partitioning) is capable of detecting the

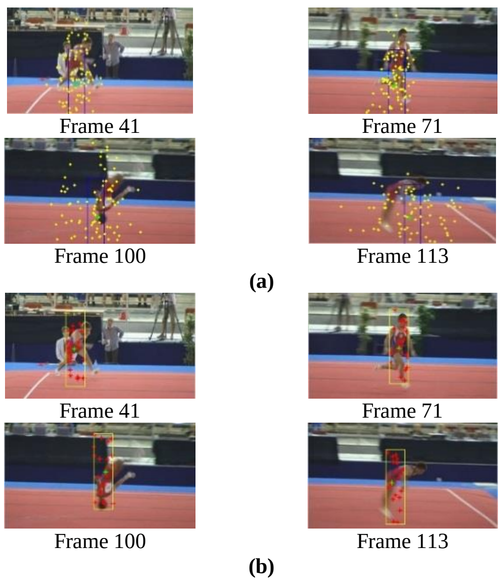

Figure 3. Tracking results on the third sequence from CHEHOVIN, (a) PF tracker; (b) RPFGA tracker

target in the frames from 80 to 100, where the object is under partial cover. Moreover, the proposed RPFGA with 20 particles is able of detecting the target up to the 320th frame. Figure 5 illustrates the fitness values of individuals in GA before resampling phase in the PF-based tracking on PET's 2000. The best fitness value curve (dotted line) along with the mean fitness value curve (solid line) was plotted in this Figure. As we employed ga(.) function of MATLAB that considers a function to be minimized, the fitness function is the minus of particle weights.

To show the high accuracy and efficiency of the proposed RPFGA method, in Figure 6 we compare it with some approaches such as ICS-PF [6], PSO-PF [12], and PF [5]. Also, both RPFGA and PF methods are tested with and without the idea of partitioning. The researchers [6] have tested the ICS-PF and PSO-PF methods on 100 frames of some sequences from CHAVIAR dataset: OneStopEnter1cor.mpg,OneShopOneWait2cor.mpg,OneShopOneWait1cor.mpg, and OneStopEnter2cor.mpg.

We have selected the same 100 frames of these sequences. Moreover, all methods are compared together in terms of the number of particles and harmonic mean whose results are shown in Table 5. The harmonic mean is defined as in Equation (13) [27]

$$F - \text{measure} = \frac{1}{\alpha \times \frac{1}{P} + (1-\alpha)\frac{1}{R}} \tag{13}$$

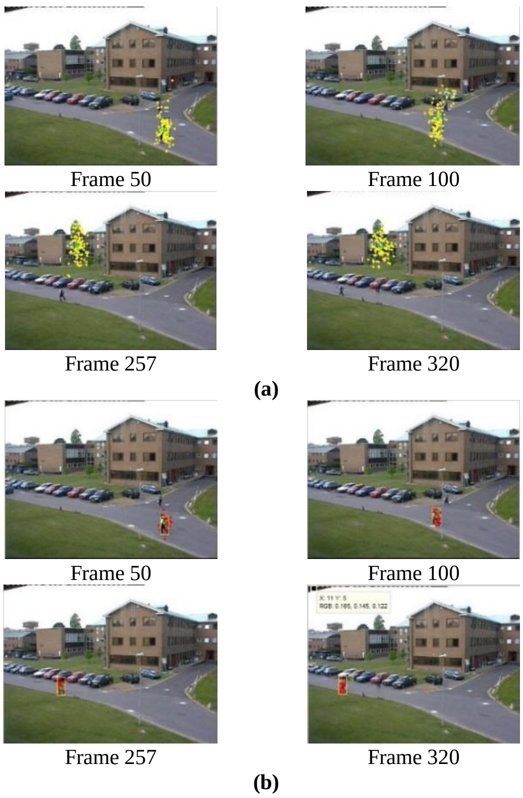

Figure 4. The obtained results for the fourth sequence from PETS'2000, (a) PF tracker; and (b) RPFGA tracker

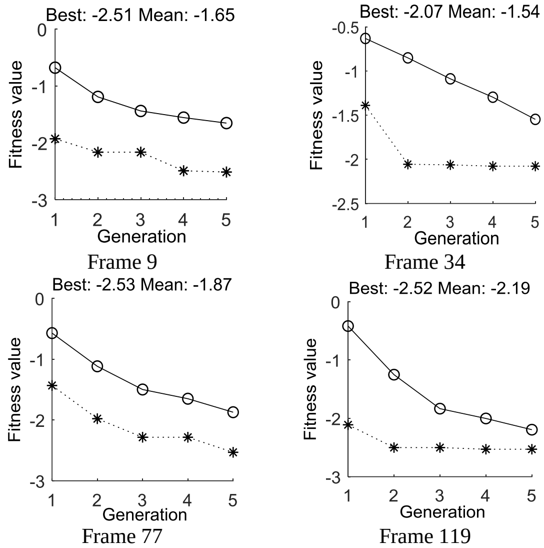

Figure 5. The fitness values of individuals before the resampling step of PF tracker in 5 generations. The best individual's fitness value and the average fitness value of 20 individuals tested on PETS'2000 dataset are shown as the dotted and solid lines, respectively

where $\alpha = 0.5$ leads to a balanced F-measure, and P and R are the precision and recall rate, respectively which are calculated by Equations (14) and (15) [25].

$$P = \frac{TP}{TP+FP} \tag{14}$$

$$R = \frac{TP}{TP+FN} \tag{15}$$

In these formulas, TP (True Positive) denotes the number of correct detections (the object is present and the result of tracking system is the same as that of the ground truth), FP (False Positive) is the number of false alarms (the detector contains at least one object, but the ground truth contains no object), and FN (False Negative) represents the number of cases that the ground truth has at least one object, but the tracker contains no object [6].

As shown in Table 5, the F-measure values of RPFGA and PF methods have been increased by using the partitioning idea compared to the RPFGA, PF, ICS-PF, and PSO-PF in which the idea of partitioning has not been employed. Employing GA with a proper elitism rate in the proposed RPFGA method along with the partitioning idea provided using fewer particles (20 particles) compared to ICS-PF, PSO-PF and PF methods and consequently reduced the tracking time. Moreover, utilizing the partitioning idea in the proposed RPFGA approach was helpful in eliminating the occlusion problem, in contrast to the PF method that failed to handle the challenge of full occlusion. To sum up, the RPFGA approach, due to the precise adjustment of elitism rate and using the partitioning idea, drastically outperforms GRPF and IGA-PF methods in terms of the number of particles and processing time.

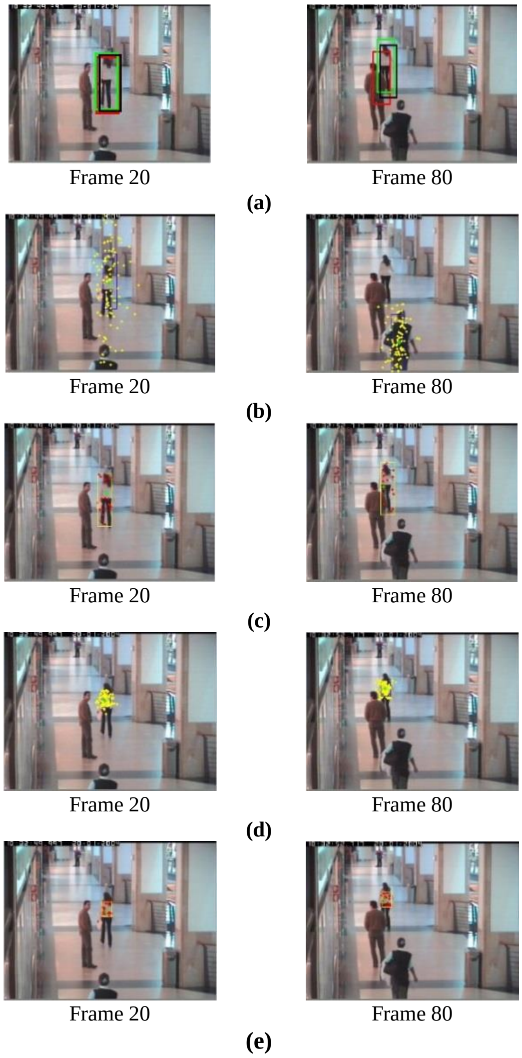

Figure 6. Comparison of the proposed RPFGA method with some hybrid methods, (a) PF (red)[5], PSO-PF (green) [12], ICS-PF (black) [6]; (b) PF without partitioning [5]; (c) RPFGA without partitioning; (d) PF with partitioning; and (e) RPFGA with partitioning

TABLE 5. Comparison of object tracking algorithms in terms of *F*-measure [6] and the number of particles

Tracking method	Num. of particles	F-Measure
PF without partitioning [5]	100	0.49
PF with partitioning	100	0.92
PSO-PF [12]	100	0.53
ICS-PF [6]	100	0.58
RPFGA without partitioning	20	0.87
RPFGA with partitioning	20	**0.99**

5. CONCLUSION

The tracking results showed the outperformance of the proposed RPFGA method compared to the generic PF and some hybrid methods including ICS-PF, IGAPF, PSO-PF and GRPF in terms of the number of particles and processing time. The reasons behind this improvement are 1) employing a GA in the PF approach, 2) precise adjustment of the elitism rate and 3) utilizing the idea of partitioning. In the proposed RPFGA algorithm, the genetic algorithm was applied before the resampling step, leading to an increase in the number of effective particles, which prevents the deviation of the detector from the target. On the other hand, the generic PF gradually diverted from the target in the challenges (such as partial and full occlusion, and etc.) due to the Sample Impoverishment. In addition, the precise selection of the target in the first frame (partitioning) increased the power of the proposed RPFGA algorithm.

It is possible to propose a simultaneous tracking method for multiple objects using RPFGA method as a future work.

6. REFERENCES

1. Toloei, A., Niazi, S., "Estimation of LOS rates for target tracking problems using EKF and UKF algorithms-a comparative study", ***International Journal of Engineering, Transactions B: Applications***, Vol. 28, No. 2, (2015), 172-178.
2. Mahdavi, M., Shahrouzi, S.N., Hasanzadeh, R., "A novel method for tracking moving objects using block-based similarity", ***International Journal of Engineering, Transactions B: Applications***, Vol. 22, No. 1, (2009), 35-42.
3. Li, S., Zhao, S., Cheng, B., Zhao, E., Chen, J., "Lightweight Particle Filter for Robust Visual Tracking", ***IEEE Access***, Vol. 6, (2018), 32310-32320.
4. Elafi, I., Jedra, M., Zahid, N., "Tracking objects with co-occurrence matrix and particle filter in infrared video sequences", ***IET Computer Vision***, Vol. 12, No. 5, (2018), 634-639.
5. Gordon, N.J., Salmond, D.J., Smith, A.F., "Novel approach to nonlinear/non-Gaussian Bayesian state estimation", IEE Proc. F-Radar and Signal Process., Vol. 140, No. 2, (1993), 107-113.
6. Walia, G.S., Kapoor, R., "Intelligent video target tracking using an evolutionary particle filter based upon improved cuckoo search", ***Expert Systems with Applications***, Vol. 41, No. 14, (2014), 6315-6326.
7. Gao, M.L., Li, L.L., Sun, X.M., Yin, L.J., Li, H.T., Luo, D.S., "Firefly algorithm based particle filter method for visual tracking", ***Optik-International Journal for Light and Electron Optics***, Vol. 126, No. 18, (2015), 1705-1711.
8. Han, H., Ding, Y.S., Hao, K.R., Liang, X., "An evolutionary particle filter with the immune genetic algorithm for intelligent video target tracking", ***Computers & Mathematics with Applications***, Vol. 62, No. 7, (2011), 2685-2695.
9. Zhao, J., Li, Z., "Particle filter based on Particle Swarm Optimization resampling for vision tracking", ***Expert Systems with Applications***, Vol. 37, No. 12, (2010), 8910-8914.

10. Zhou, Z., Wu, D., Zhu, Z., "Object tracking based on Kalman particle filter with LSSVR", ***Optik***, Vol. 127, No. 2, (2016), 613-619.
11. Zhao, B., Hu, J.W., Ji, B., "An improved particle filter based on genetic resampling", International Conference on Automation, Mechanical Control and Computational Engineering, China, (2015).
12. Tong, G., Fang, Z., Xu, X., "A Particle Swarm Optimized Particle Filter for Nonlinear System State Estimation", International Conference on Evolutionary Computation, Vancouver, Canada, (2006).
13. Hlinka, O., Hlawatsch, F., Djurić, P.M., "Likelihood consensus-based distributed particle filtering with distributed proposal density adaptation", ICASSP, (2012).
14. Hlinka, O., Sluciak, O., Hlawatsch, F., Djuric, P.M., Rupp, M., "Likelihood consensus and its application to distributed particle filtering", IEEE Trans. on Signal Process., Vol. 60, No. 8, (2012), 4334-4349.
15. Pocock, J.A., Dance, S. L., Lawless, A.S., "State estimation using the particle filter with mode tracking", ***Computers & Fluids***, Vol. 46, No. 1, (2011), 392-397.
16. Gao, M., Zhang, H., "Sequential Monte Carlo methods for parameter estimation in nonlinear state-space models", ***Computers & Geosciences***, Vol. 44, (2012), 70-77.
17. Kristan, M., Kovacic, S., Leonardis, A., Pers, J., "A two-stage dynamic model for visual tracking", IEEE Trans. on Systems, Man, and Cybernetics, Part B (Cyber.), Vol. 40, No. 6, (2010), 1505-1520.
18. Del Bimbo, A., Dini, F., "Particle filter-based visual tracking with a first order dynamic model and uncertainty adaptation", ***Computer Vision and Image Understanding***, Vol. 115, No. 6, (2011), 771-786.
19. Comaniciu, D., Ramesh, V., Meer, P., "Kernel-based object tracking", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 25, No. 5, (2003), 564-577.
20. Park, J., Anh, L.T., Lee, G., "Knowledge discovery in signal processing and multi-media color image segmentation using adaptive mean shift and statistical model-based method", In International Conference on Fuzzy Systems and Knowledge Discovery, Vol. 4, (2007).
21. Shen, C., Kim, J., Wang, H., "Generalized kernel-based visual tracking", IEEE Trans. on Circuits and Systems for Video Technology, Vol. 20, No. 1, (2010), 119-130.
22. Fisher, R., "*CAVIAR Test Case Scenarios",* http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1, (2007).
23. Zajc, L.C., "*Sequence gymnastics"*, From the ICCV2011 paper: An adaptive coupled-layer visual model for robust visual tracking. http://www.vicos.si/Research/LocalGlobalTracking, (2011).
24. PETS'2000: *Outdoor people and vehicle tracking (single camera).* http://www.cvg.rdg.ac.uk/slides/pets.html.
25. Lazarevic-McManus, N., Renno, J.R., Makris, D., Jones, G.A., "An object-based comparative methodology for motion detection based on the F-Measure", ***Computer Vision and Image Understanding***, Vol. 111, No. 1, (2008), 74-85.

An Efficient Target Tracking Algorithm Based on Particle Filter and Genetic Algorithm

S. Sadegh Moghadasi[a], N. Faraji[b]

[a] Artificial Intelligence Group, Faculty of Computer Engineering and Information Technology, Islamic Azad University, Qazvin, Iran
[b]. Electrical Engineering Department, Imam Khomeini International University, Qazvin, Iran

P A P E R I N F O

Paper history:
Received 08 January 2019
Received in revised form 14 March 2019
Accepted 02 May 2019

Keywords:
Object Tracking
Particle Filter
Genetic Algorithm
Sample Impoverishment
Resampling

چکیده

در این مقاله، ما یک روش کارای ترکیبی مبتنی بر فیلتر ذره‌ای برای ردیابی تصاویر ویدیویی پیشنهاد می‌دهیم که در آن از الگوریتم ژنتیک برای حل مساله تنوع ذرات استفاده شده است. در روش ارایه شده، هدف مورد ردیابی توسط یک پنجره مستطیلی که داخل آن تعداد کمی ذره پخش می‌شود انتخاب شده و سپس وزن ذرات براساس تشابه بردارهای ویژگی ذرات پراکنده شده و بردار ویژگی ذره مرکزی محاسبه می‌شود در مرحله بازنمونه برداری الگوریتم فیلتر ذره‌ای ، ذرات با بیشترین وزن انتخاب شده و توسط الگوریتم ژنتیک تکامل می‌یابند. مختصات ذرات تکامل یافته توسط یک مدل گام تصادفی به فریم بعدی منتقل شده و مستطیل شامل ذرات جدید تعیین می‌گردد. به علاوه، ما با استفاده از ایده تقسیم بندی (انتخاب دقیق بخشی از هدف با رنگ/بافت متمایز در اولین فریم) و کاهش ابعاد تصویر، تعداد ذرات و لذا زمان ردیابی کلی را کاهش می‌دهیم. ایده تقسیم‌بندی همچنین به روش پیشنهادی در حل مساله تداخل کمک می‌کند. نتایج شبیه سازی، عملکرد بهتر روش ترکیبی پیشنهادی را نسبت به روشهای دیگر به لحاظ دقت و زمان ردیابی نشان می‌دهد، حتی زمانی که با چالشهایی نظیر تداخل کامل یا نیمه کامل تغییرات روشنایی و مقیاس، حرکات سریع و ناگهانی و همچنین تشابه رنگ بین هدف و پس زمینه مواجه می‌شود.

doi*: 10.5829/ije.2019.32.07a.03*