Firefly Algorithm-Based Particle Filter Postprint with Adaptive Attraction Radius

Authors: Wang Hangxing, Pan Wei

Date: 2018-10-11T00:00:00+00:00

Abstract

To address the limitations of particle filtering algorithms, particularly their substantial demand for particle numbers, a particle filter based on an improved firefly algorithm is proposed. First, observation information is integrated into the firefly brightness formulation to enhance tracking accuracy. Second, an adaptive attraction radius parameter is devised to regulate the attraction range during firefly swarm optimization, thereby improving the algorithm' s real-time capability. Finally, the iterative optimization mechanism of the firefly algorithm is leveraged for particle updating. Comparative experimental results demonstrate that the proposed algorithm achieves improvements in both tracking precision and computational efficiency, confirming its ability to maintain target tracking accuracy and real-time performance even under conditions of reduced particle count.

Full Text

Particle Filter Based on Firefly Algorithm with Adaptive Attraction Radius

Wang Hangxing[1] ,[1] **, Pan Wei**[1] ,[1]
[1]. Capital Normal University
a. College of Information Engineering
b. Beijing Advanced Innovation Center for Imaging Technology, Beijing 100048, China

Abstract

To address the drawbacks of particle filter algorithms, such as their heavy reliance on a large number of particles, this paper proposes a particle filter based on an improved firefly algorithm. First, observation information is introduced into the firefly brightness formula to enhance tracking accuracy. Second, an adaptive attraction radius parameter is proposed to control the attraction range during firefly swarm optimization, thereby improving real-time performance. Finally, the iterative optimization of the firefly algorithm is employed for particle updating. Comparative experiments demonstrate that the proposed algorithm achieves improvements in both tracking accuracy and runtime, indicating that it can ensure accurate and real-time target tracking even with a small number of particles.

Key words: adaptive attraction radius; iterative optimization; target tracking; particle diversity; relative brightness

0 Introduction

Over 80% of the information humans acquire in daily life comes from vision. With the rapid development of computer vision, target tracking based on video sequences has become an important task in computer vision applications, widely used in video surveillance, vehicle tracking, gesture recognition, and many other fields [?, ?]. Traditional target tracking primarily employs Kalman filtering based on linearization and Gaussian assumptions. However, in practical applications, numerous nonlinear systems or non-Gaussian noise systems exist, making Kalman filtering clearly inadequate for system requirements [?]. Particle filtering based on Monte Carlo methods [?] can effectively solve tracking problems in nonlinear and non-Gaussian systems without requiring linear Gaussian assumptions, thus gaining widespread application [?, ?, ?].

Nevertheless, despite its advantages in target tracking—such as using Monte Carlo methods to convert integration operations into summation operations—particle filtering has several drawbacks. It requires a large number of particles to ensure tracking accuracy, leading to poor real-time performance and particle depletion problems after resampling. Therefore, particle filtering based on swarm intelligence optimization algorithms represents a promising new direction. These algorithms reduce the demand for particle numbers through iterative optimization of particle distributions and fundamentally solve the particle depletion problem by avoiding the discarding of low-weight particles.

Zhang et al. [?] utilized particle swarm optimization (PSO) to optimize the particle set before resampling, enabling particles to be distributed in high-likelihood regions as much as possible and effectively preventing the reduction of particle diversity after resampling. Wang and Heris et al. [?, ?] applied ant colony algorithms in the resampling and initialization sampling stages to optimize particle filtering for target tracking. Although these methods effectively solved particle impoverishment and degradation problems, they have not been widely adopted due to the strong empirical nature of their search processes and termination conditions. Wang et al. [?] used an improved genetic algorithm as a resampling method, designing crossover and mutation probabilities that vary proportionally with particle degradation to control sampling of new particles. While this approach improved particle effectiveness and diversity, determining the appropriate number of iterations and thresholds remains challenging.

Compared with the above algorithms, the firefly algorithm, as one of the latest swarm intelligence optimization algorithms, offers higher convergence accuracy, faster convergence speed, and easier implementation. However, there is currently limited work on optimizing particle filtering with firefly algorithms both domestically and internationally. Zhu et al. [?] stratified particle sets according to weight degradation degree and mapped optimized layer particles to high-likelihood regions, but this approach cannot truly solve the particle depletion problem. Gao et al. [?] implemented a firefly algorithm with an improved attractiveness formula to distribute particle sets in high-likelihood regions for optimizing particle filtering, but the characteristics of the brightness calculation formula result in high computational complexity. Tian et al. [?] proposed a firefly algorithm optimized particle filter (FA-PF) that uses an improved position update formula for iterative optimization, avoiding local extrema during the optimization process and effectively controlling particle diversity attenuation. However, since particles need to interact pairwise during mutual attraction, this approach increases computational complexity and runtime.

To address these issues, this paper proposes a particle filter based on a firefly algorithm with adaptive attraction radius. By introducing an adaptive attraction radius parameter to control the attraction range during particle interaction, the problem of large particle numbers involved in computations and increased computational complexity is solved.

1 Particle Filter Algorithm

The key to particle filtering algorithms is using a set of weighted particles and noisy observations to estimate the posterior density function of the system state [?], thereby achieving system state estimation [?]. Particle motion follows a first-order autoregressive state space model [?], with mathematical representations of the state and observation models as follows:

$$x_k = f(x_{k-1}, u_{k-1}) \tag{1}$$

$$y_k = h(x_k, v_k) \tag{2}$$

where f and h are the state transition function and observation function, respectively; u_{k-1} and v_k are system noise and observation noise. The dynamic representation of the state and observation models [?] is shown in [Figure 1: see original paper].

Based on the particle states at time $k-1$, the particle states at time k are predicted and updated according to the state model, observation model, and Bayesian estimation theory. The particle state prediction is calculated as follows:

$$p(x_k|y_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1}$$

However, since integration operations are not easily implementable, particle filtering employs Monte Carlo methods to convert integration operations into summation operations over finite sample points. After prediction, particle states are updated based on observations at time k and Bayesian estimation. The importance weight of each particle at time k is calculated as shown in equation (5) and normalized as shown in equation (6):

$$w_k^{(j)} \propto w_{k-1}^{(j)} \frac{p(y_k|x_k^{(j)})p(x_k^{(j)}|x_{k-1}^{(j)})}{q(x_k^{(j)}|x_{k-1}^{(j)}, y_k)}$$

$$w_k^{(i)} = \frac{w_k^{(i)}}{\sum_{j=1}^{N} w_k^{(j)}}$$

As can be seen from the above calculations, all observation data are required during particle prediction and update. Over time, this causes a significant increase in computational load. Particle filtering uses sequential importance sampling (SIS) for recursive calculation.

Based on the above, particle filtering initializes state information for all particles at time $k = 0$. As time progresses, particle states are continuously predicted and updated, all particle weights are calculated and normalized, and the need for resampling is determined. If resampling is needed, the operation is performed (replicating high-weight particles and discarding low-weight particles) to obtain a new set of particle states; if not, the algorithm proceeds to the next time step' s prediction and update, achieving cyclic target tracking.

2 Firefly Algorithm

Swarm intelligence optimization algorithms achieve optimization by mathematically modeling the collective behavior of biological populations. As one such algorithm, the firefly algorithm primarily mimics the luminescent behavior of fireflies in nature, using brightness differences among individuals for mutual attraction to complete the optimization process. The mutual attraction between fireflies mainly relies on brightness and attractiveness.

2.1 Relative Brightness of Fireflies

The relative brightness of fireflies forms the basis for mutual attraction. During the attraction process, due to brightness differences between individual fireflies, brighter fireflies attract dimmer ones, causing the population to move toward the brightest firefly to complete optimization. The relative brightness formula is as follows:

$$I_{ij} = I_i e^{-\gamma r_{ij}^2}$$

where I_{ij} is the relative brightness of firefly j as perceived by firefly i; I_i is the brightness of the brightest firefly, obtained from the objective function value at the optimal firefly' s position; γ is the light absorption coefficient, primarily reflecting the characteristic that brightness weakens with increasing distance, typically taken as $\gamma \in [0.01, 100]$; and r_{ij} is the Cartesian distance from firefly i to firefly j.

2.2 Attractiveness of Fireflies

During mutual attraction, attractiveness measures the magnitude of attraction between firefly individuals. Combined with individual relative brightness, it completes position updates. The attractiveness formula is as follows:

$$\beta_{ij} = \beta_0 e^{-\gamma r_{ij}^2}$$

where β_{ij} is the attractiveness of firefly i to firefly j; β_0 is the maximum attractiveness, i.e., the attractiveness at the light source.

2.3 Firefly Position Update

Individuals in the firefly population update their position information through the process of attraction, continuously moving closer to the brightest firefly and distributing the population in brighter regions. The position update formula is as follows:

$$x_i = x_i + \beta_{ij}(x_j - x_i) + \alpha\varepsilon_i$$

Assuming firefly i is attracted by firefly j, firefly i updates its position according to the above formula. Where x_i is the position of firefly i; β_{ij} is the attractiveness of firefly j to firefly i; α is a constant, generally taken as $\alpha \in [0, 1]$; and ε_i is a random number obtained from Gaussian distribution, uniform distribution, or other distributions.

3.2 Particle Filter Based on Improved Firefly Algorithm

To address the problem of large particle numbers required for particle filter tracking, particle filters based on swarm intelligence optimization algorithms represent the main current solution. However, since particle swarm algorithms require large data storage and are not easily updated, and ant colony algorithms rely entirely on feedback information during optimization, resulting in slow speed, this paper proposes a particle filter based on a firefly algorithm with adaptive attraction radius.

3.2.1 Improvement of Firefly Relative Brightness Formula

The original firefly relative brightness formula (7) relies only on the brightest particle information and calculates relative brightness based on distances between particles. If this relative brightness is directly applied in combination with particle filtering to complete particle iterative optimization, the latest observation information cannot be utilized in real-time, leading to low algorithm accuracy. Meanwhile, the special nature of its calculation method introduces enormous computational load during algorithm operation.

To address this deficiency, this paper incorporates the latest observation information of particles to improve algorithm accuracy and reduce runtime computational load. The improved relative brightness mathematical expression is as follows:

$$I = (y_k - \hat{y}_k)^2$$

where I is the improved relative brightness; y_k is the particle observation value; $\hat{y}_k$ is the particle prediction value. When extending the relative brightness formula to multidimensional cases, the key is to calculate the squared spatial difference between observation and prediction values in multidimensional space. For two-dimensional coordinate information, the relative brightness formula should be:

$$I = (y_k - \hat{y}_k)^2 + (x_k - \hat{x}_k)^2$$

where (x_k, y_k) are the target observation coordinates; $(\hat{x}_k, \hat{y}_k)$ are the target prediction coordinates.

Compared with the original relative brightness formula, the improved formula uses the squared difference between each particle' s own observation and prediction values at each moment to represent relative brightness, eliminating the need to compare all particles with the optimal particle. This reduces computational load during iterative optimization. Simultaneously, by incorporating particle observation information, the difference between algorithm predictions and observations is reflected in real-time in the relative brightness, which affects the estimation of particle prediction values at the next moment, thereby effectively improving tracking accuracy.

3.2.2 Proposal of Adaptive Attraction Radius

The optimization process of the firefly algorithm involves pairwise brightness comparisons among all firefly individuals, where dimmer fireflies are attracted by brighter ones and update their position and brightness information. However, using the original attraction method for optimization leads to exponential growth in computational load as the firefly swarm size increases, causing significant time consumption. Therefore, this paper designs a new parameter—the attraction radius. The attraction radius primarily limits the attraction range and particle numbers during mutual attraction between firefly individuals. The radius size is determined by brightness, with brighter particles having larger attraction radii.

The relationship between firefly brightness and attraction radius is shown in [Figure 3: see original paper]. In [Figure 3: see original paper], a, b, and c represent three fireflies with different brightness levels, with brightness $a > b > c$ and attraction radii $r_a > r_b > r_c$. For firefly a and b, since $r_{ab} < r_a$ (i.e., firefly b is within firefly a' s attraction range), firefly b will be attracted by firefly a and update its position. However, for firefly c, despite having the lowest brightness, since it is outside firefly a' s attraction range, it will not be attracted by firefly a and will not update its position. Therefore, for two fireflies with different brightness levels, position update only occurs when the inter-firefly distance is less than the brighter firefly' s attraction radius; otherwise, no update occurs.

In this paper, the squared difference between particle observation and prediction values is used to represent relative brightness. Thus, a smaller relative brightness value indicates that the particle' s prediction is closer to its observation, nearer to the optimal particle, and the particle' s attraction radius should be larger. The attraction radius calculation formula is designed as:

$$r = \frac{10}{I}$$

where r is the particle' s attraction radius; I is the particle' s brightness value.

When the firefly algorithm with adaptive attraction radius performs mutual attraction between particles, it no longer requires comparing each particle with all other particles for brightness comparison and position update. Instead, particles only need to compare brightness with particles within their attraction radius to complete attraction and position update. This significantly reduces the number of particles involved in computations during iterative optimization, thereby decreasing the number of particles required for accurate tracking, effectively lowering computational complexity, and reducing algorithm runtime.

The particle description during target tracking using the proposed algorithm is shown in [Figure 2: see original paper].

3.3 Algorithm Steps

Algorithm 1 Particle Filter Based on Firefly Algorithm with Adaptive Attraction Radius

1. Initialize particle prediction values at time $k = 0$
2. For each time step k:
    a. Calculate particle i' s relative brightness and attraction radius
    b. Calculate mutual distances between particles i and j

c. If $r_{ij} < r_i$, perform position update for particle i
d. Update weights and normalize
e. Calculate effective particle number N_{eff}
f. If $N_{eff} \leq N_{th}$, perform resampling

3. End for

Traditional particle filter algorithms update particle sets using Bayesian estimation theory. The proposed algorithm introduces a firefly algorithm with adaptive attraction radius to update particle set information. After obtaining particle prediction and observation information, the firefly algorithm is used for iterative optimization of the particle set to complete particle state updates, distributing the particle set in high-likelihood regions for more accurate tracking.

4 Experiments

4.1 Experimental Setup

The experimental software and hardware environment is shown in .

** Experimental Software and Hardware Environment**

Component	Specification
Processor	Intel(R) i5
Memory	4.00 GB
Operating System	Windows 10
Software	MATLAB R2016a

The system state model and observation model used in experiments are as follows:

$$x_t = 0.5x_{t-1} + \frac{25x_{t-1}}{1 + x_{t-1}^2} + 8\cos(1.2(t-1)) + \omega_t \tag{3}$$

$$y_t = \frac{x_t^2}{20} + v_t \tag{4}$$

This system is a univariate non-stationary growth model, a typical nonlinear system that is difficult to track accurately using traditional particle filtering methods. The observation noise variance is set to $R = 1$, and the filtering duration is 50 time steps. In firefly algorithms, the maximum attractiveness is typically set in the interval $[0.8, 1]$; this paper sets the maximum attractiveness to 0.8 and the light absorption coefficient to 1.

4.2 Quantitative Analysis

Under identical software and hardware conditions, the proposed algorithm is compared with the standard particle filter (PF) and the firefly algorithm optimized particle filter (FA-PF) from reference [?]. Different algorithms are run under varying conditions, and results are recorded for comparison and analysis. All result data are averaged over 100 repeated experiments.

4.2.1 Accuracy Test This paper uses the Root Mean Square Error (RMSE) formula as the measure of error magnitude, i.e., prediction accuracy. RMSE measures the deviation between observed and predicted values, effectively reflecting prediction precision. The RMSE calculation formula is:

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(x_t - \hat{x}_t)^2}$$

where T is the tracking duration; $\hat{x}_t$ is the state mean.

From the RMSE formula, it is clear that when the state estimate is closer to the observation, the RMSE value is smaller, indicating higher tracking accuracy and better algorithm performance, and vice versa.

1) Algorithm Performance Test

The proposed algorithm is run alongside PF and FA-PF algorithms under different particle numbers ($N = 50, 100, 200$) and different process noise conditions ($Q = 1, 0.1$). The RMSE values for tracking under different conditions are recorded, with results shown in .

** RMSE Comparison of Different Algorithms Under Various Particle Numbers**

Condition	PF	FA-PF	Proposed	Improvement vs PF	Improvement vs FA-PF
$Q = 1$					
$N = 50$	4.3735	3.5640	3.3271	23.9%	6.6%
$N = 100$	4.0221	3.4576	3.1543	21.6%	8.8%
$N = 200$	3.2053	3.1507	2.9825	6.9%	5.3%
$Q = 0.1$					
$N = 100$	3.0060	2.9693	2.8555	5.0%	3.8%
$N = 200$	-	-	-	-	-

As shown in , the proposed algorithm achieves smaller RMSE values than the other two algorithms under different particle numbers and process noise conditions, demonstrating higher accuracy. Under different process noise conditions, the accuracy of all three algorithms improves as particle number N increases.

Additionally, when $Q = 1$, the proposed algorithm' s accuracy at $N = 100$ still exceeds that of FA-PF at $N = 200$; when $Q = 0.1$, the proposed algorithm' s accuracy at $N = 100$ approximates that of FA-PF at $N = 200$, indicating that the proposed algorithm maintains tracking accuracy even with fewer particles.

This is because the proposed algorithm uses the improved firefly algorithm to perform iterative optimization on the particle set before resampling, resulting in high weights for particles distributed in high-likelihood regions and reducing the number of particles needed for accurate tracking.

From the data in , the average improvement rates of the proposed algorithm relative to PF and FA-PF under different process noise conditions are calculated and shown in .

** Average Accuracy Improvement Rate of Proposed Algorithm Under Different Process Noise**

Process Noise	vs PF	vs FA-PF
$Q = 1$	22.6%	7.0%
$Q = 0.1$	5.7%	3.8%

When $Q = 1$, the proposed algorithm' s accuracy improves by 7% compared to the state-of-the-art FA-PF algorithm and by 22.6% compared to PF. When $Q = 0.1$, the proposed algorithm' s accuracy improves by 3.8% and 5.7% compared to FA-PF and PF, respectively. This improvement stems from incorporating the latest observation information during the firefly algorithm' s iterative optimization process to ensure filter precision.

2) Algorithm Tracking Effect Test

The proposed algorithm is run alongside PF and FA-PF algorithms under the same process noise but different particle numbers ($N = 50, 100$). The state estimation values and absolute error data for the three algorithms are recorded, with results shown in [Figure 4: see original paper] and [Figure 5: see original paper].

When $N = 50$ and $Q = 1$, the experimental estimation values and absolute errors are shown in Figure 4: see original paper and (b). When $N = 100$ and $Q = 1$, the experimental estimation values and absolute errors are shown in Figure 5: see original paper and (b).

From Figure 4: see original paper and Figure 5: see original paper, it can be observed that the proposed algorithm' s filter estimates are closer to the true values than those of PF and FA-PF under different particle numbers. This is reflected in Figure 4: see original paper and Figure 5: see original paper, where the proposed algorithm' s absolute errors are smaller than those of PF and FA-PF, demonstrating higher accuracy across different particle numbers.

This improvement primarily results from the proposed algorithm' s use of observation-information-integrated relative brightness calculation in the firefly algorithm. During particle set iterative optimization, particle positions are adjusted more accurately based on relative brightness, enabling particles to be distributed more precisely near their true values and thereby enhancing algorithm accuracy.

4.2.2 Real-time Test Under identical software and hardware conditions, different algorithms are run under varying particle numbers and process noise conditions, and their runtime is recorded for comparison, as shown in .

** Runtime Comparison of Different Algorithms Under Various Particle Numbers**

Condition	PF	FA-PF	Proposed	Reduction vs FA-PF
$Q = 1$				
$N = 50$	0.0101	0.0264	0.0226	14.02%
$N = 100$	0.0175	0.0629	0.0584	7.00%
$N = 200$	0.0404	0.1685	0.1515	10.09%
$Q = 0.1$				
$N = 50$	0.0093	0.0241	0.0218	9.63%
$N = 100$	0.0168	0.0587	0.0542	7.67%
$N = 200$	0.0387	0.1624	0.1498	7.76%

As shown in , under different particle numbers and process noise conditions, the proposed algorithm' s runtime is somewhat longer than PF, but combined with the data in , the proposed algorithm' s accuracy is substantially improved compared to PF. This is because the proposed algorithm uses particle set iterative optimization before resampling, resulting in high weights for most particles. The trade-off of increased runtime yields improved accuracy. Under different process noise conditions, all three algorithms' runtime increases with particle number.

From the data in , the average runtime reduction rate of the proposed algorithm relative to FA-PF under different process noise conditions is calculated and shown in .

** Average Runtime Reduction Rate of Proposed Algorithm Under Different Process Noise**

Process Noise	Average Reduction Rate
$Q = 1$	10.4%
$Q = 0.1$	7.7%

As shown in and , under different particle numbers and process noise conditions, the proposed algorithm' s runtime is reduced to some extent compared to FA-PF. The runtime is reduced by 10.4% and 7.7% on average compared to FA-PF when $Q = 1$ and $Q = 0.1$, respectively.

This reduction occurs because, during the iterative optimization process of mutual particle attraction using the firefly algorithm, the proposed adaptive attraction radius controls the attraction range, significantly reducing the number of particles involved in computations and thereby decreasing algorithm runtime.

4.3 Qualitative Analysis

To visually demonstrate the proposed algorithm' s tracking performance, experiments are conducted on videos from the mainstream Visual Tracking Benchmark dataset [?]. During tracking, color histograms are used as tracking features with particle number $N = 100$. The proposed algorithm' s tracking effect is compared with PF, with partial frames of tracking results shown in [Figure 6: see original paper]-[Figure 9: see original paper].

In the Coke video, the tracking scenario is relatively simple, the target' s shape does not change significantly, and the target moves in irregular circular motion—a typical target tracking problem in nonlinear systems. [Figure 6: see original paper] shows the complete scene of the video sequence, [Figure 7: see original paper] shows PF tracking results, [Figure 8: see original paper] shows FA-PF algorithm tracking results from reference [?], and [Figure 9: see original paper] shows the proposed algorithm' s tracking results.

From [Figure 7: see original paper]-[Figure 9: see original paper], it is evident that PF' s particle distribution is scattered and uneven, significantly reducing the number of meaningful tracking particles—an important cause of PF tracking failure. FA-PF' s particle distribution is more concentrated than PF' s, while the proposed algorithm' s particle distribution is the most concentrated, primarily around the center of the tracking bounding box. This is because the proposed algorithm uses the improved firefly algorithm to effectively perform iterative optimization on the entire particle set.

At frame 38, the tracking target moves to the back of the scene and undergoes partial occlusion; at frames 80 and 131, scattered particles cannot concentrate in high-likelihood regions, causing PF tracking failure. Compared with FA-PF, which basically achieves accurate tracking, the proposed algorithm uses the improved firefly algorithm to effectively perform iterative optimization on all particles, enabling the particle set to be distributed as much as possible near the target. The large number of particles contributing to tracking results in more accurate tracking.

Additionally, due to the introduction of the adaptive attraction radius, the proposed algorithm effectively controls the number of particles involved in optimization and reduces the iterative optimization process. As shown in , algorithm runtime is reduced by 7%.

5 Conclusion

Particle filtering algorithms require large numbers of particles to ensure accuracy. As weights continuously diminish, fewer particles remain for tracking, necessitating resampling of the particle set. However, this severely reduces particle diversity and may even result in only one particle type remaining, leading to tracking failure. This paper optimizes the particle filtering algorithm using an improved firefly algorithm.

Based on the characteristics of both algorithms, the firefly brightness formula is improved by incorporating particle observation information to distribute particle sets in globally optimal regions, ensuring algorithm accuracy. Simultaneously, an adaptive attraction radius parameter is proposed to prevent exponential growth in computational load as particle numbers increase, reducing algorithm complexity. Both quantitative and qualitative analyses demonstrate that the proposed algorithm effectively reduces the number of particles required during tracking, maintains accurate tracking even with fewer particles, and achieves a certain reduction in runtime compared to FA-PF.

References

[1] Yan Yue, Wang Jingling, Li Chuanzhen, et al. Object tracking using SIFT features in a particle filter [C]// Proc of IEEE International Conference on Communication Software and Networks. Piscataway, NJ: IEEE Press, 2011: 406-410.

[2] Wang Yuxia, Zhao Qingjie, Cai Yiming, et al. Tracking by auto-reconstructing particle filter trackers [J]. Chinese Journal of Computers, 2016, 39 (7): 1294-1406.

[3] Khan M W, Salman N, Ali A, et al. A comparative study of target tracking with Kalman filter, extended Kalman filter and particle filter using received signal strength measurements [C]// Proc of IEEE International Conference on Emerging Technologies. Piscataway, NJ: IEEE Press, 2016: 1-6.

[4] Lin Qing, Wang Xin. Improved particle filter algorithm based on target tracking [J]. Information Technology, 2017 (10): 88-92.

[5] Kwok N M, Fang Gu, Zhou Weizhen, et al. Evolutionary particle filter: re-sampling from the genetic algorithm perspective [C]// Proc of IEEE//RSJ International Conference on Intelligent Robots and Systems. Piscataway, NJ: IEEE Press, 2016: 2935-2940.

[6] Bao Bingji. Research of video target tracking algorithm based on particle filter [D]. Hefei: Anhui University, 2016.

[7] Wu Yanhai, Xie Xiamin, Han Zishuo. Research on target tracking algorithm

based on particle filter and Mean-Shift [J]. Applied Mechanics & Materials, 2014, 457-458: 1050-1053.

[8] Zhang Caiqian, Ge Lei, Han Dong. Research on particle swarm particle filter algorithm based on target tracking [J]. Computer Simulation, 2014, 31 (8): 392-396.

[9] Wang Fasheng, Lin Baowei, Li Xucheng. An ant particle filter for visual tracking [C]// Proc of IEEE//ACIS International Conference on Computer and Information Science. Piscataway, NJ: IEEE Press, 2017: 417-422.

[10] Herisn S M K, Khaloozadeh H. Ant colony estimator: an intelligent particle filter based on ACOR [J]. Engineering Applications of Artificial Intelligence, 2014 (28): 78-85.

[11] Wang W, Tan Q K, Chen J, et al. Particle filter based on improved genetic algorithm resampling [C]// Proc of Guidance, Navigation and Control Conference. Piscataway, NJ: IEEE Press, 2017: 346-350.

[12] Zhu Wenchao, Xu Dezhang. Improved particle filter algorithm based on artificial glowworm swarm optimization [J]. Application Research of Computers, 2014, 31 (10): 2920-2924.

[13] Gao Mingliang, Li Lili, Sun Xianming, et al. Firefly algorithm (FA) based particle filter method for visual tracking [J]. Optik-International Journal for Light and Electron Optics, 2015, 126 (18): 1705-1711.

[14] Tian Mengchu, Bo Yuming, Chen Zhimin, et al. Firefly algorithm intelligence optimized particle filter [J]. Acta Automatica Sinica, 2016, 42 (1): 89-97.

[15] Wang Fasheng, Lu Mingyu, Zhao Qingjie, et al. Particle filtering algorithm [J]. Chinese Journal of Computers, 2014, 37 (8): 1679-1694.

[16] Zhou Tianrun, Ouyang Yini, Wang Rui, et al. Particle filter based on real-time compressive tracking [C]// Proc of International Conference on Audio. Piscataway, NJ: IEEE Press, 2017: 754-759.

[17] Gonzalez M, Collet C. Robust body parts tracking using particle filter and dynamic template [C]// Proc of IEEE International Conference on Image Processing. Piscataway, NJ: IEEE Press, 2011: 529-532.

[18] Nobahari H, Raoufi M, Sharifi A. A heuristic filter based on firefly algorithm for nonlinear state estimation [C]// Proc of IEEE Symposium Series on Computational Intelligence. Piscataway, NJ: IEEE Press, 2017.

[19] Su Yingya, Zhao Qingjie, Zhao Liujun, et al. Abrupt motion tracking using a visual saliency embedded particle filter [J]. Pattern Recognition, 2014, 47 (5): 1826-1834.

[20] Wu Yi, Lim Jongwoo, Yang Ming-hsuan. Online object tracking: a benchmark [C]// Proc of Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE Press, 2013: 2411-2418.