A Novel Visual Tracking Method Based on Moth-Flame Optimization Algorithm

Huanlong Zhang(✉), Xiujiao Zhang, Xiaoliang Qian, Yibin Chen, and Fang Wang

College of Electric and Information Engineering,
Zhengzhou University of Light Industry, Zhengzhou, China
zhl_lit@163.com

Abstract. Moth-flame optimization algorithm (MFO) is a new meta-heuristic optimization algorithm that mimics the motion of moths flight around the flames. In this work, a novel MFO-based visual tracking method is proposed by interaction between flames and moths. Firstly, visual tracking is expressed as searching for target in whole search space. Then, the spiral flight of moths is employed to enhance search capabilities. And the mechanism of reducing the number of flames gradually is applied for improving the convergence rate of the proposed algorithm. Finally, an appearance model based HOG feature is established to measure the similarity between the target and the candidate samples. In light of the best confidence value, the tracking target is located. Extensive experiments have demonstrated the effectiveness of proposed tracker.

Keywords: Moth-flame · Visual tracking · Swarm optimal
HOG features

1 Introduction

Visual tracking is one of the fundamental problems in computer vision. It has long been playing a key role in numerous applications like visual surveillance, military reconnaissance, motion recognition, traffic monitoring et al. Although researchers have made significant progress over the past decade, there still exist many challenging problems, such as motion blur, illumination variation, fast motion, et al. To overcome these problems, several algorithms are proposed, they can be classified into two categories, the generative-model-based approaches [1–5] and the discriminative ones [6–12].

In a sense, visual tracking can be reduced to a search task and formulated as an optimization problem. Among them, swarm intelligence algorithm has

This work is supported by National Natural Science Foundation of China (61873246, 61503173, 61501407, 61603347) and Doctor fund project of Zhengzhou University of Light Industry (2016BSJJ002).

J.-H. Lai et al. (Eds.): PRCV 2018, LNCS 11259, pp. 284–294, 2018.
https://doi.org/10.1007/978-3-030-03341-5_24

attracted more and more attention and has been applied to tracking problems successfully. Minami et al. [13] introduced the genetic algorithm into 1-step-GA (Genetic Algorithm) evolution for expressing the deviation of the target in the images. Meanwhile, with the help of the PD-type controller, the fish obtained the better identification and tracking results. Zhang et al. [14] incorporated the temporal continuity information into the PSO (Particle Swarm Optimization) for forming a multilayer importance sampling in the framework of particle filter. In this case, the tracker got better performance especially when the object has an arbitrary motion or undergoes large appearance changes. Chen et al. [15] proposed a Euclid distance based HQPSO (hybrid quantum particle swarm optimization) method, which overcome the problem that the population diversity gets easily lost during the latter period of evolution in PSO for an improved Mean Shift tracker. Hao et al. [16] proposed a particle filtering algorithm based on ACO (ant colony optimization) to enhance the performance of particle filter with small sample set, effectively improved the efficiency of video object tracking system. Nguyen et al. [17] presented a modified BFO (bacterial foraging optimization) algorithm, and designed a visual tracking system based on the bacterial foraging optimization to handle the some challenges. Gao et al. [18] proposed FA-based (firefly algorithm based) tracker which can robustly track an arbitrary target in various challenging conditions. Ljouad et al. [19] proposed a modified version of the CS (Cuckoo Search) algorithm combined with the well-known Kalman Filter, and designed a visual tracking system based on the HKCS (Hybrid Kalman Cuckoo Search) algorithm, the tracker outperforms better than the PSO-based tracker, especially in terms of computation time. In addition, Gao et al. [20,21] presented bat algorithm and flower pollination algorithm to solve tracking problem.

Recently, Moth-flame optimization algorithm (MFO), a new swarm intelligent optimization algorithm, was proposed by researcher Seyedali Mirjalili [22], which inspired by the spiral convergence toward artificial lights of moths. Since being put forward, MFO algorithm has received more and more attentions because of its good robustness, fast convergence speed and global optimization. In this paper, visual tracking is viewed as the process of searching the best solution using the MFO method in the sequence images. A new visual tracking framework based on MFO is designed to obtain a better tracking performance by high exploration and exploitation. The spiral flight of moths and reduce the number of flames gradually is hired to enhance search capabilities and speed up convergence respectively. To demonstrate the tracking performance of the MFO-based tracker, the comparison of the proposed method and the three representative optimization based trackers was given, which were CS-based tracker, PSO-based tracker and SA-based (Simulated Annealing based) tracker respectively.

2 MFO Algorithm

Inspired by the navigation method of moths in nature called transverse orientation, Seyedali Mirjalili proposes Moth-flame optimization (MFO) algorithm. In this method, a moth flies through a relatively fixed angle to the moon, which is a very effective mechanism for travelling long distances in a straight path. However, when the light source (for example, flame) is close, moths fly spirally around it and finally converge toward it after just a few corrections. This mechanism is called Moth-Flame Optimization (MFO).

The key components in the MFO algorithm are moths and flames, which are considered to be a solution, however, they differ in the way they are handled and updated. The moth is the actual search body that moves in the search space, and the flame is the best solution that the moth has acquired so far.

2.1 Spiral Flight of Moth

In order to simulate the moths behavior of convergence in the mathematical model, a logarithmic spiral is defined in the MFO algorithm:

$$P\left(A_m, B_n\right) = F_m \bullet e^{\lambda t} \bullet \cos\left(2\pi t\right) + B_n \tag{1}$$

where F_m indicates the distance between the $m-th$ moth and the $n-th$ flame. λ is a constant that impacts the shape of the logarithmic spiral. And t is a random number in [-1,1], where -1 and 1 indicate the closest and farthest to the flame, respectively. F_m can be calculated as follows:

$$F_m = |B_n - A_m| \tag{2}$$

where A_m indicates the $m-th$ moth, B_n indicates the $n-th$ flame.

Equation (1) describes the spiral flying path of moths. As may be seen in this equation, the next position of a moth is defined with respect to a flame. The t parameter in the spiral equation determines how much the next position of the moth should be close to the flame. In order to further emphasize exploitation, it is assumed that t is a random number in [r,1]. Adaptive convergence constant r linearly decreases from -1 to -2 to accelerate convergence around the flames over the course of iteration.

2.2 The Number of Flames

Another concern here is that the position updating of moths with respect to flame positions in the search space may degrade the exploitation of the best promising solutions. To solve the problem, an adaptive mechanism is employed for the number of flames, as shown in Eq. (3):

$$flame\_no = round\left(N - k * \frac{N-1}{K}\right) \tag{3}$$

where k is the current number of iteration, N is the maximum number of flames, and K is the maximum number of iterations.

In MFO algorithm, the spiral equation is equipped with efficient local exploitation and global exploration capabilities. Global exploration occurs when the next position is outside the space between the moth and flame, local exploitation happens when the next position lies inside the space between the moth and flame. The gradual decrement in number of flames balances the search space for global exploration and local exploitation. It shows superior capabilities of challenging optimization problems with unknown search spaces.

3 MFO Algorithm-Based Tracking System

Suppose there is a ground-truth corresponding to the object (best flame) in the image (state space) being searched and a group of target candidates (moths and flames) are randomly generated in the image (state space). The aim of the MFO-based tracker is to find the best target candidate in all candidates using the MFO algorithm.

In the tracking system, the moths are actual search agents that searching the candidate image patches in the image, whereas flames can save the best positions of image patches that moths have searched so far. Meanwhile, the flames guide the next searching of moths. The best image patches are saved in the flames so they never get lost. Based on this, a MFO-based tracking architecture is designed.

Algorithm 1. MFO-based tracking system

```
Input: Image sequence
Initialization: Locate the target object $T$ in the first frame manually; Define objective function $E$; Set initial parameters: The initial number of moths and flames $N$; Maximum number of iterations $K$; Logarithmic spiral shape constant $b$.
Tracking:
for ($i$ from 2 to the last frame) do
    Generate the initial positions of moth candidates and flame candidates
    for ($k = 1 : K$) do
        Calculating the similarity value of moths candidates and flame candidates
        Replace an flame candidate with its corresponding moth candidate it if becomes more similarity
        Select the best flame candidate.
        Update the number of flames candidates using Eq.(3)
        Update the position of moths candidates using Eq.(1)
    end for
    Display the frame indicating the best flame candidate, which is the target.
end for
```

In the Algorithm 1, the candidates similarity value is computed as:

$$\rho(X, Y) = \frac{Cov(X, Y)}{\sqrt{D(X)}\sqrt{D(Y)}} \tag{4}$$

where $D(\bullet)$ denotes the variance and $Cov(\bullet)$ denotes covariance. X and Y is the HOG feature of the target and candidate samples respectively. The objective function E is introduced as follows:

$$E = 2 + 2 * \rho(X, Y) \tag{5}$$

The similarity value affects how to update the search of moths and select the positions of flames.

After that, visual tracking is considered an optimization problem and the search space of the optimization problem is the image. The MFO method is used to find the best candidate the image as the tracking output according to the maximizing similarity value.

4 Experiments

The MFO-based tracker that was proposed in this paper should be tested in MATLAB R2014a. The experiments were managed on a PC with Intel Core i5 2.50 GHz and 16GB RAM. To prove the ability of MFO-based tracker in visual tracking, we compared MFO-based tracker with three representative optimization based tracker, including CS-based tracker, PSO-based tracker and SA-based tracker.

In MFO-based tracker, the parameters are set as follows: the number of moths/flames $N = 500$, maximum number of iterations $K = 50$ and the constant of logarithmic spiral shape $b = 2$. To make a fair comparison, the same feature (HOG feature), and each algorithm executed for 25000 objective function evaluations, which is adequate for relative the performances of the trackers. In MFO-based tracker, when the search space is too small or too large, the parameters can be adjusted to prevent overwork or underwork. We keep the parameters constant in this test.

There are 10 challenging sequences in our experiments. The source of the FACE1 is the dataset AVSS2007, and ZXJ is our own. Others are available on the website http://visualtracking.net. Note that we extract the frames 306–310 in BLURFACE sequences, which can represent the problem of frame dropping.

4.1 Qualitative Analysis

As shown in Fig. 1(a), there are the severe motion blur at frame #0150 and #0272, which reduces the discriminative information in feature vectors, it is difficult to predict their locations. At the beginning, all trackers perform well in sequences but SA-based tracker drifts when target objects undergo abrupt motion at frame #0311. In the BLURFACE sequence, the MFO-based tracker and CS-based tracker outperform the other trackers.

As shown in Fig. 1(b), it describes several deer run and jump in a river. The target undergoes large motion and some frames are blurred, for example frame #0036. Meanwhile, there are similar targets at frame #0052 that interfere with

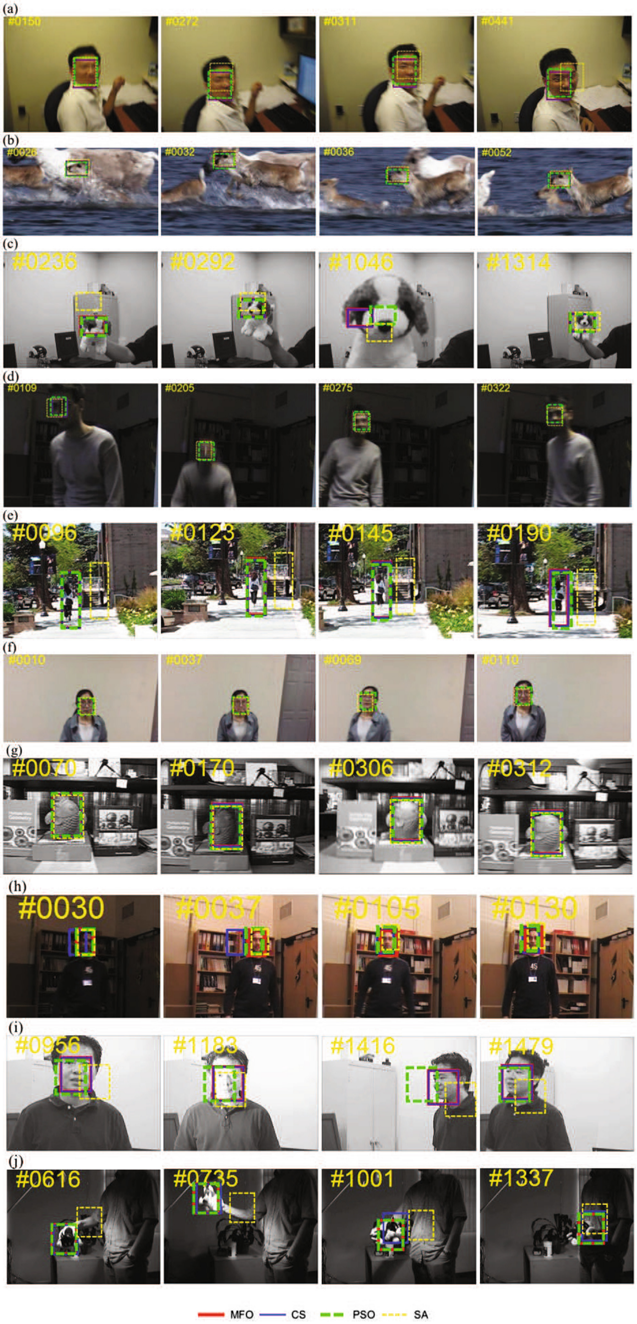

Fig. 1. A visualization of tracking results

tracking. MFO-based tracker generates accurate results even with heavy motion blur and water occlusion. Compared with these trackers, although all trackers keep track of the target object, the tracking accuracy of MFO-based tracker is better than the others.

As shown in Fig. 1(c), in the DOG1 sequence, a toy dog moves towards the camera. At frames #1046 and #1314 there are large scale variation. In addition, at frames #0236 and #0292 the target is rotated. All methods can merely track the target to the end. Only SA-based tracker failure in some frames. The PSO-based tracker has the best merit, the MFO-based tracker and CS-based tracker are the next simultaneously.

As shown in Fig. 1(d), at frame #0109, the target occur scale changes. Furthermore, at frames #0322 and #0275 the target becomes blurry. From Fig. 1(d), we observe that SA-based tracker took the worst to execute. The average execution merit of MFO-based tracker, CS-based tracker, and PSO-based tracker are all comparable.

As shown in Fig. 1(e), in the HUMAN7 sequence, the target undergoes fast motion at frames #0096 and #0123 with camera drastic camera shaking. It is rather tough to accurately locate targets position. In addition, at frame #0190 the brightness of the light changes when target through the shade. The SA-based tracker had lost the target before the frame #0096, MFO-based tracker, CS-based tracker and PSO-based tracker work well. However, MFO-based tracker and CS-based tracker have the superior results compared with PSO-based tracker.

As shown in Fig. 1(f), for ZXJ sequence, before the frame #0037, all tracker work well. However, when the target has an abrupt motion at frame #0069, SA-based tracker off its track to some extent. The MFO-based tracker and CS-based tracker obtain the best performance.

As shown in Fig. 1(g), for FISH sequence, the target occurs slight motion blur because camera shaking continuously. In addition, the illumination of the target at frame #0306 and #0312 is larger than the target at frame #0170. Due to the relative smooth motion, four trackers show good tracking results.

As shown in Fig. 1(h), for MAN sequence, the target experienced a large change in brightness. The tracker of CS fails at frame #0037. Although it catches the target at frame #0105 and #0130, it is failing for a complete video. MFO-based tracker stands out in all trackers.

As shown in Fig. 1(i), for MHYANG sequence, the target deformed and the light changed. Comparing frame #0956 with #1183, we can see the changes in the light of the target. By comparing frame #1416 and #1479, we can see the changes in the appearance of the target.

As shown in Fig. 1(j), for SYLVESTER sequence, as the toy continues to flip, the shape of the target changes dramatically. At the same time, the target has a large change in light. SA-based tracker has the worst performance.

4.2 Quantitative Analysis

Table 1 list image sequences comparison of MFO-based tracker to CS-based tracker, PSO-based tracker and SA-based tracker. The table mainly refers to average overlap rate.

Table 1. Average overlap rate

Sequence	MFO	CS	PSO	SA
BLURFACE	0.71	0.71	0.66	0.49
DEER	0.74	0.73	0.69	0.69
DOG1	0.46	0.46	0.47	0.32
FACE1	0.69	0.69	0.70	0.56
HUMAN7	0.47	0.47	0.46	0.16
ZXJ	0.85	0.85	0.82	0.67
FISH	0.84	0.84	0.82	0.78
MAN	0.83	0.37	0.70	0.67
MHYANG	0.75	0.75	0.60	0.47
SYLVESTER	0.63	0.65	0.63	0.29

Figure 2 shows the average success plot. The y-axis shows ratio of frames, where, amount of overlap is above threshold, to total frames. More the area under the curve then better is the tracker. Figure 3 shows the precision plot of the average precision values of all sequences. Here, also the y-axis shows the ratio of frames, where, distance of predicted and ground truth bounding box is below the threshold, to total frames. In precision plots if the slope is higher, then the tracker is better, this is because more sequences have distance of centers lower than the threshold. It is clearly seen, in Table 1, Figs. 2 and 3 that MFO-based tracker and CS-based tracker performs much better than 2 other trackers. It's worth noting that although the MFO-based tracker has better performance in the DOG1 and HUMAN7 sequences, we can see from the tracking field that the MFO-based tracker does not adapt to the video sequence with the scale change.

4.3 Average Time Costs

In order to analyze the time complexity, the average time costs of the trackers in the tracking process are recorded and the comparative results are shown in Table 2.

It can be seen from Table 2 that the average time cost of the MFO-based tracker is a little more than that of the SA-based tracker, but it is much better than the other three trackers. The time cost of the MFO-based tracker is especially advantageous compared with the CS-based tracker, although their tracking accuracy is often similar.

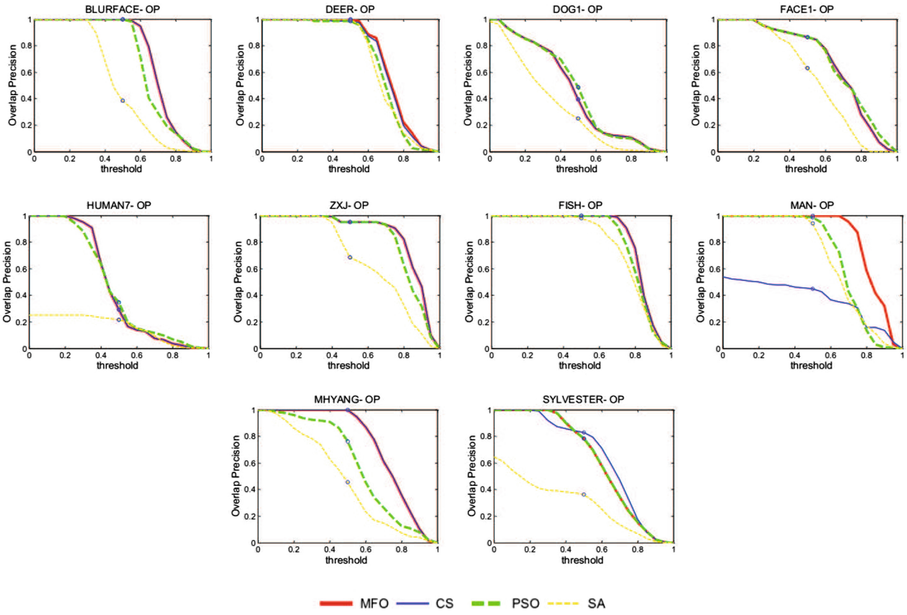

Fig. 2. Precision plots of OPE

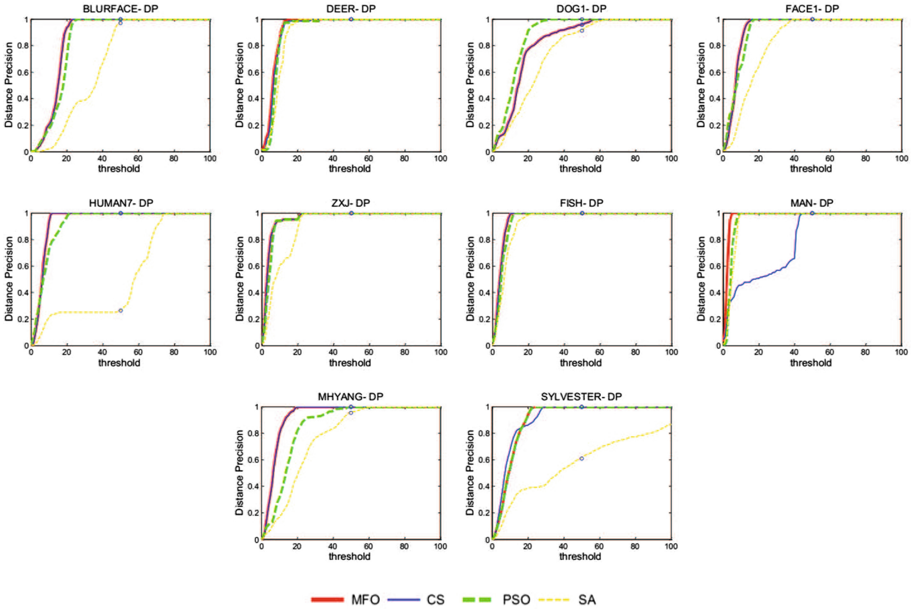

Fig. 3. Success plots of OPE

Table 2. Average time costs of the trackers

Sequence	MFO	CS	PSO	SA
BLURFACE	33.5	34.5	32.1	21.9
DEER	21	40.3	23.3	15.6
DOG1	9.7	12.6	9.9	8.9
FACE1	19.5	24.5	22	15.3
HUMAN7	15.3	20.6	17.4	12.7
ZXJ	12.8	30.1	15.5	10.7
FISH	12.2	15	9.1	9.7
MAN	4.7	14.1	6.7	4.6
MHYANG	9.8	13.2	12.1	8.3
SYLVESTER	8.7	11.7	9.9	7.4

5 Conclusion

In this paper, visual tracking is considered to be a process of searching for target using MFO in sequential images. The moths are the actual exploit body that move in the search space to determine candidate sample set, and the flames are the best solutions that the moth has acquired so far. Meanwhile, an appearance model based on HOG feature is constructed to measure the similarity between the target and candidates. Finally, according to the best fitness value, the tracking states are obtained. To the best of our knowledge, this is the first time that MFO has been adapted for use in a visual tracking system.

A tracker can be evaluated in many ways, and in this article, we mainly analyze MFO-based tracker from the perspective of optimization. The accuracy of the MFO-based tracker compared with three optimization based tracker including the CS, PSO and SA. Experimental results show that the accuracy of MFO-based tracker outperformed the PSO-based tracker and SA-based tracker, and the time cost is much better than CS-based tracker and PSO-based tracker. However, optimization algorithm is used to solve the problem of tracking that is a very time-consuming especially the number of image sequence is large. Future work is expected to combine MFO-based tracking system with the traditional tracking algorithm for high efficiency and high accuracy of target tracking.

References

1. Wang, D., Lu, H., Xiao, Z., Yang, M.: Inverse sparse tracker with a locally weighted distance metric. IEEE Trans. Image Process. **24**(9), 2646–2657 (2015)
2. Meshgi, K., Maeda, S., Oba, S., et al.: An occlusion-aware particle filter tracker to handle complex and persistent occlusions. Comput. Vis. Image Und. **150**, 81–94 (2016)

3. Bibi, A., Zhang, T., Ghanem, B.: 3D part-based sparse tracker with automatic synchronization and registration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, pp. 1439–1448 (2016)
4. Zhang, H., Wang, Y., Luo, L., et al.: SIFT flow for abrupt motion tracking via adaptive samples selection with sparse representation. Neurocomputing **249**(2), 253–265 (2017)
5. Zhou, T., Bhaskar, H., Liu, F., et al.: Graph regularized and locality-constrained coding for robust visual tracking. IEEE Trans. Circ. Syst. Video Technol. **27**(10), 2153–2164 (2017)
6. Zhou, T., Liu, F., Bhaskar, H., et al.: Robust visual tracking via online discriminative and low-rank dictionary learning. IEEE Trans. Cybern. **PP**, 1–13 (2017)
7. Zhao, H., Wang, X.: Robust visual tracking via discriminative appearance model based on sparse coding. Multimedia Syst. **23**(1), 75–84 (2017)
8. Sun, C., Li, F., Lu, H., et al.: Visual tracking via joint discriminative appearance learning. IEEE Trans. Circ. Syst. Video Technol. (2016)
9. Liu, F., Zhou, T., Fu, K., et al.: Robust visual tracking via constrained correlation filter coding. Patt. Recog. Lett. **84**, 163–169 (2016)
10. Nam, H., Han B.: Learning multi-domain convolutional neural networks for visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, pp. 4293–4302 (2016)
11. Du, D., Zhang, L., Lu, H., et al.: Discriminative hash tracking with group sparsity. IEEE Trans. Cybern. **46**(8), 1914–1925 (2016)
12. Zhang, H., Zhang, J., Wu, Q., et al.: Extended kernel correlation filter for abrupt motion tracking. KSII Trans. Int. Inf. Syst. **11**(9), 4438–4460 (2017)
13. Minami, M., Agbanhan, J., Asakura, T.: Manipulator visual servoing and tracking of fish using a genetic algorithm. Indus. Robot Int. J. **26**(4), 278–289 (1999)
14. Zhang, X., Hu, W., Maybank, S., et al.: Sequential particle swarm optimization for visual tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008, pp. 1–8. IEEE (2008)
15. Chen, J., Zhen, Y., Yang, D., et al.: Fast moving object tracking algorithm based on hybrid quantum PSO. WSEAA Trans. Comput. **12**, 375–383 (2013)
16. Hao, Z., Zhang, X., Yu, P., et al.: Video object tracing based on particle filter with ant colony optimization. In: 2010 2nd International Conference on Advanced Computer Control (ICACC), vol. 3, pp. 232–236. IEEE (2010)
17. Nguyen, H., Bhanu, B.: Real-time pedestrian tracking with bacterial foraging optimization. In: 2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance (AVSS), pp. 37–42. IEEE (2012)
18. Gao, M., He, X., Luo, D., et al.: Object tracking using firefly algorithm. IET Comput. Vis. **7**(4), 227–237 (2013)
19. Ljouad, T., Amine, A., Rziza, M.: A hybrid mobile object tracker based on the modified Cuckoo Search algorithm and the Kalman Filter. Patt. Recogn. **47**, 3597–3613 (2014)
20. Gao, M., Zang, Y., Shen, J., et al.: Visual tracking based on flower pollination algorithm. In: Proceedings of the 35th Chinese Control Conference, Chengdu, pp. 3866–3868 (2016)
21. Gao, M., Shen, J., Yin, L., et al.: A novel visual tracking method using bat algorithm. Neurocomputing **177**, 612–619 (2016)
22. Mirjalili, S.: Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl.-Based Syst. **89**, 228–249 (2015)