

Published in IET Computer Vision  
 Received on 23rd September 2012  
 Revised on 19th January 2013  
 Accepted on 1st February 2013  
 doi: 10.1049/iet-cvi.2012.0207



ISSN 1751-9632

# Object tracking using firefly algorithm

Ming-Liang Gao, Xiao-Hai He, Dai-Sheng Luo, Jun Jiang, Qi-Zhi Teng

College of Electronics and Information Engineering, Sichuan University, Chengdu 610064, People's Republic of China  
 E-mail: hxx@scu.edu.cn

**Abstract:** Firefly algorithm (FA) is a new meta-heuristic optimisation algorithm that mimics the social behaviour of fireflies flying in the tropical and temperate summer sky. In this study, a novel application of FA is presented as it is applied to solve tracking problem. A general optimisation-based tracking architecture is proposed and the parameters' sensitivity and adjustment of the FA in tracking system are studied. Experimental results show that the FA-based tracker can robustly track an arbitrary target in various challenging conditions. The authors compare the speed and accuracy of the FA with three typical tracking algorithms including the particle filter, meanshift and particle swarm optimisation. Comparative results show that the FA-based tracker outperforms the other three trackers.

## 1 Introduction

Object tracking has drawn a great deal of attention in recent years. The reason is that object tracking has found its way into many real-world applications, for example, surveillance [1, 2], vision-based control [3, 4] and robotics [5, 6]. However, object tracking in video sequences is still a challenging task because of the large amount of data used and the common requirement for real-time computation. Moreover, most of the models encountered in visual tracking are non-linear, non-Gaussian, multi-modal or any combination of these.

To solve the problems encountered in the tracking process, researchers have performed visual tracking using a variety of methods and algorithms which can be generally divided into two categories: probabilistic methods and deterministic methods [7]. Probabilistic methods view the tracking algorithm as a state solving problem under the Bayesian framework, modelling uncertainty and propagating the conditional densities through the tracking process. Representative methods are Kalman filter [8] and particle filter (PF) [9]. Deterministic methods localise the tracked object in each frame by iteratively searching for a region which maximises the similarity measure between this region and the target window. These methods are computationally efficient but may converge to local maximum and are sensitive to background distractors, clusters, occlusions and quick moving objects. Meanshift is the representative one [10].

Essentially speaking, meanshift is a gradient-based optimisation algorithm and it has been used successfully in visual tracking. This inspired more and more researchers to investigate other optimisation algorithms using different strategies to solve tracking problems. Typical methods are particle swarm optimisation (PSO) algorithm [11–13] and genetic algorithm (GA) [14]. Meanwhile, in recent years, another novel optimisation algorithm named harmony search has been proposed and analysed in visual tracking [15, 16].

These contributions, as related to the scope of our research, are discussed below. The PSO algorithm was successfully applied to visual tracking by Zhang *et al.* [11]. In their work, the parameters that control the movement of the particles in the swarm were updated dynamically depending on the fitness values of the particles. Experimental results showed that the PSO-based tracker was more robust and effective than the state-of-the-art PF and unscented PF-based tracking systems especially in fast and erratic motion situation. Subsequently, Zhang *et al.* [12] proposed an annealed PSO-based particle filter algorithm for articulated three-dimensional (3D) human body tracking. Experiments with multi-camera walking sequences showed that their tracker was robust to noise and body self-occlusion and could alleviate the problem of inconsistency between the image likelihood and the true model. Recently, John *et al.* [13] proposed a hierarchical PSO (HPSO) algorithm to solve the markerless full-body articulated human motion tracking from multi-view video sequences. Their results showed that the HPSO performed well in sequences with sudden and fast motion and the accuracy and consistency were better than PF and annealed PF.

A visual system using GA was proposed by Minami *et al.* [14]. In their work, a fish was tracking and GA was used to optimise a function that minimises the difference between a previously defined fish template and the image captured by the camera. Experiments indicated that their system performed well in recognising and robustly tracking a fish target in real-time using limited amounts of computational resources.

Recently, Fourie *et al.* [15] designed a visual tracking system based on the improved harmony search (IHS) algorithm. In their work, the target was modelled as a colour histogram and the best estimated target location was obtained by using the Bhattacharyya coefficient as a fitness metric. Experimental results showed that IHS was able to track poorly modelled targets in real time. Gao *et al.* [16] carried on a further study on harmony search (HS)-based

tracking system. In their work, the performance of four prominent improved variations of HS, namely IHS, global-best HS, self-adaptive HS, differential HS were tested and analysed comparatively on multiple challenging video sequences.

Overall, with the rapid development of modern optimisation algorithms, more and more researchers started to investigate optimisation algorithms to perform visual tracking. The advantage of these methods is that no assumptions are made about the shape of the distribution or the noise in the system. Therefore it enables these methods a potential method for accurate solutions even in challenging ambiguous environments.

Lately, a new biologically inspired algorithm, namely firefly algorithm (FA), was proposed by Yang [17]. FA algorithm is based on the idealised behaviour of the flashing characteristics of fireflies. Preliminary studies indicated that FA was superior over GA and PSO [17]. Since the emergence of this algorithm, it has been successfully applied to various optimisation problems, for example, economic dispatch, structural optimisation, image compression etc. [18–22]. In this paper, FA is applied to solve the object tracking problem. To demonstrate the tracking ability of FA-based tracker, the tracking performances of FA, PF, meanshift and PSO are studied comparatively.

The rest of the paper is organised as follows: In Section 2, the relationship between the optimisation and tracking is discussed and a general optimisation-based tracking architecture is designed. In Section 3, the basic concepts and procedure of FA are discussed. In Section 4, the parameters' sensitivity and adjustment of FA in the tracking system are analysed. In Section 5, the tracking performance of FA is compared with that of PF, meanshift and PSO and their results are analysed. In Section 6, we conclude the paper and identify future research directions that stem from this study.

## 2 Optimisation and tracking

### 2.1 Relationship between optimisation and tracking

Optimisation is the process of selecting the best element from some sets of available alternatives under certain constraints [23, 24]. Optimisation techniques are used on a daily basis for industrial planning, resource allocation, econometrics problems, scheduling, decision making, engineering and computer science applications [23]. Research in the optimisation field is very active and new optimisation methods are being developed regularly [25, 26].

Mathematically speaking, all optimisation problems with explicit objectives can be expressed as non-linearly constrained optimisation problems in the following generic form

$$\text{maximise/minimise } f(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^n \quad (1)$$

where  $f$  is considered as an objective function or cost function.  $\mathbb{R}^n$  is the search space. The vector  $\mathbf{x}$  is often called a decision vector which varies in a  $n$ -dimension space  $\mathbb{R}^n$ .

Essentially speaking, tracking object in video sequences, or the problem of locating the target in each frame can be interpreted as an optimisation problem. The observation distance between the target and candidate forms the similarity function (fitness function). Locating the target can be interpreted as minimising or maximising the similarity function in the candidate solution. In this regard, visual

tracking, as an optimisation problem, can be achieved using optimisation techniques.

### 2.2 Optimisation-based tracking system

To compare the tracking performance of optimization-based trackers, we had designed a general optimisation-based tracking architecture to which other optimisers could also be applied [16]. The architecture is illustrated in Fig. 1.

As described in Fig. 1, the process starts with a target chosen by the user marked in a rectangle or ellipse. Then, the state vector is initialised. The state vector in our work is defined as  $\mathbf{x} = [x, y, s]$ , where  $x, y$  is the target's location in pixel coordinates and  $s$  denotes the scale parameter that controls the size of the object. Then, once the target is chosen in the first frame the state vector is initialised as  $\mathbf{x}_0 = [x_0, y_0, 1.0]$ , where  $x_0, y_0$  is the target's initial position and  $s = 1.0$  indicates there is no scale change in the initial frame.

Once a target is selected and the state vector is initialised, new candidates' state vectors are generated by a dynamic model. Considering that the object moves very little between frames, the random walk model is applied. Note: we can also initialise the vector using a motion model that assumes constant velocity or constant acceleration and in this case the state vector is a 5D problem. In our work, we choose a simple motion model to compare the novel method with the PF, meanshift and PSO.

An observation model is established to describe the correlation between the appearance and the state of the object. In our work, the spatial color histogram is adopted [10]. A similarity (fitness) function is formed to measure the observation distance between the target and candidate. Typically, the Bhattacharyya coefficient is used to measure the similarity between two histograms [10, 15]. It is defined as

$$B(h_1, h_2) = \sum_{i=1}^N \sqrt{h_1(i)h_2(i)} \quad (2)$$

where  $N$  is the number of bins in the histograms and  $h_1$  and  $h_2$  are the histograms being compared. It is noted that  $B(h_1, h_2)$  is large when the histograms are similar while small when they are very different.

The dash box in the architecture denotes the optimisation process. This is the core part in the optimisation-based tracking algorithm. In this part, an optimiser is adopted to

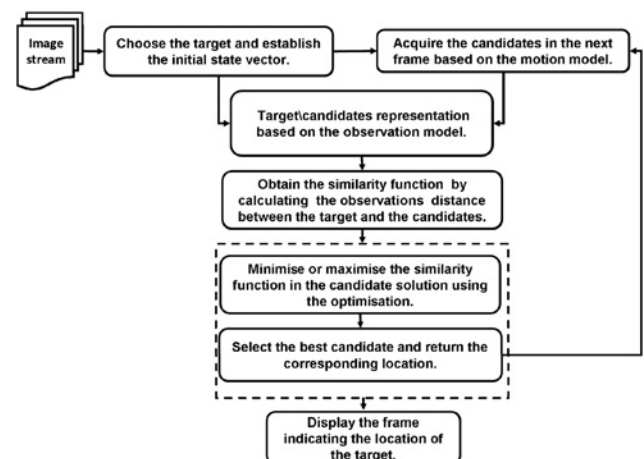


Fig. 1 Architecture of optimisation-based tracking algorithm

select the candidate solution. This process can be carried out by minimising or maximising the similarity function. Every time the optimiser is queried for the target location, the frame is displayed to indicate the location of the target. The whole loop continues until no more frame is available.

### 3 Firefly algorithm

The FA was developed by Yang [17]. It mimics the social behaviour of fireflies flying in the tropical and temperate summer sky. Fireflies communicate, search for prey and find mates using bioluminescence with varied flashing patterns. FA is based on the following idealised behaviour of the flashing characteristics of fireflies:

1. All fireflies are unisex so that one firefly will be attracted to other fireflies regardless of their sex.
2. Attractiveness is proportional to their brightness, thus for any two flashing fireflies, the less brighter one will move towards the brighter one. The attractiveness is proportional to the brightness and they both decrease as their distance increases. If there is no brighter one than a particular firefly, it will move randomly.
3. The brightness of a firefly is affected or determined by the landscape of the objective function.

The basic steps of the FA are summarised by the pseudo code shown in Fig. 2 [17, 19].

The brightness  $I$  of a firefly at a particular location  $\mathbf{x}$  is determined by the objective function  $I(\mathbf{x}) \propto f(\mathbf{x})$ . The attractiveness  $\beta$  is relative and it should be seen in the eyes of the beholder or judged by the other fireflies. Therefore it will vary with the distance  $r_{ij}$  between firefly  $i$  and firefly  $j$  by

$$\beta(r_{ij}) = \beta_0 e^{-\gamma r_{ij}^2} \quad (3)$$

As it is often faster to calculate  $1/(1+r^2)$  than an exponential function, the above function can conveniently be replaced by

$$\beta(r_{ij}) = \beta_0 / (1 + \gamma r_{ij}^2) \quad (4)$$

where  $\beta_0$  is the attractiveness at  $r=0$  and in our work  $\beta_0=1$ . The parameter  $\gamma$  characterises the variation of the attractiveness, and its value is crucially important in determining the speed of the convergence and how the FA algorithm behaves.

The distance between any two fireflies  $i$  and  $j$  at  $\mathbf{x}_i$  and  $\mathbf{x}_j$  can be the Cartesian distance

$$r_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2} \quad (5)$$

where  $x_{i,k}$  is the  $k$ th component of the spatial coordinate  $\mathbf{x}_i$  of the  $i$ th firefly.

The movement of a firefly  $i$  which is attracted to another more attractive firefly  $j$  is determined by

$$\mathbf{x}_i = \mathbf{x}_i + \beta_0 e^{-\gamma r_{ij}^2} (\mathbf{x}_j - \mathbf{x}_i) + \alpha S_k \left( \text{rand} - \frac{1}{2} \right) \quad (6)$$

where the second term is because of the attraction while the third term is randomisation with  $\alpha \in [0, 1]$ . Without alpha term, the system will start evolve from initial configurations, but in a deterministic manner. On the contrary, with the alpha term, the agents in FA explore the search space locally, aided by randomisation which increases the diversity of the solutions. Thus, there is a fine balance between local intensive exploitation and global exploration.  $S_k (k=1, 2, \dots, d)$  are the scaling parameters in the  $d$  dimensions which are determined by the actual scopes

---

#### Firefly Algorithm

---

```

Objective function  $f(\mathbf{x})$      $\mathbf{x} = (x_1, x_2, \dots, x_d)^T$ 
Generate initial population of fireflies  $\mathbf{x}_i (i = 1, 2, \dots, n)$ 
Light intensity  $I_i$  at  $\mathbf{x}_i$  is determined by  $f(\mathbf{x}_i)$ 
Define light absorption coefficient  $\gamma$ 
while( $t < \text{MaxGeneration}$ )
  for  $i = 1:n$  all  $n$  fireflies
    for  $j = 1:i$  all  $n$  fireflies
      if ( $I_i < I_j$ )
        Move firefly  $i$  towards  $j$  in  $d$ -dimension
        Attractiveness varies with distance  $r$  via  $\exp[-\gamma r]$ 
        Evaluate new solutions and update light intensity
      end if
    end for  $j$ 
  end for  $i$ 
  Rank the fireflies and find the current best
End while
Postprocess results and visualization

```

---

Fig. 2 Pseudo code of FA



of the problem of interest [17]. rand is a random number generator uniformly distributed in [0, 1].

In this paper, the scaling factors (velocity noise variance and the scale noise variance) are empirical and they should be chosen based on the size of the video frame and the expected movement of the target [27].

For example, consider surveillance footage from a security camera. Let the video capture frames of  $500 \times 500$  pixels and the primary targets for tracking be people walking by the camera. If the frame rates 30 frames per second and the area in the frame represents 5 m of a corridor, we can calculate the number of pixels that a target is expected to move based on the average walking speed of people. The average walking speed of adults is considered as 80 m/min. If each video frame represents 1/30th of a second we therefore expect a human target to move 44.4 mm in each frame. If 500 pixels represent 5 m of corridor each pixel of the frame then represents 10 mm. Therefore even if a target was moving at twice the average speed we would not expect it to move more than 10 pixels (8.8 pixels) per frame. For this application a reasonable choice for the velocity noise variance is therefore 20 (Note: this is because there is a coefficient (rand-0.5) as shown in (6)).

The scale noise variance is a little more difficult to calculate and it is usually determined empirically. In our experiments, a reasonable assumption was that a target would never grow by more than 10% between frames so the scale noise variance was set as 0.2.

Therefore, in our work, the scales of the vector  $x$ ,  $y$  and  $s$  are chosen as 20, 20 and 0.2, respectively.

#### 4 Parameters' sensitivity and adjustment

It is worth mentioning that parameter tuning often seems a self-contradicting problem in optimisation algorithms. The speed and accuracy should be considered simultaneously during the parameter tuning. In FA, we used various population sizes from 10 to 50 and found that it's sufficient to use 15–20 population sizes for most tracking problems. Therefore we have used the fixed population size of  $n = 20$  in all our experiments. Like in [15], the optimisation process was terminated by using three termination conditions as follows:

1. The fitness of the worst solution ( $f_{\text{worst}}$ ) is good enough.
2. The best solution ( $f_{\text{best}}$ ) is good enough and the Euclidean distance between the best and the lowest solution ( $d$ ) is below a certain threshold.
3. Maximum number of iterations (MaxGeneration) reached the algorithm terminates. MaxGeneration is set as 500 in our work.

It is worth mentioning that the values used in the first two criteria are highly dependent on the objective function and in this case is the Bhattacharyya coefficient as mentioned in [10]. Before determining these parameters, we have done a lot of researches on the Bhattacharyya coefficient in visual tracking. Experiments showed a general trend that an area with Bhattacharyya coefficient being higher than 0.6 can mainly represent the target. Therefore, to be on the safe side, in the first termination condition, the value of  $f_{\text{worst}}$  is set as 0.6 to guarantee that all the potential solutions are near the true target. To guarantee all candidates close together with best candidate near the target, we set  $f_{\text{best}}$  as 0.8 and the distance  $d$  as 5.

Subsequently, we focus on the absorption coefficient  $\gamma$  and the randomisation parameter  $\alpha$  and find the optimal value of these parameters that will give us the most accurate estimate in the least amount of generations. We are also interested in the sensitivity to change of each of these parameters. This will give us an indication of the robustness of the algorithm and an indication of how much time has to be spent in fine-tuning the parameters.

We tested the algorithm's performance using a series of different parameter values on a challenging tracking problem involving a panda moving erratically along an area with various challenges. The video is recorded using a low-cost web cam and the quality of images is poor. Moreover, the panda is often occluded by the trees and the appearances of the panda change frequently. All those challenging factors will provide ample local distractors that can cause the tracker to lose its target. A video frame from this example is shown in Fig. 3.

We started our analysis with the parameter  $\gamma$  and divided its ranging (0, 1) into 20 equal parts, with each a spacing of 0.05. During the testing process, the tracking accuracy was considered in the first place and the speed (number of iterations) was considered in the next place. From the experiment results, we noted that when  $\gamma = 0.05$  and  $\gamma = 0.1$ , the tracker performed equally well, and there was no amount of optimisation saved the tracker from losing the target at certain points in the sequence. In contrast, when  $\gamma > 0.1$ , the performance declined evidently. Generally, from the 575th frame, the target was lost and could not be re-acquired in the following frames. One example illustrating this statement is shown in Fig. 4.

In order to determine the final optimal value  $\gamma$ , we further divided its ranging (0, 0.1) into 20 equal parts (with each a spacing of 0.005) and concentrated on the speed. Fig. 5 shows a graph comparing the speed of 20 different implementations corresponding to 20 different values of the  $\gamma$ . The values in this graph were gathered by running the algorithm once for each frame in the sequence and calculating the average over all frames.

It can be seen from Fig. 5 that the best choice of  $\gamma$  for this example is  $\gamma = 0.06$ .

Next, the randomisation parameter  $\alpha$  was investigated and its value was varied from 0.1 to 1, with each a spacing of 0.05. Likewise, our first concern is the tracking accuracy, and then



**Fig. 3** Challenging tracking problem used for parameter sensitivity and adjustment



Fig. 4 Example showing the lost target when  $\gamma > 0.1$

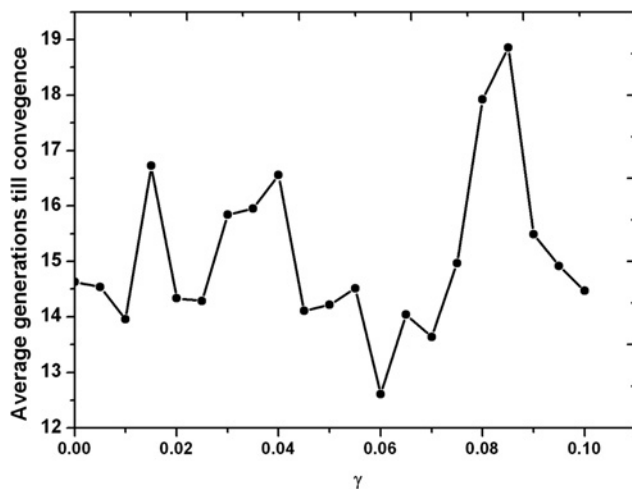


Fig. 5 Performance comparison using different values of  $\gamma$

the speed. Experiment results showed that the tracker performed well when  $0 \leq \alpha \leq 0.2$ . Then, we further divided it's ranging  $[0, 0.2]$  into 20 equal parts (with each a spacing of 0.01) and concentrated on the speed. Fig. 6 shows the performance comparison using different values of  $\alpha$ .

It can be seen from Fig. 6 that when  $0.05 \leq \alpha \leq 0.2$ , the convergence rate is equally fast. In our work,  $\alpha$  is set as

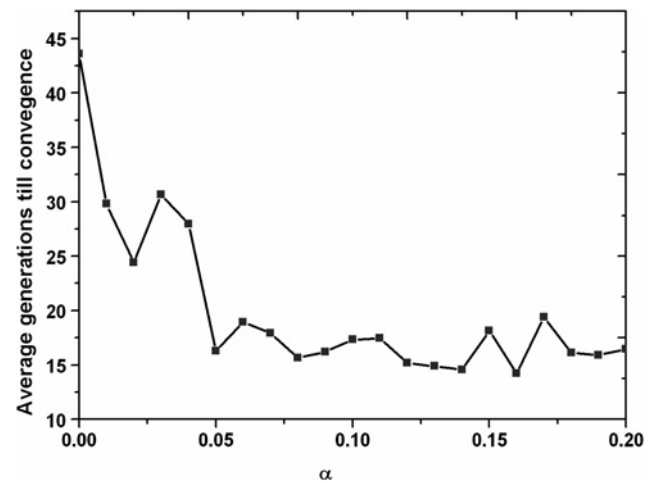


Fig. 6 Performance comparison using different values of  $\alpha$

0.16. So far, the main parameters of FA are determined and all these parameters remain fixed in the following experiments.

## 5 Experiments and discussions

To demonstrate the ability of FA in visual tracking, we compared our method with three representative tracking methods including PF with 300 particles (PF\_300) and 500 particles (PF\_500) [9] (probabilistic method), meanshift [10] (deterministic method) and PSO [11] (optimization-based method).

To carry on the comparison, we collected four challenging videos recorded indoors and outdoors. These video sequences contain various objects in challenging conditions including appearance changes, rapid and irregular motions, large scale changes, similar object interferences, poor image qualities and partial or full occlusions. A few snapshots from the video clips are shown in Fig. 7 [(a) is download from: <http://www.vision.ucsd.edu/project/tracking-online-multiple-instance-learning>, (b) and (c) are provided by Jaco Fourie and (d) comes from CAVIAR: <http://www.groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1>].

To make the comparison fair, the same target model and motion model were used in the implementations. For the FA, we used the parameter values found by the sensitivity analysis of Section 4.



Fig. 7 Four challenging tracking examples

- a Panda
- b Person outside
- c Basketball
- d Kitbag



In the PSO algorithm, the particle updates its velocity and state using the following equations in the  $n$ th iteration [11]

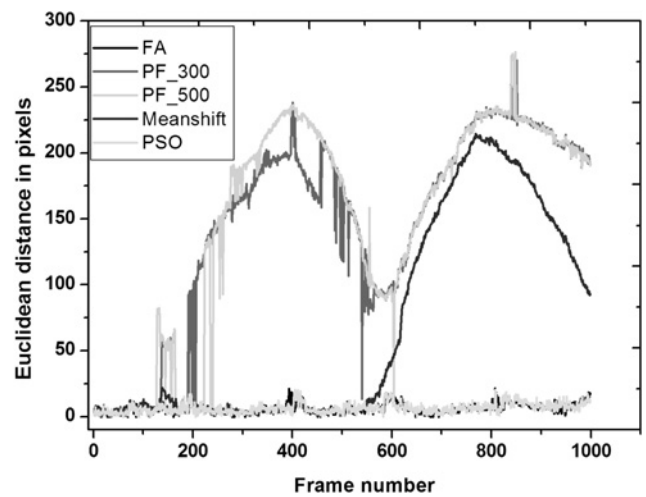
$$\begin{aligned} v^{i,n+1} &= \chi(v^{i,n} + \varphi_1 \mu_1(p^i - x^{i,n}) + \varphi_2 \mu_2(g - x^{i,n})) \\ x^{i,n+1} &= x^{i,n} + v^{i,n+1} \end{aligned} \quad (7)$$

where  $\varphi_1, \varphi_2$  are the acceleration constants,  $u_1, u_2 \in (0, 1)$  are uniformly distributed random numbers and  $\chi$  is a constriction factor which confines the velocity within a reasonable range:  $\|v^{i,n}\| \leq v^{\max}$ .

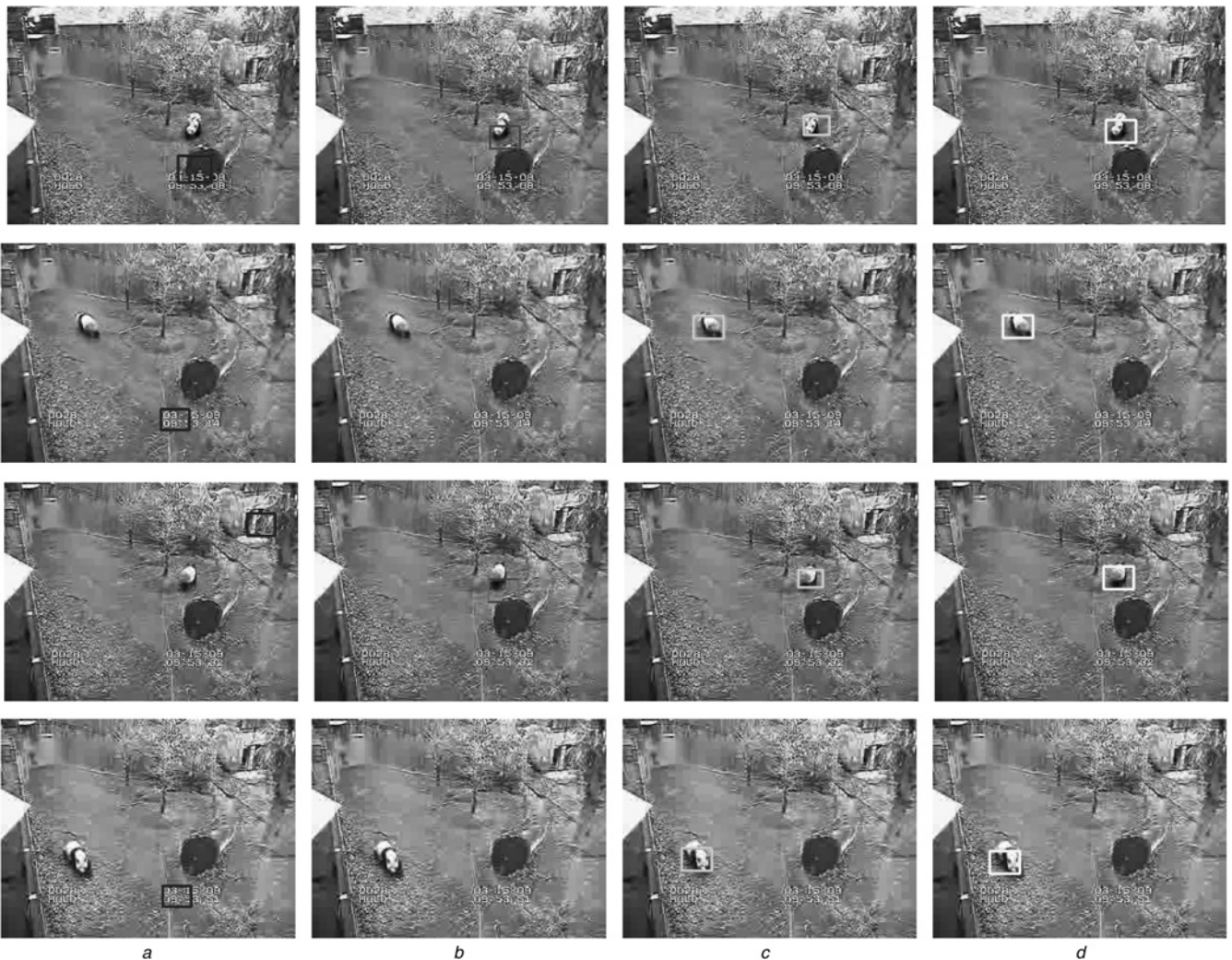
To make the comparison fair, in PSO algorithm, we used the same population size ( $N=20$ ), the same number of max iterations (MaxGeneration=500), and the same termination conditions as those used in FA. The acceleration constants  $\varphi_1, \varphi_2$  are set adaptively as follows [11]

$$\begin{aligned} \varphi_1 &= 2f(p^i)/(f(p^i) + f(g)) \\ \varphi_2 &= 2f(g)/(f(p^i) + f(g)) \end{aligned} \quad (8)$$

where  $f(p^i)$  and  $f(g)$  are the fitness values of the individual best and global best, respectively.



**Fig. 8** Tracking accuracy comparisons of different trackers in 'Panda' sequence



**Fig. 9** Examples illustrating the differences of accuracy in 'Panda' sequence

- a PF
- b Meanshift
- c PSO
- d FA

The constriction factor  $\chi$  is set to

$$\chi = \begin{cases} \|v_t^{\max}\| / \|v_t^{i, n+1}\|, & \text{if } \|v_t^{i, n+1}\| > \|v_t^{\max}\| \\ 1, & \text{else} \end{cases} \quad (9)$$

where

$$v_t^{\max} = 1.2 * (g_{t-1} - g_{t-2})$$

Our first example, named ‘Panda’ sequence, is the same one used for the parameters’ sensitivity analysis in the previous section. It includes the tracking challenges associated with appearance changes, rapid and irregular motions, similar object interferences, poor image qualities and partial or full occlusions.

In order to test the trackers’ accuracy, the video was manually labelled by identifying the centre of the tracked object in each frame visually. Then, the Euclidean distance between the true centre and the centre estimated by the trackers was calculated and used as an accuracy metric. Comparative results are shown in Fig. 8.

It can be seen from Fig. 8 that the four trackers perform equally well in the beginning (the first 135 frames). However, when the target suffered from rapidly appearance changes (from the 136th frame), the PF (both with 300 and 500 particles) loses the target and cannot recover the target in the following frames. This is likely due to the fact that the local distractors pull the particle clouds away from the target. When the target suffered from serious appearance changes and similar object interferences (570th frame), the meanshift-based tracker begins to lose the target and never obtains the opportunity to recover it properly in the following frames. Unlike the PF and meanshift, the two optimization-based trackers, PSO and FA, can robustly track the target. Examples illustrating the differences of accuracy are shown in Fig. 9.

As FA is a heuristic optimisation technique, it is interesting to carry an experiment of robustness analysis. We run these four algorithms on this video sequence for 20 times and obtained the results of stability analysis. We calculated the standard deviation (Std.Dev) of each frame, obtained a sequence of Std.Dev values. Generally, parameter Std.Dev denotes the stability of a tracker objectively. Comparative results are shown in Fig. 10.

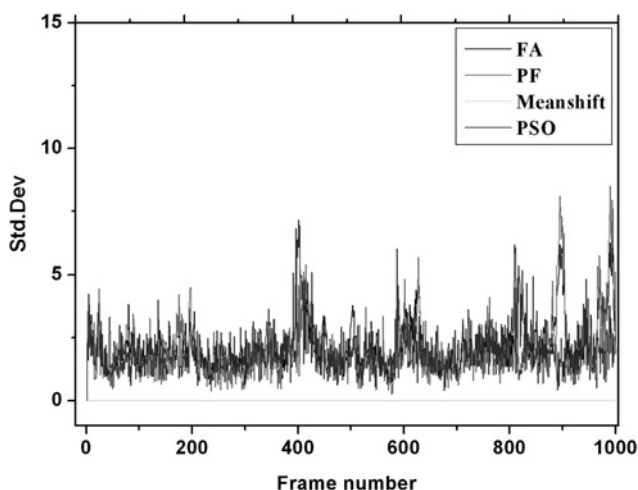


Fig. 10 Tracking stability comparisons of different trackers

Table 1 Average time cost for ‘Panda’ sequence

Tracking methods	Time cost, ms
PF_300	8.46777
PF_500	17.6592
meanshift	7.37305
PSO	80.1564
FA	6.10811

It can be seen from Fig. 10 that meanshift is the most stable tracker among the four trackers. Although FA and PSO are heuristic optimisation techniques, the stability of two trackers are as good as particle PF (500 particles in this experiment). It is worth mentioning that the robustness analysis of a tracker is built on the basis of its tracking accuracy. It is not of much value in analysis the stability of a tracker with poor tracking accuracy.

In order to analyse the time complexity, the average time costs of the four trackers are calculated and the comparative results are shown in Table 1.

As depicted in Table 1, the FA-based tracker is faster than the other four trackers on average. Especially, we can see that the speed of FA-based tracker is by far faster than the PSO-based tracker.

The second example, named ‘Person outside’ sequence, was also recorded using a low-cost web cam. In this video, a human target moves through an area with strong light and tree shadows and occlusions. Besides, the quality of the video is very low and the movement of the target is erratic. All these challenging conditions can easily cause the tracker to lose its target. Likewise, the Euclidean distance between the true centre and the centre estimated by the trackers were calculated and comparative results are shown in Fig. 11.

Fig. 11 shows a similar situation to the one in the first example. The four trackers perform equally well in the first 150 frames where the target is not suffered from occlusion. However, when suffered from serious occlusion, the PF-based tracker and meanshift-based tracker lose the target. In contrast, the PSO and FA-based tracker can still robustly track it. Examples that illustrate the differences in accuracy are shown in Fig. 12.

The average time of the four trackers in this example are recorded and shown in Table 2.

It can be seen from Table 2 that, the average time cost of FA-based tracker is a little more than meanshift but it is

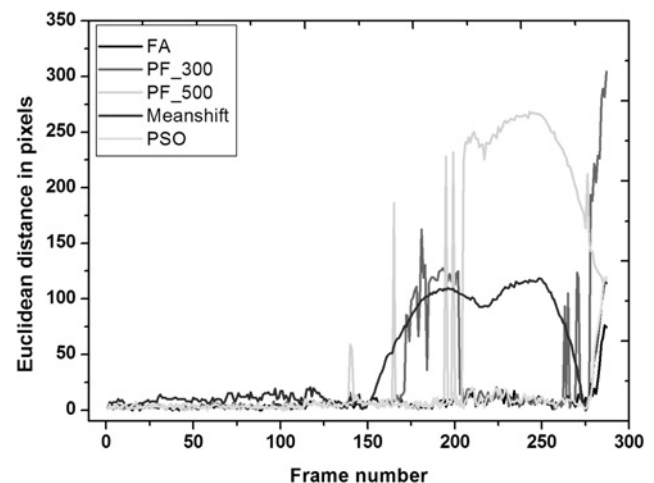


Fig. 11 Tracking accuracy comparisons of different trackers in ‘Person outside’





**Fig. 12** Examples illustrating the differences of accuracy in ‘Person outside’ sequence

- a PF
- b Meanshift
- c PSO
- d FA

better than the PF and PSO. Given that the meanshift-based tracker loses the target when the target is suffered from occlusion, the FA-based tracker outperforms the other trackers.

In the third example, ‘Basketball’ sequence, the target moves fast and irregularly and there are similar objects (e.g. other balls and basketball hoop) that may cause the tracker to lose its target. We choose this video to test the performances of the four versions in tracking object with rapid and erratic motion and similar object interference. Speed and accuracy comparisons from this example are shown in Table 3 and Fig. 13, respectively.

It can be seen from Table 3 that the time cost of FA, although a little bigger than the MS-based tracker, is much smaller than the other two trackers. Fig. 13 shows that the PF-based tracker loses the target quickly at the 10th frame and follows on the PSO-based tracker losing the target at the 15th frame. The tracking performance of MS-based tracker is better than the PF and PSO-based trackers. However, the target still loses at about the 62th frame where the target’s motion is rapid and erratic. Compared

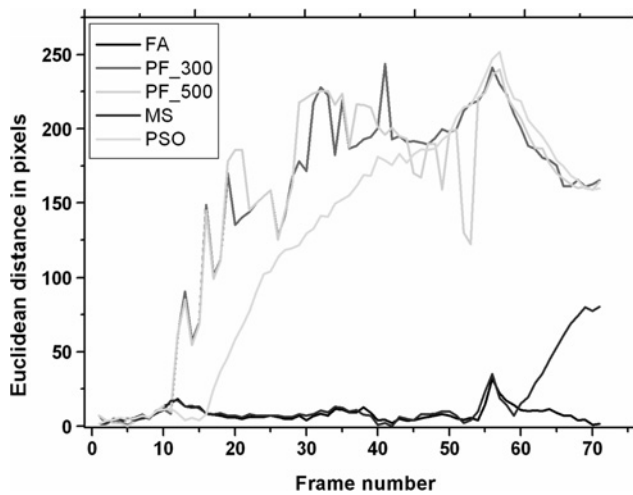
**Table 2** Average time cost for ‘Person outside’ sequence

Tracking methods	Time cost, ms
PF_300	15.6098
PF_500	30.3728
meanshift	7.71777
PSO	85.6481
FA	10.4688

**Table 3** Average time cost for ‘Basketball’ sequence

Tracking methods	Time cost, ms
PF_300	9. 6282
PF_500	15.4359
meanshift	3.1587
PSO	15.337
FA	3.3974





**Fig. 13** Tracking accuracy comparisons of different trackers in 'Basketball' sequence

with the three methods mentioned, the FA-based trackers can track the object successfully in the entire tracking process. Tracking examples from this sequence are shown in Fig. 14.

The last example, 'Kitbag' sequence, comes from a sequence recorded for the CAVIAR project [28]. In this example, the target kitbag has a large scale change and there are many passers which may interfere with the tracking performance. We chose this video to test the four trackers' abilities to track object with scale change and similar object interference. Tracking accuracy and speed comparisons from this example are shown in Fig. 15 and Table 4, respectively.

Table 4 shows that the speed comparison is similar to the previous examples with the meanshift being the fastest and the FA being the second fastest. However, it can be seen from Fig. 15 that FA still outperforms the meanshift because the tracking results of FA are more accurate than the meanshift-based tracker. Tracking examples from this sequence are shown in Fig. 16.

Before we move on to the conclusions it is worth mentioning the theoretical advantages of the FA-based



**Fig. 14** Examples illustrating the differences of accuracy in 'Basketball' sequence

- a PF
- b Meanshift
- c PSO
- d FA

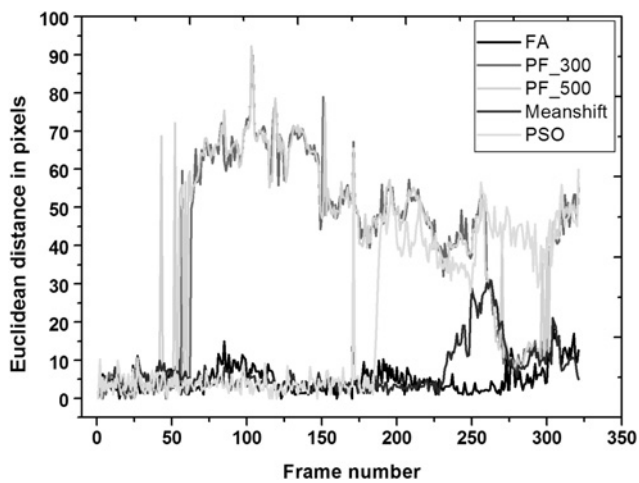


Fig. 15 Tracking accuracy comparisons of different trackers in ‘Kitbag’ sequence

tracker. Firstly, compared with PF, FA makes better use of the most recent observational information. In PF, the observational information is only used to evaluate

Table 4 Average time cost for ‘Kitbag’ sequence

Tracking methods	Time cost, ms
PF_300	29.6801
PF_500	55.069
meanshift	11.5482
PSO	122.124
FA	18.5421

the particles and is not used to change the location of the particles in the search space. In contrast, FA is based on swarm intelligence and the agents in the population can exchange information frequently based on the current observational information during the movement. Secondly, compared with the meanshift, FA is more robust to local distractors. As we know, most of models encountered in visual tracking are nonlinear, non-Gaussian, multi-modal or any combination of these. Meanshift is a gradient-based optimisation algorithm and it is computationally efficient. However, it may converge to local maximum and sensitive to background distractors, clusters, occlusions and quick moving objects [10]. Contrarily, in FA, the agents explore

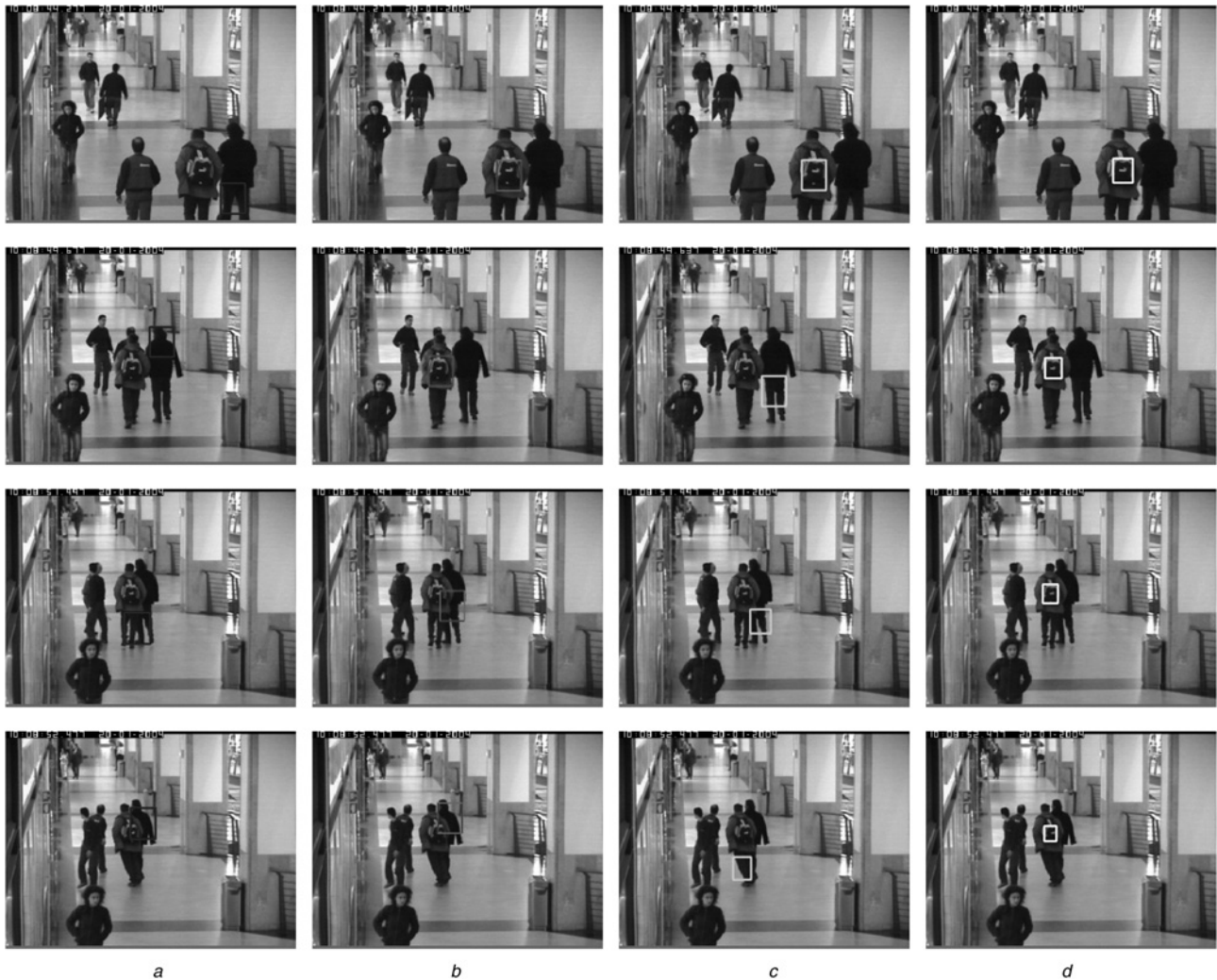


Fig. 16 Examples illustrating the differences of accuracy in ‘Kitbag’ sequence

- a PF
- b Meanshift
- c PSO
- d FA



the search space locally, aided by randomisation which increases the diversity of the solutions on a global scale, and thus there is a fine balance between local intensive exploitation and global exploration. Thirdly, as pointed out by Yang [17], compared with PSO, FA is more general. There are two important limiting cases in (6), say  $\gamma \rightarrow 0$  and  $\gamma \rightarrow \infty$ . If  $\gamma \rightarrow 0$ , the light intensity does not decrease in an idealised sky and a flashing firefly can be seen anywhere in the domain. It corresponds to a special case of PSO. If  $\gamma \rightarrow \infty$ , each firefly roams in a completely random way. As FA is usually in somewhere between these two extremes, it can outperform PSO by adjusting the parameter  $\gamma$  and  $\alpha$ .

## 6 Conclusions and future work

In this paper, a novel application of FA is presented to solve tracking problem. A general optimisation-based tracking architecture is proposed and the parameters' sensitivity and adjustment of the FA are studied. Experimental results show that the FA-based tracker can robustly track an arbitrary target in various challenging conditions. We compare the speed and accuracy of the FA with three typical tracking algorithms including the PF, meanshift and PSO. Comparative results show that the current version of the FA-based tracker outperformed the PF, meanshift and PSO in all of our experiments. To the author's knowledge this is the first time that the FA has been adapted for use in a visual tracking system and our initial results showed it to be a superior alternative to the popular PF, meanshift and PSO approaches.

It is worthwhile to note that in the current version only a single object is tracked. How to track multiple objects in the FA based tracking system would be left behind to further research. Besides, in Section 4, the parameter tuning is made under the assumption that these parameters are not correlated and can be optimised independently. Relaxing this assumption would likely deliver more optimal values and further study into this is anticipated. Furthermore, recent developments from research in the adaptive appearance model showed that on-line boosting feature selection algorithm for tracking is a good way to adjust the tracking algorithm to various challenging tracking environments [29, 30]. Therefore it is the theme of ongoing research to introduce this feature selection algorithm into our tracking system.

## 7 Acknowledgments

The authors thank Jaco Fourie from the University of Canterbury and Xin-she Yang from the University of Cambridge for their kind suggestions and helps. This work is supported by NSAF under grant no. 11176018 and National Natural Science Foundation of China under grant no. 61071161. Many thanks to the anonymous reviewers for their valuable comments that helped to improve this paper.

## 8 References

- 1 Szpak, Z.L., Tapamo, J.R.: 'Maritime surveillance: tracking ships inside a dynamic background using a fast level-set', *Expert Syst. Appl.*, 2011, **38**, (6), pp. 6669–6680
- 2 Huang, C.M., Fu, L.C.: 'Multitarget visual tracking based effective surveillance with cooperation of multiple active cameras', *IEEE Trans. Syst. Man Cybern. B*, 2011, **41**, (1), pp. 234–247
- 3 Akella, M.R.: 'Vision-based adaptive tracking control of uncertain robot manipulators', *IEEE Trans. Robot.*, 2005, **21**, (4), pp. 747–753
- 4 Coulaud, J.B., Campion, G., Bastin, G., De Wan, M.: 'Stability analysis of a vision-based control design for an autonomous mobile robot', *IEEE Trans. Robot.*, 2006, **22**, (5), pp. 1062–1069
- 5 Grigorescu, S.M., Macesanu, G., Cocias, T.T., Puiu, D., Moldoveanu, F.: 'Robust camera pose and scene structure analysis for service robotics', *Robot. Auton. Syst.*, 2011, **59**, (11), pp. 899–909
- 6 Tadayyon, H., Lasso, A., Kaushal, A., Guion, P., Fichtinger, G.: 'Target motion tracking in MRI-guided transrectal robotic prostate biopsy', *IEEE Trans. Bio-Med. Eng.*, 2011, **58**, (11), pp. 3135–3142
- 7 Maggio, E., Smerladi, F., Cavallaro, A.: 'Adaptive multifeature tracking in a particle filtering framework', *IEEE Trans. Circuits Syst. Video*, 2007, **17**, (10), pp. 1348–1359
- 8 Kalman, R.E.: 'A new approach to linear filtering and prediction problems', *J. Basic Eng.*, 1960, **82**, (1), pp. 35–45
- 9 Arulampalam, M.S., Maskell, S., Gordon, N., Clapp, T.: 'A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking', *IEEE Trans. Signal Process.*, 2002, **50**, (2), pp. 174–188
- 10 Comaniciu, D., Ramesh, V., Meer, P.: 'Kernel-based object tracking', *IEEE Trans. Pattern Anal.*, 2003, **25**, (5), pp. 564–577
- 11 Zhang, X.Q., Hu, W.M., Maybank, S., Li, X., Zhu, M.L.: 'Sequential particle swarm optimization for visual tracking'. IEEE Conf. Computer Vision and Pattern Recognition (CVPR) Anchorage, AK, USA, 2008, pp. 1–8
- 12 Zhang, X.Q., Hu, W.M., Wang, X.Y., et al.: 'A swarm intelligence based searching strategy for articulated 3D human body tracking'. 2010 IEEE Computer Society Conf. Computer Vision and Pattern Recognition Workshops (CVPR Workshops), San Francisco, CA, USA, 2010, pp. 45–50
- 13 John, V., Trucco, E., Ivekovic, S.: 'Markerless human articulated tracking using hierarchical particle swarm optimisation', *Image Vis. Comput.*, 2010, **28**, (11), pp. 1530–1547
- 14 Minami, M., Agbanhan, J., Asakura, T.: 'Manipulator visual servoing and tracking of fish using a genetic algorithm', *Ind. Robot., Int. J.*, 1999, **26**, (4), pp. 278–289
- 15 Fourie, J., Mills, S., Green, R.: 'Harmony filter: a robust visual tracking system using the improved harmony search algorithm', *Image Vis. Comput.*, 2010, **28**, (12), pp. 1702–1716
- 16 Gao, M.L., He, X.H., Luo, D.S., Yu, Y.M.: 'Object tracking based on harmony search: comparative study', *J. Electron. Imag.*, 2012, **21**, (4), p. 043001
- 17 Yang, X.S.: 'Firefly algorithms for multimodal optimization. Stochastic algorithms: foundations and applications'. Proc. Fifth Int. Symp., SAGA 2009, pp. 169–178
- 18 Chandrasekaran, K., Simon, S.P.: 'Firefly algorithm for reliable/emission/economic dispatch multi objective problem', *Int. J. Electr. Eng. Educ.*, 2012, **7**, (1), pp. 3414–3425
- 19 Yang, X.S., Hosseini, S.S.S., Gandomi, A.H.: 'Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect', *Appl. Soft Comput.*, 2012, **12**, (3), pp. 1180–1186
- 20 Jumadinova, J., Dasgupta, P.: 'Firefly-inspired synchronization for improved dynamic pricing in online markets'. Proc. 2008 Second IEEE Int. Conf. Self-Adaptive and Self-Organizing Systems (SASO), Venice, Italy, October 2008, pp. 403–412
- 21 Gandomi, A.H., Yang, X.S., Alavi, A.H.: 'Mixed variable structural optimization using firefly algorithm', *Comput. Struct.*, 2011, **89**, pp. 2325–2336
- 22 Ming-Huwi, H.: 'Vector quantization using the firefly algorithm for image compression', *Expert Syst. Appl.*, 2012, **39**, (1), pp. 1078–1091
- 23 Yang, X.S.: 'Introduction to mathematical optimization from linear programming to metaheuristics' (Cambridge, UK, 2008)
- 24 Yang, X.S.: 'Review of meta-heuristics and generalised evolutionary walk algorithm', *Int. J. Bio-Inspired Comput.*, 2011, **3**, (2), pp. 77–84
- 25 Yang, X.S., Deb, S.: 'Engineering optimisation by cuckoo search', *Int. J. Math. Model. Numer. Optimisation*, 2010, **1**, (4), pp. 330–343
- 26 Yang, X.S.: 'A new metaheuristic bat-inspired algorithm', *Stud. Comput. Intell.*, 2010, **284**, pp. 65–74
- 27 Fourie, J.: 'The application of harmony search in computer vision'. PhD thesis, University of Canterbury, 2011
- 28 EC funded CAVIAR project/ist 2001 37540, available at: <http://www.homepages.inf.ed.ac.uk/rbf/CAVIAR/> (July 2008)
- 29 Parag, T., Porikli, F., Elgammal, A.: 'Boosting adaptive linear weak classifiers for online learning and tracking'. Proc. 2008 IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Anchorage, AK, USA, June 2008, pp. 1–8
- 30 Grabner, H., Sochman, J., Bischof, H., Matas, J.: 'Training sequential on-line boosting classifier for visual tracking'. ICPR 2008 19th Int. Conf. Pattern Recognition, Tampa, FL, USA, December 2008, pp. 1–4