Zhang *et al. EURASIP Journal on Image and Video Processing* (2020) 2020:4
https://doi.org/10.1186/s13640-020-0491-y

EURASIP Journal on Image and Video Processing

RESEARCH Open Access

Hybridizing extended ant lion optimizer with sine cosine algorithm approach for abrupt motion tracking

Huanlong Zhang[1], Zeng Gao[1], Jie Zhang[1], Junfeng Liu[1], Zhicheng Nie[1] and Jianwei Zhang[2*]

Abstract

In view of the problem that the conventional tracker does not adapt to abrupt motion, we propose a tracking algorithm based on the hybrid extended ant lion optimizer with sine cosine algorithm (EALO-SCA) in this paper. Firstly, the multiple elites is used to replace the single elite in the standard ant lion optimizer (ALO). The extended ALO (EALO) can enhance the global exploration ability, which can handle abrupt motion. Secondly, considering that sine cosine algorithm (SCA) has strong local exploitation operator, a hybrid EALO-SCA tracker is proposed using the advantages of both EALO and SCA. The proposed approach can improve tracking accuracy and efficiency. Finally, extensive experimental results in both quantitative and qualitative measures prove that the proposed algorithm is very competitive compared to 7 state-of-the-art trackers, especially for abrupt motion tracking.

Keywords: Abrupt motion, Sine cosine algorithm, Ant lion optimizer, Multiple elites

1 Introduction

Visual tracking is a fundamental problem pertinent to many real-world applications including video surveillance, autonomous vehicle navigation, human-computer interaction, medical imaging, and many more. Although visual tracking has been studied for decades, it remains a challenging problem due to various factors such as partial occlusion, fast or abrupt motion, illumination changes, motion blur, etc. To further improve tracking performance, many tracking methods have been proposed [1, 2]. These methods can be roughly divided into generative model [3, 4] and discriminative model [5, 6].

In recent visual tracking, significant attention has been paid to discriminative correlation filters (DCF) [7, 8] and deep learning [9, 10] based methods. Subsequently, many improved DCF-based methods have been proposed because of the high efficiency, such as scale adaptive kernel correlation filter tracker (SAMF) [11] and accurate scale estimation for robust visual tracking (DSST) [12] for scale variation, spatially regularized correlation filter (SRDCF) [13] and context-aware correlation filter tracking (CACF) [14] to reduce boundary effects, etc. However, most of these methods use manual features, which reduces their accuracy and robustness. Therefore, methods based on deep learning have attracted attention of researchers. Considering the effectiveness of DCF, many methods have been proposed to improve the robustness of the tracker by replacing the manual feature with the depth feature, such as hierarchical convolutional features (HCF) [15], deep features for SRDCF (DeepSRDCF) [16], etc. Of course, another trend of deep trackers is to design a network and pre-train it to improve tracking performance, such as convolutional residual learning for visual tracking (CREST) [17], cascaded region proposal networks (C-RPN) [18], etc.

Although achieving state-of-the-art performance, most of these methods assume that the tracked target has smooth motion conditions. However, the abrupt or fast motion of targets often occur in real-world scenarios, which will make the traditional methods difficult to track targets well. In order to solve these problems, many methods have been proposed, including detection based tracking methods [19, 20] and motion model based tracking methods [4, 21].

*Correspondence: zhl_lit@163.com
[2]Software engineering college, Zhengzhou University of Light Industry, No.5 Dongfeng Road, 450002 Zhengzhou, People's Republic of China
Full list of author information is available at the end of the article

Generally speaking, the first consideration method to handle the problem of abrupt motion is that it has the strong global search ability. Therefore, I would say that it can be regarded as an optimization problem, which has attracted more and more attention of researchers [22]. For example, Gao et al. [23] introduced firefly algorithm into visual tracking, which had a strong superiority toward other trackers in accuracy and speed. Zhang et al. [24] proposed a sequential particle swarm optimization (PSO)-based visual tracking, which was integrated sequential information into PSO method to form a robust tracking framework. This method was more robust and effective in arbitrary motion or large appearance changes. Nguyen et al. [25] presented a modified bacterial foraging optimization algorithm, which was introduced into real-time tracking with changes. The method had a good merit compared with PSO and particle filter in speed and accuracy. These proposed methods have some advantages in visual tracking. However, each optimization algorithm has its limitations, there is no single optimization algorithm that can solve all the problems perfectly according to no free lunch (NFL) theorem [26].

As we all know, exploration and exploitation are two important components of any meta-heuristic algorithm. The purpose of global exploration is to find the best desired region in a wider search area. Local exploitation is to find a better solution by further searching some areas of the group, based on the prior knowledge or the new information found during the search process. Therefore, an algorithm can improve convergence speed and accuracy by properly balancing exploration and exploitation performance. Hybridizing two algorithm is the latest research trend for solving global optimization to overcome the poor exploration ability of one algorithm and poor exploitation ability of the other algorithm. There are many hybrid meta-heuristic algorithms such as hybrid gray wolf optimizer and genetic algorithm [27], krill herd-differential evolution [28], hybrid bat algorithm with harmony search [29], hybrid gravitational search algorithm with PSO [30], etc. Of course, many researchers have introduced hybrid optimization algorithms to visual tracking. Chen et al. [31] proposed a euclid distance based hybrid quantum PSO, which overcame the problem that population diversity reduced during the latter period of evolution in PSO. It improved tracking efficiency and decreased detection time cost. Hao et al. [32] proposed a particle filtering algorithm based on ant colony optimization, which enhanced the performance of particle filter with a small sample set. The method improved the efficiency of visual tracking. Ljouad et al. [33] presented the hybrid Kalman cuckoo search tracker using a modified cuckoo search algorithm combined with the Kalman filter. The proposed algorithm had better performance than PSO-based tracker. These hybrid optimization algorithms have improved tracking performance of the original optimization algorithm. In addition, in recent years, we have focused a lot of attention on the problem of abrupt motion tracking. Zhang et al. [34, 35] proposed extended kernel correlation filter (KCF) tracker based on swarm intelligence and extended cuckoo search-based KCF tracker. A unified framework was designed to capture abrupt motion. The performance of the hybrid approach is better than that of the standard algorithm in these literatures.

In recent years, Mirjalili [36, 37] presented two novel nature-inspired algorithms, ALO and SCA, respectively. These two optimization algorithms had been applied in lots of fields successfully [38, 39]. Subsequently, some improved methods were proposed [40–43]. Especially, Mirjalili presented that the SCA could be hybridized with other algorithms in the field of stochastic optimization to improve its performance in[37], and lots of hybrid methods had been applied [44–46]. Thus, a hybrid EALO-SCA is proposed in order to solve abrupt motion tracking in the paper (see Fig. 1). The approach leverages the complementary properties between exploratory stage of EALO and the exploitation operator of SCA to improve tracking performance.

The main contribution of our work includes three folds:

(1) EALO, an optimization algorithm called EALO is proposed to enhance the global search ability, thus increasing the diversity of candidate samples to handle abrupt motion.
(2) EALO-SCA, a hybrid EALO-SCA is proposed using global exploration of EALO and local exploitation of SCA, which can properly balance exploration and exploitation
(3) A unified tracking framework is designed based on EALO-SCA, which can improve tracking performance.

2 The basic sine cosine algorithm and ant lion optimizer

2.1 Sine cosine algorithm (SCA)

SCA is a population-based optimization algorithm that depends on sine and cosine operators, differing from other optimization algorithms [47], for updating the movement of the search agents toward the best solution [37]. Each search agent in the initial population updates its position with respect to the best solution using the Eq. (1).

$$X_i^{t+1} = \begin{cases} X_i^t + r_1 \times \sin(r_2) \times \left| r_3 X^{*t}{}_i - X_i^t \right|, r_4 < 0.5 \\ X_i^t + r_1 \times \cos(r_2) \times \left| r_3 X^{*t}{}_i - X_i^t \right|, r_4 \geq 0.5 \end{cases} \tag{1}$$

where X_i^t indicates the position of the i-th individual at the t-th iteration, $X^{*t}{}_i$ is the position of the best individual obtained so far in i-th individual, r_4 is a random number in

Fig. 1 Example tracking results of different methods. Sequence 1–2: BLURFACE, ZT

$[0, 1]$. r_1 , r_2, and r_3 are random numbers. r_4 is a random number in [0,1], and || indicates the absolute value. Basic principle of SCA is represented in Fig. 2.

Figure 2 shows that the current solution will be away from the destination and explore the search space when the individual moves in $[-2, -1)$ or $(1, 2]$. However, the current solution will be toward the destination and exploit the search space when the individual moves in $[-1, 1]$.

Exploration means that particles need to be searched in a wider area. Exploitation is that particles are searched locally in small areas. Therefore, the exploration and exploitation phases seek conflicting objects. For SCA, the

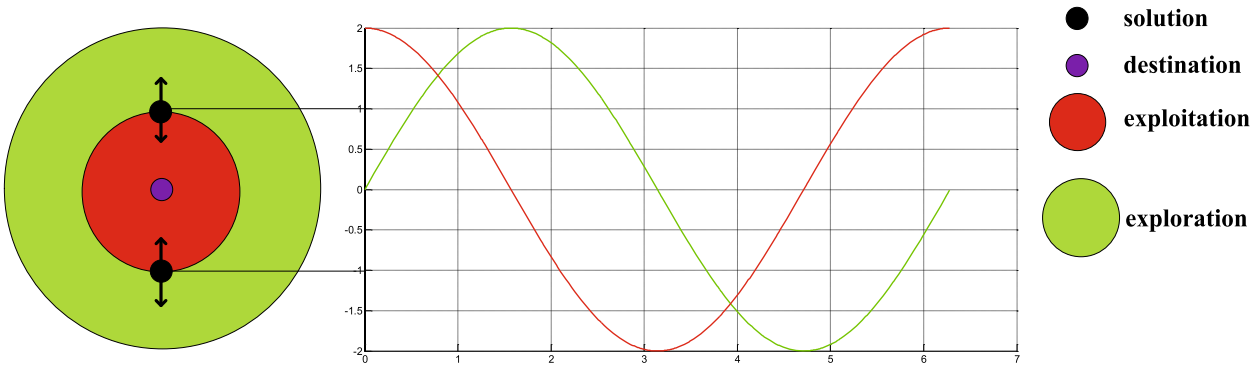

Fig. 2 Updating toward or away from destination [37]

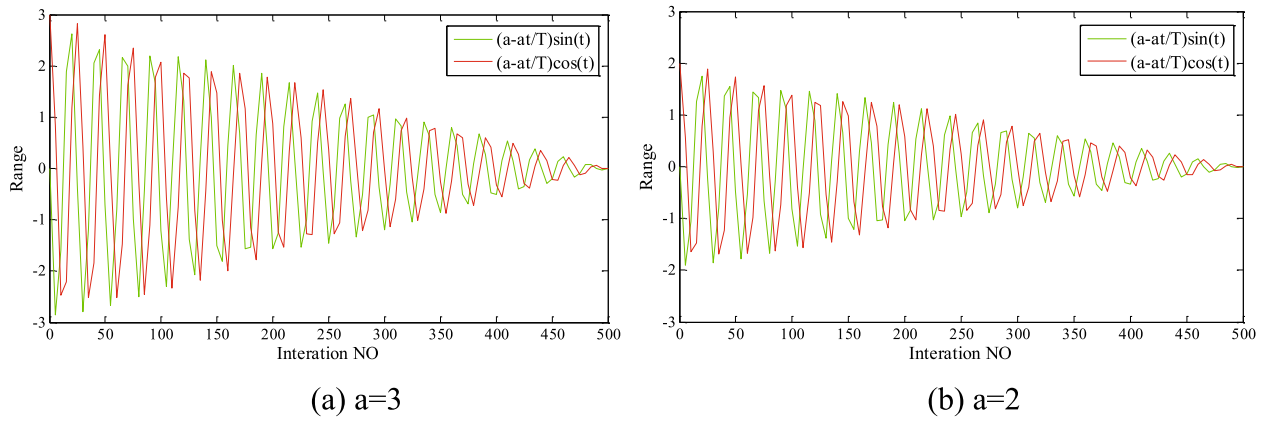

Fig. 3 Decreasing range of the sine and cosine with the increase of iterations [37]. **a** $a = 3$, **b** $a = 2$

range of sine and cosine is decreased during optimization using Eq. (2) to balance exploration and exploitation.

$$r_1 = a - t\frac{a}{T} \tag{2}$$

where a is a constant, t is the current iteration, and T is the maximum number of iterations. Figure 3 shows how this equation decreases the range of sine and cosine functions with the increase of iterations.

It can be seen from Fig. 3 that the range of the search space gradually decreases and the exploitation becomes more and more important with the increase of iterations. In addition, the range of Fig. 3b is smaller than Fig. 3a. This phenomenon shows that $a = 2$ is easier to enter the exploitation phase than $a = 3$, but the ability to explore is weakened. That is to say, the parameter a leads to balanced exploration and exploitation of the search space. In this study, we set the parameter model to be $a = 2$. In a word, the basic SCA utilizes Eq. (1) as a special path to converge to global optimum. Therefore, the basic SCA performs well in exploitation operator.

2.2 The ant lion optimizer (ALO)

The ALO algorithm mimics the hunting mechanism of antlions in nature [36]. This behavior can be summarized into five main steps: the random walk of ants, building traps, entrapment of ants in traps, catching prey, and re-building traps. The following subsections discuss the mathematical model in details.

2.2.1 The random walk of ants

This random walk is chosen for modeling ants' movement as follows:

$$\begin{aligned} Q(m) =& [0, \text{cumsum}(2r(m_1) - 1), \text{cumsum}(2r(m_2) \\ & -1), ..., \text{cumsum}(2r(m_T) - 1)] \end{aligned} \tag{3}$$

where cumsum calculates the cumulative sum, T shows the maximum number of iteration, m is the step of random walk (iteration in this study), and $r(m)$ is a random number 0 or 1. In order to ensure that the ants are walking within the search space, they are normalized using the following equation:

$$Q_i^t = \frac{(Q_i^t - a_i) \times (d_i^t - c_i^t)}{(b_i - a_i)} + c_i \tag{4}$$

where a_i is the minimum of random walk of i-th variable, b_i shows the maximum of random walk in i-th variable, c_i^t indicates the minimum of i-th variable at t-th iteration, and d_i^t is the maximum of i-th variable at t-th iteration.

2.2.2 Building traps

Random walks of ants are affected by antlions' traps using mathematic model:

$$c_i^t = \text{Antlion}_j^t + c^t, d_i^t = \text{Antlion}_j^t + d^t \tag{5}$$

where c^t is the minimum of all variables at t-th iteration, d^t is the vector including the maximum of all variables at t-th iteration, Antlion_j^t shows the position of the selected j-th antlion at t-th iteration. In this study, the best antlion obtained so far in each iteration is saved and considered as an elite. Since the elite is the fittest antlion, it should be able to affect the movements of all the ants during iterations. Therefore, the ants' position update can be expressed as:

$$\text{Ant}_i^t = \frac{R_A^t + R_E^t}{2} \tag{6}$$

where R_A^t is the random walk around the antlion selected by the roulette wheel at t-th iteration, R_E^t is the random walk around the elite at t-th iteration, and Ant_i^t indicates the position of i-th ant at t-th iteration.

2.2.3 Entrapment of ants in traps

When an ant fell into a trap, antlions shoot sands outwards the center of the pit, so ants gradually moved to the antlion. Finally, antlions preyed on ants. To simulate this behavior, $c^t = c^t / I$ and $d^t = d^t / I$ are proposed. Parameter I controls the trade-off between exploration and exploitation in the ALO. I is a ratio ($I = 10^w \frac{t}{T}$). w is described as follows:

$$w = \begin{cases} 2; t > 0.1T \\ 3; t > 0.5T \\ 4; t > 0.75T \\ 5; t > 0.9T \\ 6; t > 0.95T \end{cases} \tag{7}$$

where t is the current iteration, T is the maximum number of iterations. w gradually increases, then I gradually increases, but c^t and d^t gradually decrease.

2.2.4 Catching preys and re-building traps

The final stage of hunt is when an ant reaches the bottom of the pit and is caught in the antlion's jaw. An antlion is then required to update its position to the latest position of the hunted ant to enhance its chance of catching new prey. The following equation is proposed in this regard:

$$\text{Antlion}_j^t = \text{Ant}_i^t \quad if \quad f\left(\text{Ant}_i^t\right) > f\left(\text{Antlion}_j^t\right) \tag{8}$$

where Antlion_j^t shows the position of selected j-th antlion at t-th iteration, and Ant_i^t indicates the position of i-th ant at t-th iteration. The antlion will update its position where the ant is last caught and rebuild the trap to hunt the next ant.

3 The extended ant lion optimizer method

In the section, the motivation and implementation of EALO algorithm are explained in detail.

3.1 Motivation of EALO

In the original ALO algorithm, the updated position of ant for the next generation is given by Eq. (6). However, it is observed that the roulette wheel selection method is more likely to be selected R_E^t. That is, ants tend to move around single elite. In addition, the information of single elite is extremely limited, which will lead to local trapping problem. Therefore, it is necessary to establish elite library for storing better individual (antlion).

3.2 Implementation of EALO

Based on the above problems, we introduce multiple elites, and they are designed as an elite library (as shown in Eq. (10)). During the early search, multiple elites parallel competition can not only enhance exploration ability of the original ALO algorithm, but also ensure the convergence speed of the algorithm. However, n becomes smaller during the later search, and it can reduce the unnecessary movement of ants. In conclusion, EALO enhances the optimization capability, meanwhile, the efficiency of the original ALO has improved. Elite library is introduced into Eq. (6)in order to achieve these effects and the resultant modified equation is written as in Eq. (9).

Algorithm 1 Pseudo-code of the EALO

```
Initialize: the population of ants and antlions,Max_iter
Calculate the fitness of ants and antlions
Find the best antlion and assume it as the elite
while Current_iter<Max_iter+1 do
  for every ant do
    Select an antlion using Roulette wheel
    Update c and d using Eqs. (9) and (10)
    Create a random walk and normalize it using Eqs. (3) and (5)
    Update the position of ant using Eq. (13)
  end for
  Calculate the fitness of all ants
  Replace an antlion with its corresponding ant if it becomes fitter (Eq. (12))
  Update elite library if an antlion becomes fitter than the elite(Eq. (15))
end while
Return elite
```

$$\text{Ant}_i^t = \frac{R_A^t + \frac{1}{n}R_{E1}^t + \frac{1}{n}R_{E2}^t + \cdots + \frac{1}{n}R_{En}^t}{2} \tag{9}$$

$$n = \text{round}(10 - \frac{9t}{T}) \tag{10}$$

where n is the number of elites, t is the current iteration, T is the maximum number of iteration, R_{En}^t is the random walk around the n-th elite at t-th iteration. Elite library is described as follows:

$$\text{sort\_antlion}_p^t \rightarrow \text{elite}_q^t \; f\left(\text{sort\_antlion}_p^t\right) > f\left(\text{elite}_q^t\right) \quad p = 1, 2...N; q = 1, 2, ...n \tag{11}$$

where f is fitness function, N is the number of antlions, $\rightarrow$ is the former n antlions as the elite library of the next iteration.

Equations (9), (10), and (11) are used instead of Eq. (6) in the original ALO to form a new optimization algorithm, which we call EALO. This algorithm enhances the global optimization ability through parallel cooperation of multiple elites and avoids trapping in local optima. The pseudo-code of the EALO is as shown in Algorithm 1.

The main difference between ALO and EALO is how the ants' location is updated. In addition, the EALO contains the concept of elite library compared with ALO.

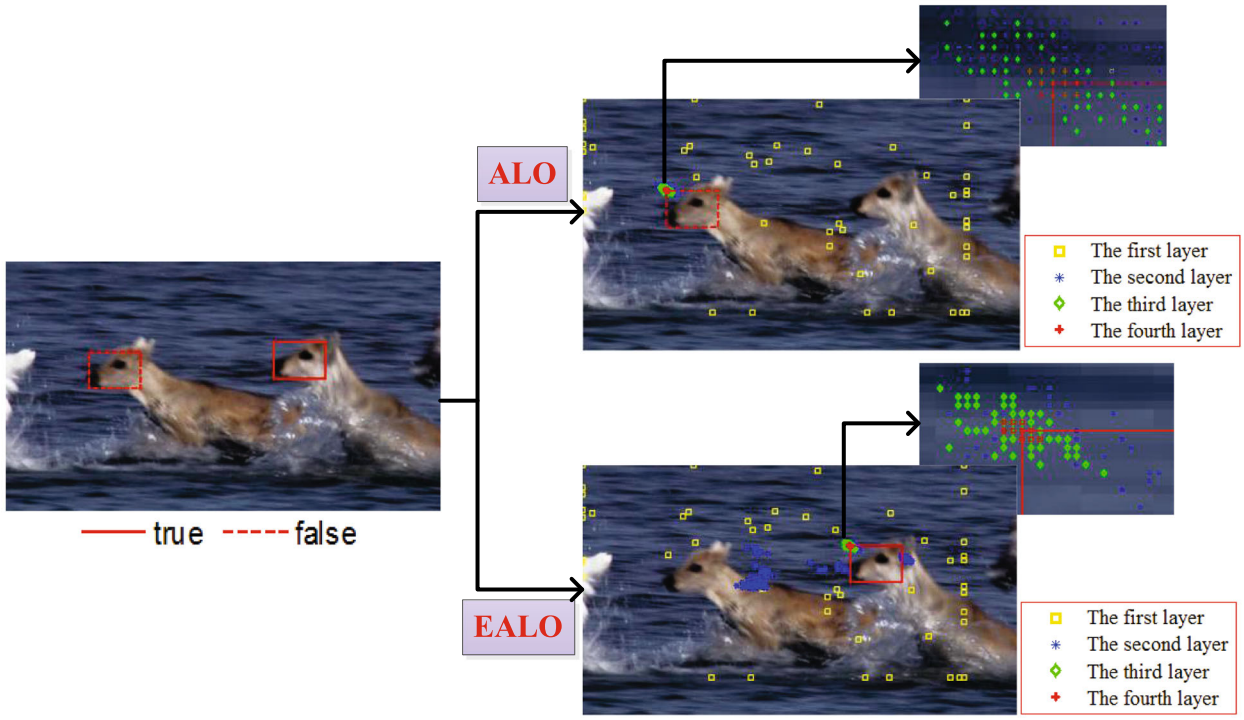

Fig. 4 The tracking result with adaptive boundary shrinking mechanism

3.3 Performance evaluation of EALO

In order to verify the feasibility of our proposed method, we selected the #0043 frame of the DEER video sequence, as shown in Fig. 4, which had two similar targets. Further, to make a fair comparison, the same target model (histogram of oriented gradient, HOG), motion model (random walk model), and parameters were used. As aforementioned, there are three main parameters, namely Np (population size), T (the number of iterations), and I (the shrinkage factor, the value of I heavily depends on w), respectively. In this study, $n = 75$; $T = 200$; $t > 0.5T, w = 1; t > 0.7T, w = 2; t > 0.9T,$ and $w = 2.7$.

Figure 4 shows the searching process using adaptive boundary shrinking mechanism. Firstly, at the beginning of the first layer, we guarantee that original ALO and EALO have the same initial position of search agent. Secondly, in the second layer, all search agents in the original ALO algorithm always randomly walk around the false DEER. In contrast, EALO has several different search areas since it has elite library. Thirdly, in the third and fourth levels, the original ALO algorithm always searches around the false DEER and it is difficult to jump out of the local optimum because of single elite. However, the EALO can track the target successfully. And as shown by Eq. (10), the number of elites decreases with the increase of iterations. Therefore, in the later stages of the search, EALO can reduce unnecessary search space and ensure convergence speed. Moreover, we repeat this experiment 60 times. The original ALO is successful 47 times, and the success rate is 78.3%. The EALO is successful 60 times, and the success rate is 100%. Considering the above, the proposed EALO has some advantages in this paper.

4 The EALO-SCA tracking method

In this paper, in order to enhance the exploitation performance of EALO, a hybrid EALO-SCA is proposed, which combines the better exploratory performance of EALO with the better exploitation performance of SCA. Visual tracking is considered to be a process of searching for target by various ants and antlions in sequential images. To this end, an EALO-SCA-based tracking framework is presented as shown in Fig. 5.

As shown in Fig. 5, the target is first chosen by the user in the first frame, and the state vector $X = [x, y, s]$ is initialized, where $[x, y]$ denotes the location of the target pixel coordinates, and s is the scale parameter. Secondly, according to a dynamic model, the predicted position of the target is given in the next frame. Then, the features of candidate sample generated by EALO-SCA and template are extracted. Finally, the target candidate with the largest similarity value is found by similarity function, which is the target. In this work, our main contribution is how to switch thresholds between EALO and SCA to generate candidate sample images.

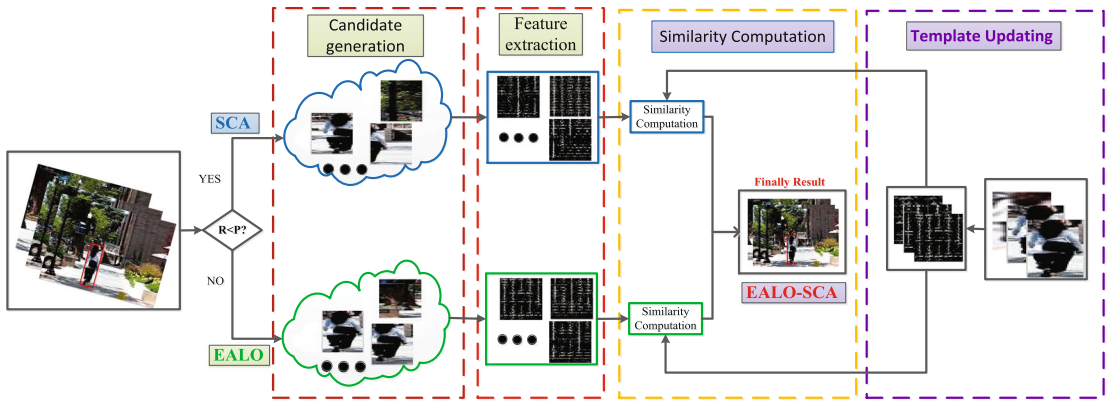

Fig. 5 The flowchart of the EALO-SCA tracker

4.1 Basic idea

Although EALO enhances significantly in the exploratory phase and avoids falling into local optimum, the method suffers from a lack of an efficient exploitation operator due to random walk rather than specific paths. Conversely, the SCA has an efficient exploitation operator and suffers from a lack of an exploration phase, and SCA adopts a specific path to the target solution (the best solution so far). Therefore, a novel hybrid EALO-SCA is proposed, which can balance exploration and exploitation. In hybrid EALO-SCA, EALO algorithm is utilized for global search, which makes most solution move toward a more promising area. After exploration phase, SCA is utilized to search locally with a small step to get the best solution.

The first advantage of the hybrid algorithm is to introduce the updated formula of SCA into EALO, which will force all solutions to update their positions faster toward the target solution. This helps the EALO to enhance exploitation operator. The second advantage of the hybrid algorithm is that it can enhance the performance of SCA in the exploratory phase. Because the solution in SCA is limited to $[-2, 2]$ and cannot move further away from the destination.

Considering the above, EALO-SCA has obvious advantages over the EALO and SCA.

4.2 The proposed tracking method

Suppose there is target in the image being searched. And a group of target candidates are randomly generated in the image. The aim of EALO-SCA tracker is to find the "best" candidate using the proposed hybrid algorithm. In the EALO-SCA tracker, ants and antlions can be considered as search candidates and storage candidates, respectively. They are both target candidates. Search candidates and storage candidates are the locations generated by ants and antlions using the EALO-SCA. In addition, ants and antlions are mixed, and the individuals with high fitness value are selected as antlions (storage candidates).

4.2.1 The fitness function

The fitness function measures the values of correlation between the target and the target candidate. In addition, the HOG is the structural feature of the extracted edge with well geometric and optical invariance. Therefore, their similarity is computed as:

$$\rho(X, Y) = \frac{\operatorname{cov}(X, Y)}{\sqrt{D(X)}\sqrt{D(Y)}} \tag{12}$$

where $D(\cdot)$ denotes the variance and $\mathrm{Cov}(\cdot)$ denotes covariance. X and Y are the HOG feature of the target and candidate sample respectively. The fitness function is defined as follows:

$$E = 2 + 2 * \rho(x, y) \tag{13}$$

where $\rho(X, Y)$ indicates the value of the correlation coefficient, and the function $E(s)$ to be minimized is analogous to the internal energy of the system. Finally, the optimal value points to the best target candidate as the target image.

4.2.2 The global motion model

Recently, tracking algorithms, using motion models to capture uncertain motion, have attracted much attention. In traditional visual tracking, a random walk model, a nearly constant velocity model, and a Gaussian distribution model are commonly selected. Obviously, the three models cannot solve abrupt motion tracking effectively. Therefore, we design a global motion model to estimate the position of candidate image samples.

$$x_{t+1} = K_t \times (\mathrm{EALO}) + (1 - K_t) \times (\mathrm{SCA}) \tag{14}$$

where $x = [x, y]$ is vector and it denotes a pixel location, K is the model parameter and is obtained as follows:

$$K_{t+1} = \begin{cases} 1 \ if \ R > P \\ 0 \ if \ R \leq P \end{cases} \quad (15)$$

In the next iteration, the model selection depends on the parameter K. R is a random number in [0,1]. P shows the threshold and its analysis is detailed in section 4.3. Equations (14) and (15) are used to design a global motion model, which combines the effective exploratory stage of EALO with the effective exploitation operator of SCA. The purpose of global motion model is to solve abrupt motion tracking.

4.2.3 The tracking implement

In the EALO-SCA tracker, target candidates are firstly generated by motion model of EALO-SCA. Then, the features of template and target candidate are extracted, and the best target candidate is found by similarity measurement. Finally, the best target candidate is used as the output of the current frame and the template of the next frame. In addition, EALO-SCA tracker makes use of EALO's better global search ability and SCA's better local search ability to improve tracking accuracy and efficiency. The pseudocode of proposed hybrid EALO-SCA tracker is shown in Algorithm 2.

Algorithm 2 Pseudo-code of the EALO-SCA tracker

Input: Image sequence
Initialize: Locate the target object in the first frame manually; The initial number of ants and antlions(Np), Max_iter(T), the threshold(P); EALO: The shrinkage factor(I); SCA: Constant(a), r_2, r_3, r_4
Calculate the fitness of ants (search candidates) and antlions (storage candidates)
Find the best antlion and assume it as the elite;
Tracking:
for i from 2 to the last frame **do**
  while Current_iter<Max_iter+1 **do**
    for every ant **do**
      Update r_1,r_2,r_3,r_4
      R=create a random number between 0 and 1
      if (R≤0.5) **then**
        if (r_4 ≤0.5) **then**
          $X_i^t + r_1 \times \sin(r_2) \times \left| r_3 P_i^t - X_i^t \right|$
          else if (r_4 >0.5)
          $X_i^t + r_1 \times \cos(r_2) \times \left| r_3 P_i^t - X_i^t \right|$
        end if
        else if (R>0.5)
        Select an antlion using Roulette wheel
        Update c and d using Eqs. (9) and (10)
        Create a random walk and normalize it using Eqs. (3) and (5)
        Update the position of ants using Eq. (13)
      end if
    end for
    Calculate the fitness of all ants
    Replace an antlion with its corresponding ant(if it becomes fitter (Eq. (12))
    Update elite library if an antlion becomes fitter than the elite(Eq. (15))
  end while
  Display the elite indicating the best antlion, which is the target
end for

4.3 Parameters' sensitivity and adjustment

Parameter tuning is often a key aspect in optimization algorithms. The speed and accuracy should be considered simultaneously during the parameter tuning. In hybrid EALO-SCA tracker, there are four main parameters, namely Np (the numbers of ants and antlions), I (the shrinkage factor), T (the numbers of iteration), and P (threshold), respectively. In this study, we discuss two parameters Np and P; then, the parameter I and T are designed to be $t > 0.5T, w = 1; t > 0.7T, w = 2; t > 0.9T, w = 2.7$ and 500. We choose BOY video sequence to test tracking performance because the target undergoes fast motion and motion blur caused by camera shaking. In this video sequence, their maximum displacement between frames reaches 21 pixels. The BOY video sequence is available on the website http://www.visual-tracking.net.

We first analyze the population size Np. Other parameters are fixed. We test the performance of the EALO-SCA tracker by calculating the number of lost frames, the number of frames where the distance precision between its upper left corner and the ground-truth is more than 17, and costing time. The result is shown in Fig. 6.

Figure 6 shows that when the population size $Np < 150$, the number of lost frames relatively decreases along with the increase of Np. In other words, the accuracy is not achieved. However, when the population size $Np > 150$, a large amount of time is wasted. That is to say, the tracking efficiency is reduced. Therefore, comprehensive considering the tracking accuracy and speed, we set the population size of the parameter model to be $Np = 150$.

Another parameter is the threshold P. The proper threshold P makes an appropriate trade-off between exploration and exploitation, which is key factor for a good tracking result. The Euclidean distance between the ground-truth and estimated by the different thresholds is calculated in each frame visually, and the comparisons results are shown in Fig. 7. It can be seen from Fig. 7 that the EALO-SCA performs well in the BOY video sequence at threshold $P = 0.5$. It can track the target successfully. However, other parameters make the method fail to track the whole BOY video sequence successfully.

In addition, we show partial sequences of BOY video to visualize the results as shown in Fig. 8. All the selected

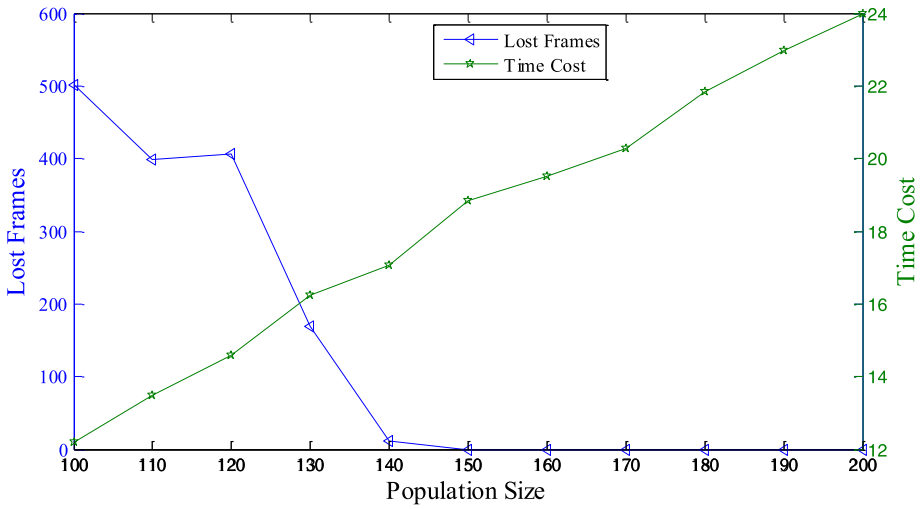

Fig. 6 Performance comparison with different population size *Np*

parameters P can track by the tracker except for $P = 0.3$ at frame #0085. Only the selected parameters $P = 0.5$ and $P = 0.1$ can track the target at frame #0115, and $P = 0.5$ completes the whole video sequence successfully. Thus, we set the threshold of the parameter model to be $P = 0.5$.

5 Experimental results and discussion

We implemented the proposed tracker in MATLAB R2014a. The experiments were conducted on a PC with Intel Core i5-7500 3.40GHz and 8GB RAM. In order to verify the effectiveness of EALO-SCA tracker, the experiment is divided into three parts:

1. Comparison of tracking performance among EALO-SCA, SCA, ALO, and EALO
2. Comparison of the state-of-the-art trackers
3. Discussion

In these experiment, the parameters of EALO-SCA are set as follows: the threshold ($P = 0.5$); EALO, the shrinkage factor ($t > 0.5T, w = 1; t > 0.7T, w = 2; t > 0.9T, w = 2.7$; t is the current number of iteration); and SCA, the constant ($a = 2$), $r_2 \in [0, 2\pi]$, $r_3 \in [0, 2]$ and $r_4 \in [0, 1]$. The parameters of SCA, ALO, and EALO are consistent with EALO-SCA.

5.1 Comparison of tracking performance among EALO-SCA, SCA, ALO, and EALO

Figure 9 shows the relation between the convergence accuracy and the parameters setup, including the population size and the iteration number. The X-axis represents the population size, the Y-axis shows the

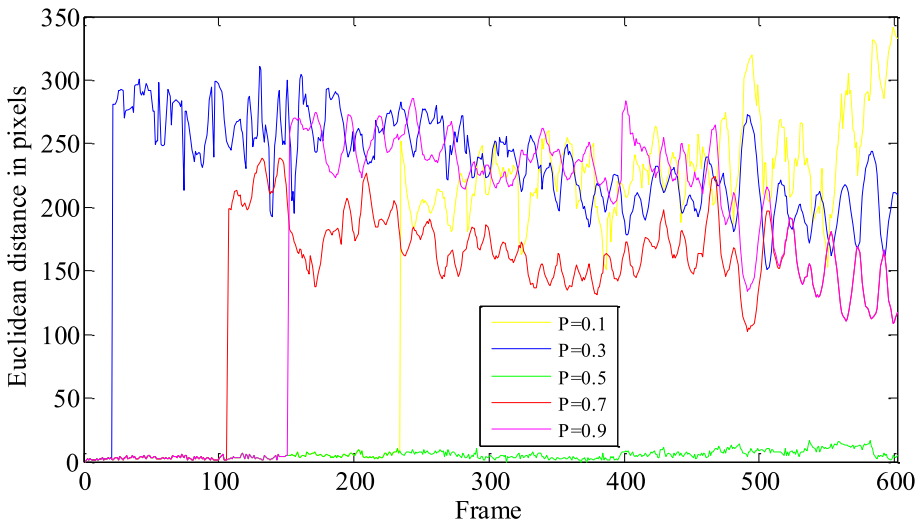

Fig. 7 Tracking accuracy comparisons of different threshold *P*. *P*=0.1, 0.3, 0.5, 0.7, 0.9

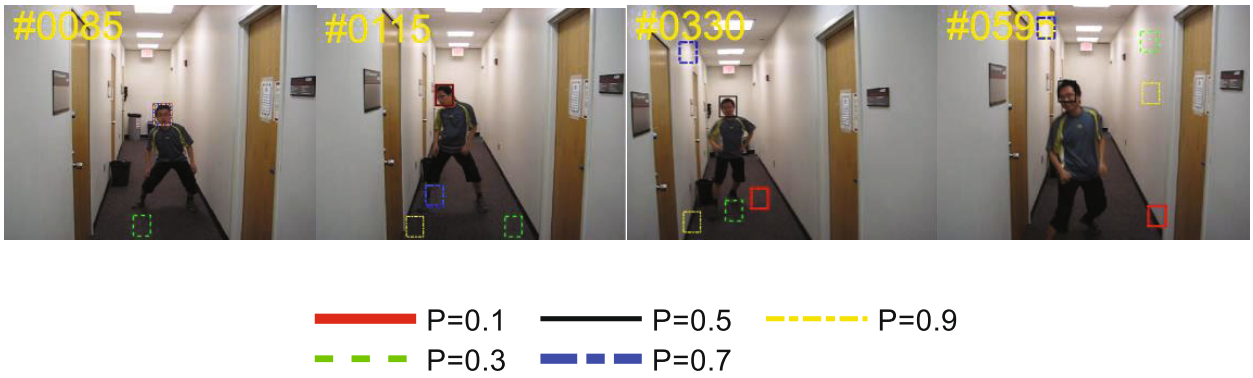

Fig. 8 A visual tracking results with different threshold *P*. *P*=0.1, 0.3, 0.5, 0.7, 0.9

iteration number, and the *Z*-axis denotes the similarity value.

It can be clearly seen from Fig. 9 that SCA-based tracker has the best result when $Np = 50$ and $T = 60$, and its fitness value is 0.9961. It still shows poor performance because SCA does not have good global exploratory performance. The result of EALO-based tracker is better than that ALO-based tracker because EALO has the ability to jump out of local optimum. In addition, the tracker based on EALO-SCA shows the best performance and requires less population size and iteration number to achieve stable tracking results than the other three optimization algorithms. EALO-SCA tracker combines EALO's global exploration with SCA's local exploitation, which makes an appropriate trade-off between exploration and exploitation for improving tracking performance.

5.2 Comparison of the state-of-the-art trackers

To prove the feasibility of the proposed tracker, we selected 12 video sequences in the experiment. The source of the FACE1 is the dataset AVSS2007. ZT, FHC, and ZXJ are our own. Other sequences are available on the website http://www.visual-tracking.net (listed in Table 1). Note that we extracted the frames 306-310 in BLURFACE video sequence, which could represent the problem of frame dropping. Thus, it can provide scenes for abrupt motion. The population size and iteration number are set to 150 and 500, respectively, in EALO-SCA tracker.

We compared our tracker EALO-SCA with 7 state-of-the-art trackers(including DSST [12], fast compressive tracking (FCT) [48], kernelized correlation filters (KCF) [49], exploiting the circulant structure of kracking-by-detection with kernels (CSK) [50], fast tracking via spatio-temporal context learning (STC) [51], least soft-threshold squares tracking(LSST) [52], and CACF [14]).In order to ensure the consistency of the experiment, the parameters of our tracker are consistent in the experiment. The experimental data of other trackers come from the experimental results executed by source code. In this experiment, we divided the 12 video sequences into 3 groups based on

Table 1 The video sequences

Video	Frame	Max displacement	*X* max displacement	*Y* max displacement
MHYANG	1490	7	7	4
FISH	476	15	15	13
BOY	602	21	21	19
HUMAN7	250	31	31	21
JUMPING	313	36	18	36
DEER	71	38	38	34
FACE1	380	39	22	39
ZXJ	118	70	70	18
BLURBODY	334	76	76	26
FHC	123	188	188	104
BLURFACE	488	202	202	71
ZT	115	256	256	149

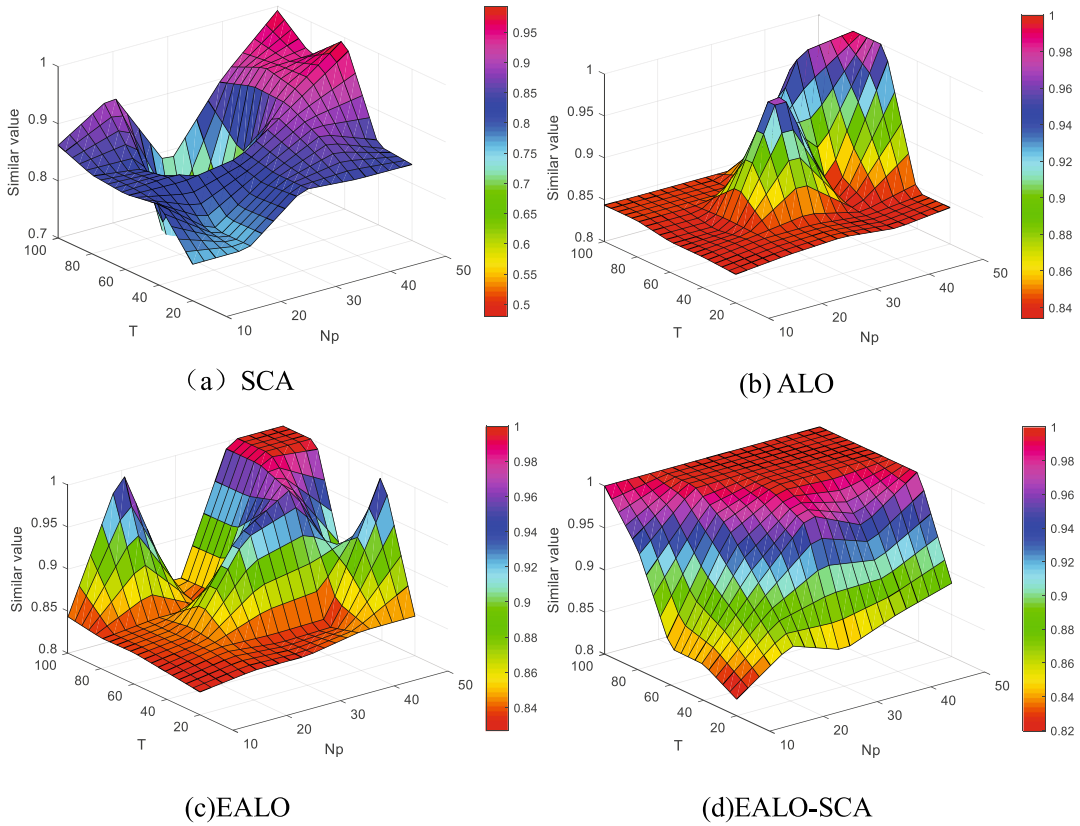

Fig. 9 Similar values against population size *Np* and number of iteration *T*. **a** SCA, **b** ALO, **c** EALO, **d** EALO-SCA

the target's displacement between image frames, including the slight motion group, the middle motion group, and the large motion group. The displacement of slight motion group is less than 35 pixels, including the MHYANG, FISH, BOY, and HUMAN7 video sequences. The middle motion group contains the JUMPING, DEER, FACE1, and ZXJ video sequences, whose motion displacement is more than 35 pixels and less than 75 pixels. The displacement of large motion group is more than 75 pixels, the video sequences are BLURBODY, FHC, BLURFACE, and ZT, respectively.

Additionally, to evaluate the accuracy of the trackers, the videos were manually labeled by identifying the upper left corner of the tracked target in each frame visually. The tracking results were evaluated by using distance precision (DP), center location error (CLE), and overlap precision (OP) in [53]. DP is the relative number of frames in the sequence where the center location error is smaller than a certain threshold. CLE is computed as the average Euclidean distance between the ground-truth and tracking result. OP is defined as the percentage of frames where the bounding box overlap exceeds a threshold $t \in (0, 1)$.

5.2.1 Qualitative analysis

The slight motion group The tracking results with the slight motion are shown in Fig. 10. MHYANG video sequence has smallest motion at 7 pixels, it has the larger illumination changing at frame #0378. FCT has slightly worse tracking results, while others have similar performance. In FISH video sequence, it is obvious that there is camera shake at frame #0058 and #0312, and the brightness is dimmed at frame #0178. Nearly all trackers can catch up with the target successfully except for CSK. Our tracker and KCF perform best. BOY video sequence has severely blurred at frame #0330 because of camera shaking. At first, LSST is failure at frame #0330, #0508, and #0595, respectively. Then, STC is failure at frame #0508 and #0595, respectively, and CSK deviates the target because of serious blur. At the end, CSK is failure at frame #0595, and FCT slightly drifted away from the target. Other trackers complete the whole video sequence. For HUMAN7 video sequence, it is covered by tree shadows several times at frame #0090, #0168, and #0208. All trackers are failure except for CACF and our tracker. Although the target is shaded under the tree, our tracker still can track the target and get the better performance. All in

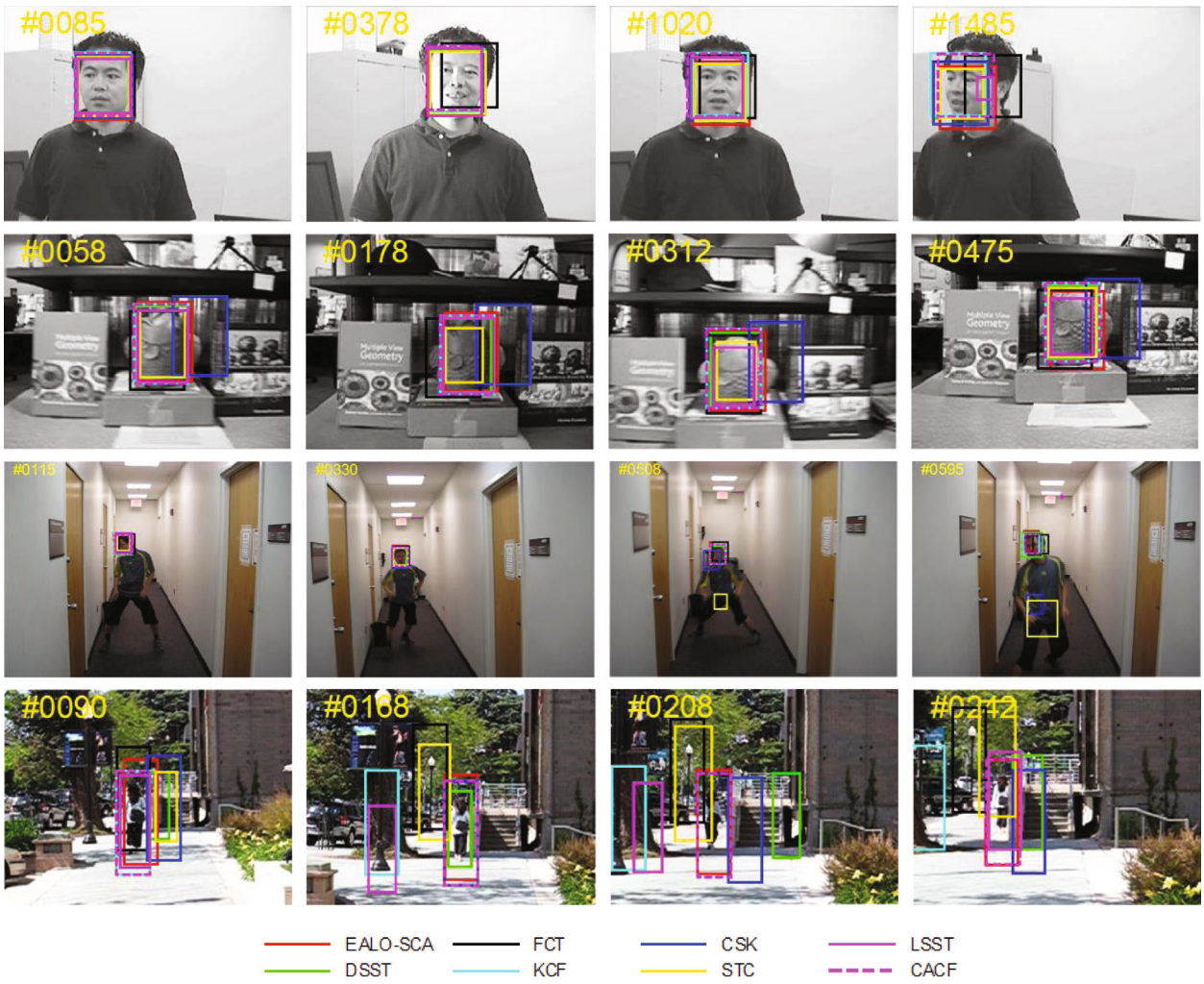

Fig. 10 The tracking results with the slight motion. Sequence 1-4: MHYANG, FISH, BOY, HUMAN7

all, our tracker can keep the better performance with the slight motion group.

The middle motion group In order to verify the excellent performance of our tracker, we continue to enhance the motion displacement. The tracking results with the middle motion are shown in Fig. 11. It is obvious that JUMPING video sequence has been seriously blurred because of the camera shaking. In this case, CSK, DSST, FCT, KCF, and STC lose the target unfortunately before the frame #0096. Although our method drifts the object slightly, it recovers quickly. And our tracker has a better performance. In DEER video sequence, the target undergoes fast motion and multiple similar targets. DSST, KCF, STC, and CACF have been lost target at frame #0032. Although KCF and CACF recover on several frames, they still cannot complete the whole video sequence. However, FCT, CSK, and our tracker complete the whole video sequence, and CSK and our tracker obtain the better tracking results. FACE1 video sequence displays that the scale changes occur in video frames. Obviously, CSK and LSST have lost target at frame #0210. Although other methods work well, DSST shows the best performance. In ZXJ video sequence, the target experiences abrupt motion. All tracks can track the target before at frame #0042, but all tracks deviate the target besides CACF and our tracker at frame #0069. In addition, LSST deviates the target because of abrupt motion, but it can recover tracking. Our tracker obtains the best performance. On the whole, for the middle motion group, our tracker achieves the better track performance compared with other track methods.

The large motion group We continue to enhance the motion displacement. For the large motion group, we choose BLURBODY, FHC, BLURFACE, and ZT video

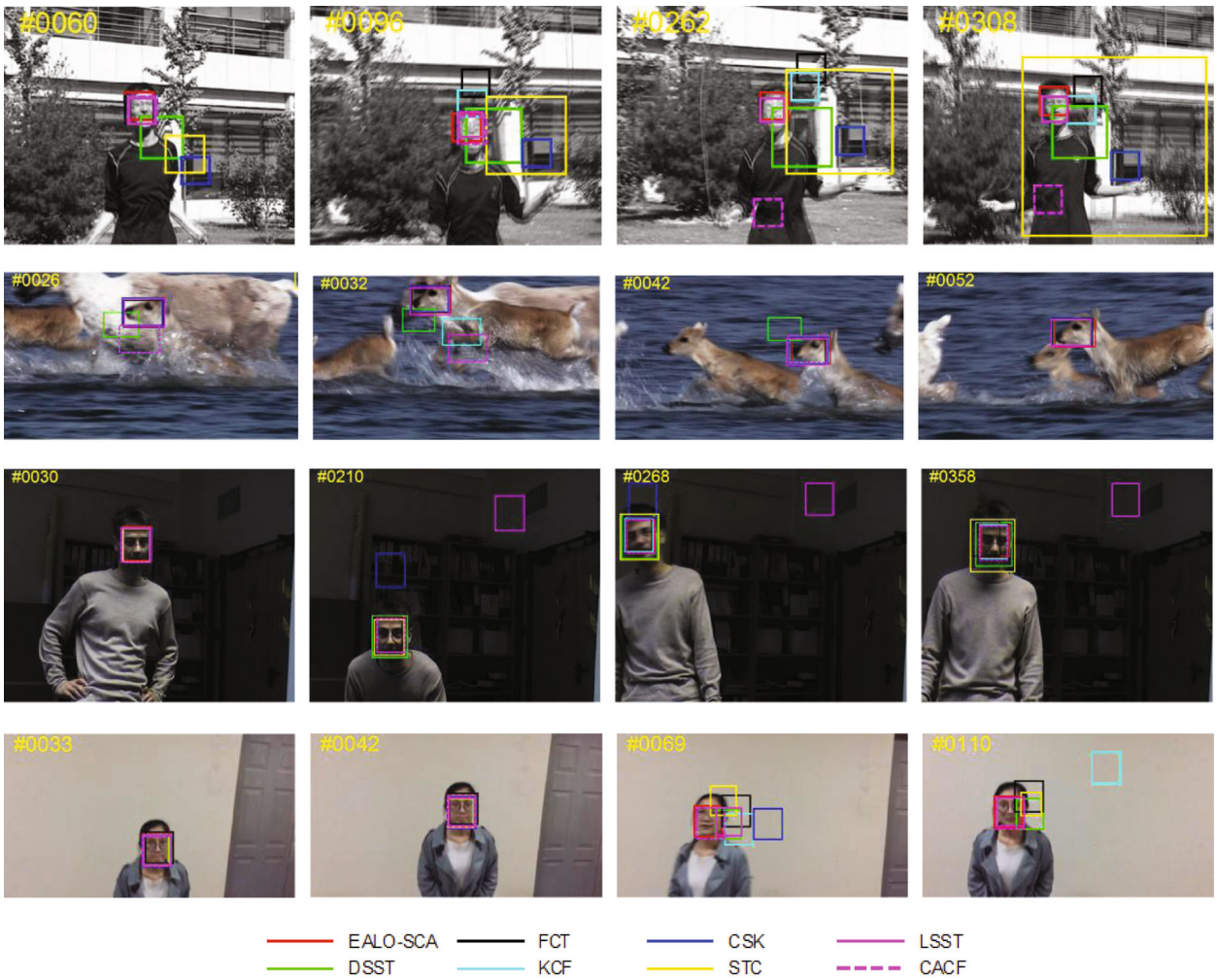

Fig. 11 The tracking results with the middle motion. Sequence 1-4: JUMPING, DEER, FACE1, ZXJ

sequences, and their maximum displacements are 76, 188, 202, and 256 pixels, respectively. The tracking results with the large motion are shown in Fig. 12. In BLURBODY video sequence, the target goes through a large displacement, and there was a very severe blur because of the camera shaking. Firstly, LSST is failure at frame #0100. Then, CSK and STC are also failure at frame #0212. Finally, all tracks lose the target at frames #0271 except for CACF and our tracker. Our tracker obtains the best performance. For FHC video sequence, the maximum displacements are 188 pixels. Our tracker performs much better than other trackers, while other trackers all fail at frame #0073 except for CACF and our tracker. For BLURFACE video sequence, we design the problem of the frame dropping. And there has a severe motion blur because of fast motion at the frames #0153, #0241, and #0310. Other trackers are failure, but our tracker still can track the target successfully. ZT video sequence has the largest motion at 256 pixels, all tracks deviate or lose the target except for CACF and our tracker at frame #0046. Although some tracking methods recover at frame #0105, they still cannot complete the whole video sequence. Therefore, our tracker and CACF obtain the better performance. In a word, our tracker has a strong superiority toward other trackers for the larger motion group.

5.2.2 Quantitative analysis

Figures 13 and 14 show the OP and DP of 12 different video sequences respectively. For Fig. 13, the X-axis represents the threshold, and the Y-axis represents the ratio between the number of frames, overlap is greater than the threshold, and the total number of frames. The larger the area under the curve, the better the tracker. For Fig. 14, the Y-axis also shows the ratio between the number of frames, distance of predicted and ground truth bounding box is below the threshold, and the total number of frames. The

Fig. 12 The tracking results with the large motion. Sequence 1-4: BLURBODY, FHC, BLURFACE, ZT

higher the slope, the better the tracking performance, this is because the center distance of more sequences is lower than the threshold. Figures 13 and 14 can vividly show the performance of different tracking methods, including DSST, FCT, KCF, CSK, STC, LSST, CACF and our tracker.

In addition, in order to clearly observe the tracking results of different trackers, we utilize data to display the tracking results. Tables 2 and 3 list a per-sequence comparison of our tracker to DSST, FCT, KCF, CSK, STC, LSST, and CACF, while Table 2 is concerned with average overlap rate and Table 3 refers to average center error rate. For the average overlap rate, the larger the value, the higher the tracking accuracy; however, for the center error rate, the smaller the value, the higher the tracking accuracy. In the tables, the best and second best results are shown in bold and italic. As can be seen from Tables 2 and 3, our tracker has a strong advantage in large displacement video sequences. At the same time, our tracker performs best on average in all videos.

As a result, it is obviously seen in Figs. 13, 14, Tables 2 and 3 that our tracker performs much better than 7 other trackers especially for large displacement video sequences. In other video sequences, our tracker also shows a better performance. As a whole, the proposed tracker has a good merit for abrupt motion tracking compared with other trackers.

5.3 Discussion

All the aforementioned experiments have validated the proposed tracker in smooth and abrupt motions. For these 12 video sequences, ZXJ, FHC, and ZT video sequences mainly go through abrupt motion. FACE1 video sequence faces scale variation, motion blur, and abrupt motion. The challenges of other video sequences can be seen in [53, 54].

FCT employs non-adaptive random projections, and through training compressive samples the classifier is update in time. It can handle pose variation well (e.g. MHYANG video sequence). However, it adopts sampling Haarlike features to represent the object appearance; when target experiences motion blur and the occlusion, the method loses or drifts the target frequently

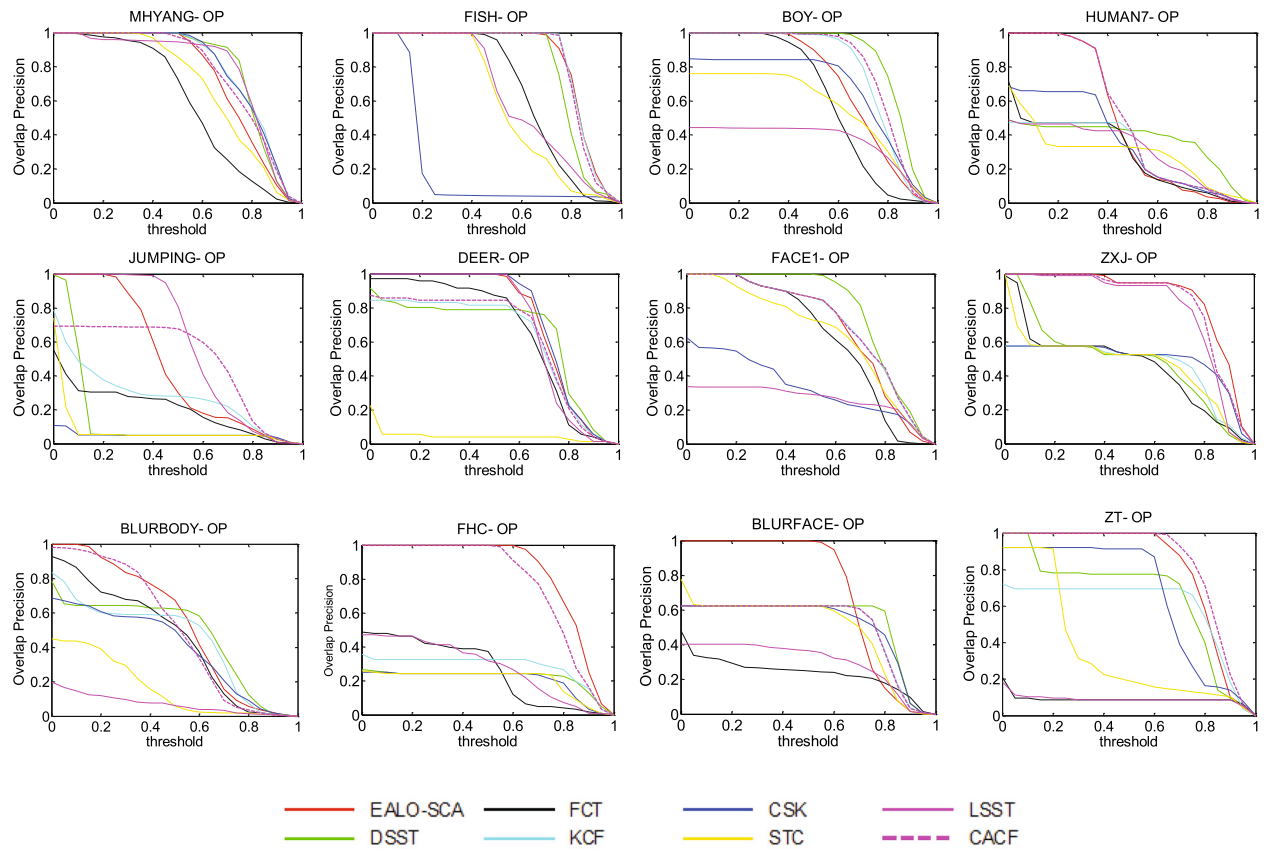

Fig. 13 The average precision of OP

(e.g., HUMAN7, JUMPING, BLURBODY, and BLURFACE video sequences). In addition, FCT obtains samples nearby the target position in the previous frame. Therefore, when the target appears large displacement motion, the sampling radius could not cover the uncertainty motion so that it is difficult to maintain tracking (e.g., ZXJ, FHC, and ZT video sequences).

CSK uses a density sampling strategy to process multiple sub-windows in a frame and classify them into a cyclic matrix. STC and CACF use context information to design a tracking framework so that they can handle background clutters very well (e.g., MHYANG video sequence). The main idea of the former is to model the spatio-temporal relationships between the object of interest and its locally dense contexts in a Bayesian framework. The latter presents a framework that allows the explicit incorporation of global context within correlation filter trackers. Therefore, CACF performs well in the above video sequences.

DSST and LSST show the relationship between all the candidates and the targets in different forms. They perform well in the smooth motion (e.g., MHYANG and FISH video sequences). However, when the tracking object encounters motion blur (e.g., HUMAN7 video sequence), or abrupt motion (e.g., BLURBODY, FHC, BLURFACE, and ZT video sequences), they perform the bad tracking results. The main difference between KCF and DSST is that DSST has scale variation. Therefore, the tracking result of DSST is generally better than that of KCF. However, they are hard to cover the uncertain motion states, which performs poor (e.g., ZXJ, BLURBODY, FHC, BLURFACE, and ZT video sequences).

Our proposed method shows the better performance in the smooth motion and abrupt motion since the method benefits the mechanism by using EALO and SCA strategies to produce samples. As shown in Tables 2 and 3, our tracker can maintain good performance in smooth motion (e.g., MHYANG, FISH, BOY, and HUMAN7 video sequences). Further, it exhibits the best tracking results in abrupt motion (e.g., ZXJ, BLURBODY, FHC, and BLURFACE video sequences). Meanwhile, the method also can cope with illumination variation and motion blur well.

6 Conclusion

A hybrid optimization called EALO-SCA is proposed in this paper, for solving abrupt motion tracking. Firstly, elite library is introduced into standard ALO, which

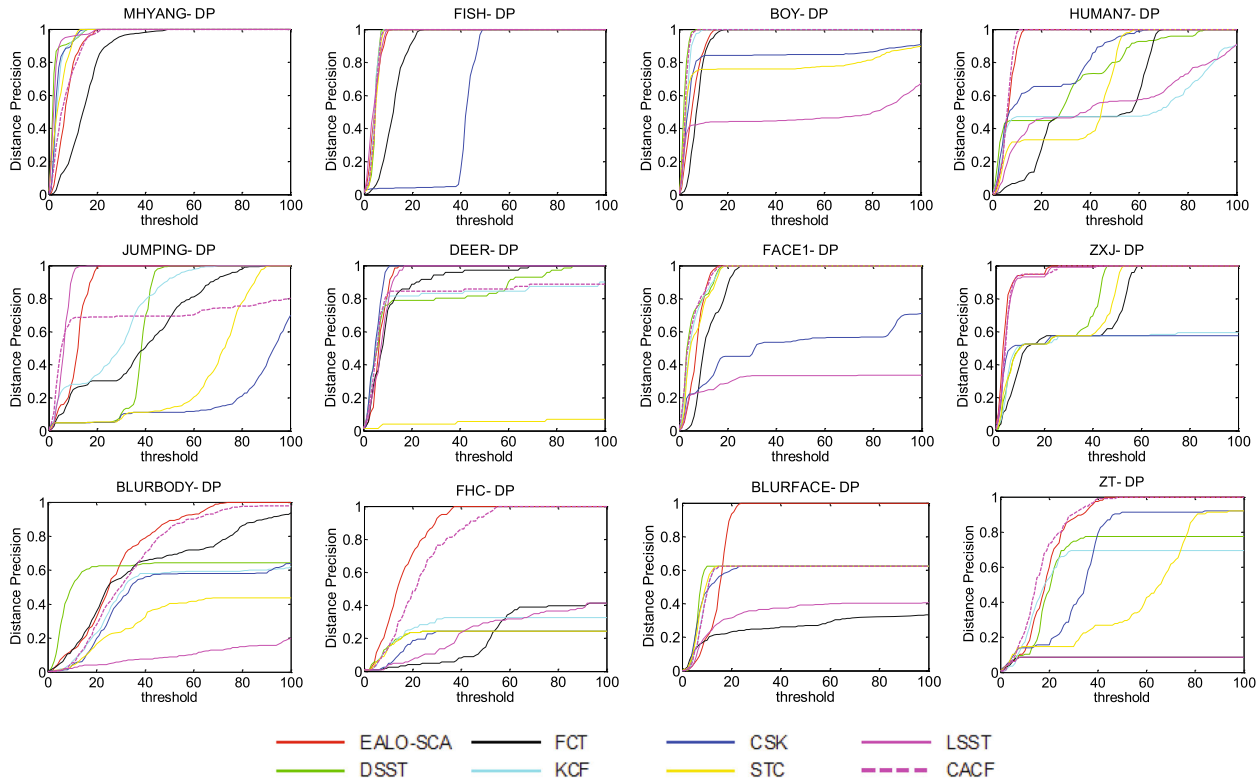

Fig. 14 The average precision of DP

enhances the global optimization capability to adapt to abrupt motion. Secondly, EALO enhances exploitation operator by hybridizing with SCA to improve tracking accuracy. Finally, a model switching method is designed and a unified tracking framework is constructed, which makes an appropriate trade-off between exploration and exploitation for improving tracking performance. The experimental results show that the algorithm has better tracking performance, especially for abrupt motion.

Table 2 Average overlap rate

Sequences	EALO-SCA	DSST	FCT	KCF	CSK	STC	LSST	CACF
MHYANG	0.75	**0.81**	0.59	*0.80*	*0.80*	0.69	0.78	0.78
FISH	**0.84**	0.80	0.66	**0.84**	0.21	0.58	0.63	0.83
BOY	0.69	**0.84**	0.60	0.77	0.65	0.55	0.36	*0.79*
HUMAN7	*0.47*	0.36	0.28	0.28	0.34	0.28	0.30	**0.49**
JUMPING	0.48	0.14	0.20	0.27	0.05	0.07	**0.60**	*0.50*
DEER	*0.74*	0.64	0.66	0.62	**0.75**	0.04	0.71	0.63
FACE1	0.69	**0.79**	0.63	*0.72*	0.33	0.65	0.26	*0.72*
ZXJ	**0.85**	0.48	0.45	0.45	0.49	0.46	0.79	*0.83*
BLURBODY	**0.54**	0.46	0.44	0.44	0.39	0.16	0.07	*0.50*
FHC	**0.83**	0.22	0.26	0.28	0.20	0.20	0.28	*0.78*
BLURFACE	**0.71**	*0.53*	0.23	0.51	0.51	0.30	0.51	0.51
ZT	*0.81*	0.65	0.09	0.59	0.66	0.35	0.09	**0.84**
Average	**0.70**	0.56	0.42	0.55	0.45	0.36	0.45	*0.68*

The best and second best results are shown in bold and italic

Table 3 Average center error rate

Sequences	EALO-SCA	DSST	FCT	KCF	CSK	STC	LSST	CACF
MHYANG	7.27	**2.43**	14.72	3.61	3.61	4.22	*2.60*	6.71
FISH	4.28	4.36	11.99	*4.08*	41.19	4.64	**3.97**	4.32
BOY	5.58	**1.96**	7.43	3.12	20.29	25.69	59.01	*2.47*
HUMAN7	*6.36*	25.67	40.84	48.43	17.89	33.07	45.29	**5.77**
JUMPING	*11.19*	36.55	37.23	26.26	85.72	66.70	**6.05**	33.84
DEER	*6.74*	16.65	10.68	21.13	**4.79**	509.55	7.20	22.96
FACE1	7.14	*5.30*	12.09	5.82	100.45	6.71	183.77	**5.08**
ZXJ	**4.08**	21.19	26.74	88.46	189.55	23.68	5.06	*4.84*
BLURBODY	**27.90**	90.77	40.68	68.35	73.24	147.65	208.55	*33.80*
FHC	**15.58**	616.05	371.37	364.79	577.79	576.47	392.42	*23.04*
BLURFACE	**14.19**	*74.93*	116.33	84.83	1573.68	89.75	162.05	111.94
ZT	*19.01*	54.01	642.35	127.53	53.52	99.65	684.85	**16.71**
Average	**10.78**	79.16	111.04	70.53	228.48	132.32	146.74	*22.62*

The best and second best results are shown in bold and italic

The contribution of this paper attempts to utilize swarm optimization method to handle abrupt motion. However, the method obtains a large number of iterations to ensure tracking accuracy, this will lead to a large amount of time consumption.

In the future, we will design a unified tracking framework based on state-of-the-art trackers and swarm optimization methods, so that the real-time tracking can be achieved by using swarm optimization when only the target undergoes large displacement between image frames.

Abbreviations

ALO: Ant lion optimizer; C-RPN: Cascaded region proposal networks; CACF: Context-aware correlation filter tracking; CLE: Center location error; CREST: Convolutional residual learning for visual tracking; CSK: Exploiting the circulant structure of tracking-by-detection with kernels; DeepSRDCF: Deep features for SRDCF; DCF: Discriminative correlation filters; DP: Distance precision; DSST: Accurate scale estimation for robust visual tracking; EALO-SCA: Extended ant lion optimizer with sine cosine algorithm; EALO: Extended ant lion optimizer; FCT: Fast compressive tracking; HCF: Hierarchical convolutional features; HOG: Histogram of oriented gradient; KCF: Kernel correlation filter; LSST: Least soft-threshold squares tracking; NFL: No free lunch; OP: Overlap precision; PSO: Particle swarm optimization; SAMF: Scale adaptive kernel correlation filter tracker; SCA: Sine cosine algorithm; SRDCF: Spatially regularized correlation filter; STC: Spatio-temporal context learning

Acknowledgements

The authors thank the editor and anonymous reviewers for their helpful comments and valuable suggestions.

Authors' contributions

HLZ designed the algorithm. ZG carried out the experiments and drafted the manuscript. JZ gave suggestions on the structure of manuscript and participated in modifying the manuscript. JFL and ZCN built the dataset and gave suggestions on the experimental analysis. JWZ participated in the survey of large motion and gave suggestions on the experimental analysis. All authors read and approved the final manuscript.

Funding

This work was supported in part by the National Natural Science Foundation of China (61873246, 61672471), the Graduate Innovation Foundation of Zhengzhou University of Light Industry (2018024).

Availability of data and materials

Please contact author for data requests.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]College of Electric and Information Engineering, Zhengzhou University of Light Industry, No.5 Dongfeng Road, Zhengzhou, People's Republic of China. [2]Software engineering college, Zhengzhou University of Light Industry, No.5 Dongfeng Road, 450002 Zhengzhou, People's Republic of China.

Received: 9 May 2019 Accepted: 31 December 2019
Published online: 21 January 2020

References

1. T. Zhou, H. Bhaskar, F. Liu, et al., Graph regularized and locality-constrained coding for robust visual tracking. IEEE Trans. Circ. Syst. Video Technol. **27**(10), 2153–2164 (2017)
2. T. Zhou, F. Liu, H. Bhaskar, et al., Robust visual tracking via online discriminative and low-rank dictionary learning. IEEE Trans. Cybern. **48**(9), 2643–2655 (2018)
3. T. Zhang, B. Ghanem, S. Liu, et al., Robust visual tracking via exclusive context modeling. IEEE Trans. Cybern. **46**(1), 51–63 (2016)
4. H. Zhang, S. Hu, X. Zhang, Sift flow for large-displacement object tracking. Appl. Opt. **53**(27), 6194–6205 (2014)
5. H. Zhang, Y. Wang, L. Luo, et al., Sift flow for abrupt motion tracking via adaptive samples selection with sparse representation. Neurocomputing. **249**(2), 253–265 (2017)
6. S. Hare, S. Golodetz, A. Saffari, et al., Struck: Structured output tracking with kernels. IEEE Trans. Pattern. Anal. Mach. Intell. **38**(10), 2096–2109 (2016)
7. M. Danelljan, A. Robinson, F. S. Khan, et al., in *Computer Vision – ECCV 2016*. Beyond correlation filters: Learning continuous convolution operators for visual tracking (Springer International Publishing, 2016), pp. 472–488. https://doi.org/10.1007/978-3-319-46454-1_29
8. M. Danelljan, G. Bhat, F. S. Khan, et al., in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Eco: Efficient convolution operators for tracking (IEEE, 2017). https://doi.org/10.1109/cvpr.2017.733
9. D. Held, S. Thrun, S. Savarese, in *Computer Vision – ECCV 2016*. Learning to track at 100 fps with deep regression networks (Springer International Publishing, 2016), pp. 749–765. https://doi.org/10.1007/978-3-319-46448-0_45
10. J. Valmadre, L. Bertinetto, J. F. Henriques, et al., in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. End-to-end

representation learning for correlation filter based tracking (IEEE, 2017). https://doi.org/10.1109/cvpr.2017.531
11. Y. Li, J. Zhu, in *European Conference on Computer Vision (ECCV)*. A scale adaptive kernel correlation filter tracker with feature integration, (2014), pp. 254–265. https://doi.org/10.1007/978-3-319-16181-5_18
12. M. Danelljan, G. Häger, F. S. Khan, et al., in *Proceedings of the British Machine Vision Conference 2014*. Accurate scale estimation for robust visual tracking (British Machine Vision Association, 2014). https://doi.org/10.5244/c.28.65
13. M. Danelljan, G. Hager, F. Shahbaz Khan, et al., in *2015 IEEE International Conference on Computer Vision (ICCV)*. Learning spatially regularized correlation filters for visual tracking (IEEE, 2015). https://doi.org/10.1109/iccv.2015.490
14. M. Mueller, N. Smith, B. Ghanem, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Context-aware correlation filter tracking (IEEE, 2017). https://doi.org/10.1109/cvpr.2017.152
15. C. Ma, J. B. Huang, X. Yang, et al., in *2015 IEEE International Conference on Computer Vision (ICCV)*. Hierarchical convolutional features for visual tracking (IEEE, 2015). https://doi.org/10.1109/iccv.2015.352
16. M. Danelljan, G. Hager, F. Shahbaz Khan, et al., in *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*. Convolutional features for correlation filter based visual tracking (IEEE, 2015). https://doi.org/10.1109/iccvw.2015.84
17. Y. Song, C. Ma, L. Gong, et al., in *2017 IEEE International Conference on Computer Vision (ICCV)*. Crest: Convolutional Residual Learning For Visual Tracking (IEEE, 2017). https://doi.org/10.1109/iccv.2017.279
18. H. Fan, H. Ling, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Siamese cascaded region proposal networks for real-time visual tracking, (2019), pp. 7952–7961. https://arxiv.org/pdf/1812.06148.pdf
19. D. A. Migliore, M. Matteucci, M. Naccari, in *Proceedings of the 4th ACM International Workshop on Video Surveillance and Sensor Networks – VSSN'06*. A revaluation of frame difference in fast and robust motion detection (ACM Press, 2006). https://doi.org/10.1145/1178782.1178815
20. A. Elgammal, D. Harwood, L. Davis, in *European Conference on Computer Vision (ECCV)*. Non-parametric model for background subtraction, (2000), pp. 751–767. https://doi.org/10.1007/3-540-45053-X_48
21. M. Khan, M. Valstar, T. Pridmore, in *Proc. Brit. Mach. Vis. Conf. (BMVC)*. A multiple motion model tracker handling occlusion and rapid motion variation, (2013), pp. 1–12. http://www.bmva.org/bmvc/2013/Workshop/0006.pdf
22. H. Zhang, X. Zhang, Y. Wang, et al., An experimental comparison of swarm optimization based abrupt motion tracking methods. IEEE Access. **6**, 75383–75394 (2018)
23. M. Gao, X. He, D. Luo, et al., Object tracking using firefly algorithm. IET Comput. Vis. **7**(4), 227–237 (2013)
24. X. Zhang, W. Hu, S. Maybank, in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. Sequential particle swarm optimization for visual tracking (IEEE, 2008). https://doi.org/10.1109/cvpr.2008.4587512
25. H. Nguyen, B. Bhanu, in *2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance*. Real-time pedestrian tracking with bacterial foraging optimization (IEEE, 2012). https://doi.org/10.1109/avss.2012.60
26. D. H. Wolpert, W. G. Macready, No free lunch theorems for optimization. IEEE Trans. Evol. Comput. **1**, 67–82 (1997)
27. M. A. Tawhid, A. F. Ali, A hybrid grey wolf optimizer and genetic algorithm for minimizing potential energy function. Memetic Comput. **9**(4), 347–359 (2017)
28. G. G. Wang, A. H. Gandomi, A. H. Alavi, et al., Hybrid krill herd algorithm with differential evolution for global numerical optimization. Neural Comput. Appl. **25**(2), 297–308 (2014)
29. G. Wang, L. Guo, A novel hybrid bat algorithm with harmony search for global numerical optimization. J. Appl. Math. **2013**, 1–21 (2013). https://doi.org/10.1155/2013/696491
30. S. Mirjalili, S. Hashim, *International Conference on Computer and Information Application (ICCIA). A new hybrid PSOGSA algorithm for function optimization (IEEE, 2010)*, (2010). https://doi.org/10.1109/ICCIA.2010.6141614
31. J. Chen, Y. Zhen, D. Yang, et al., Fast moving object tracking algorithm based on hybrid quantum PSO. WSEAS Trans. Comput. **12**(10), 375–383 (2013)
32. Z. Hao, X. Zhang, P. Yu, et al., in *2010 2nd International Conference On Advanced Computer Control (ICACC), vol. 3*. Video object tracing based on particle filter with ant colony optimization (IEEE, 2010), pp. 232–236. https://doi.org/10.1109/icacc.2010.5486857
33. T. Ljouad, A. Amine, M. Rziza, A hybrid mobile object tracker based on the modified Cuckoo search algorithm and the Kalman filter. Pattern Recog. **47**(11), 3597–3613 (2014)
34. H. Zhang, J. Zhang, Q. Wu, et al., Extended kernel correlation filter for abrupt motion tracking. KSII Trans. Internet Inf. Syst. **11**(9), 4438–4460 (2017)
35. H. Zhang, X. Zhang, Y. Wang, et al., Extended Cuckoo search-based kernel correlation filter for abrupt motion tracking. IET Comput. Vis. **12**(6), 763–769 (2018)
36. S. Mirjalili, The ant lion optimizer. Adv. Eng. Softw. **83**, 80–98 (2015)
37. S. Mirjalili, Sca: A sine cosine algorithm for solving optimization problems. Knowl.-Based Syst. **96**, 120–133 (2016)
38. H. M. Zawbaa, E. Emary, B. Parv, in *2015 Third World Conference on Complex Systems (WCCS)*. Feature selection based on antlion optimization algorithm (IEEE, 2015), pp. 1–7. https://doi.org/10.1109/icocs.2015.7483317
39. A. F. Attia, R. A. E. Sehiemy, H. M. Hasanien, Optimal power flow solution in power systems using a novel sine-cosine algorithm. Int. J. Electr. Power Energy Syst. **99**, 331–341 (2018)
40. E. Emary, H. M. Zawbaa, A. E. Hassanien, Binary ant lion approaches for feature selection. Neurocomputing. **213**(12), 54–65 (2016)
41. Y. Peng, H. Wang, Dynamic adaptive ant lion optimizer applied to route planning for unmanned aerial vehicle. Soft Comput. **21**(18), 5475–5488 (2017)
42. R. Sindhu, R. Ngadiran, Y. M. Yacob, et al., Sine-cosine algorithm for feature selection with elitism strategy and new updating mechanism. Neural Comput. Applic. **28**(10), 2847–2958 (2017)
43. K. S. Reddy, L. K. Panwar, B. K. Panigrahi, et al., A new binary variant of sine-cosine algorithm: Development and application to solve profit-based unit commitment problem. Arab. J. Sci. Eng. **43**(8), 4041–4056 (2018)
44. H. Nenavath, D. R. K. Jatoth, D. S. Das, A synergy of the sine-cosine algorithm and particle swarm optimizer for improved global optimization and object tracking. Swarm Evol. Comput. **43**, 1–30 (2018)
45. H. Nenavath, R. K. Jatoth, Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking. Appl. Soft Comput. **62**, 1019–1043 (2018)
46. S. Khalilpourazari, S. Khalilpourazary, SCWOA: An efficient hybrid algorithm for parameter optimization of multi-pass milling process. J. Ind. Prod. Eng. **35**(3), 135–147 (2018)
47. V. Dabra, K. K. Paliwal, P. Sharma, et al., Optimization of photovoltaic power system: a comparative study. Protect. Control Mod. Power Syst. **2**(2), 29–39 (2017)
48. K. Zhang, L. Zhang, M. H. Yang, Fast compressive tracking. IEEE Trans. Pattern. Anal. Mach. Intell. **36**(10), 2002–2015 (2014)
49. J. F. Henriques, R. Caseiro, P. Martins, et al., High-speed tracking with kernelized correlation filters. IEEE Transactions on Pattern Analysis and Machine Intelligence. **37**(3), 583–596 (2015)
50. J. F. Henrigues, R. Caseiro, P. Martins, et al., in *Computer Vision – ECCV 2012*. Exploiting the circulant structure of tracking-by-detection with kernels (Springer Berlin Heidelberg, 2012), pp. 702–715. https://doi.org/10.1007/978-3-642-33765-9_50
51. K. Zhang, L. Zhang, Q. Liu, et al., in *European conference on computer vision (ECCV)*. Fast visual tracking via dense spatiotemporal context learning, (2014), pp. 127–141. https://doi.org/10.1007/978-3-319-10602-1_9
52. D. Wang, H. Lu, M. Yang, Robust visual tracking via least soft-threshold squares. IEEE Trans. Circ. Syst. Video Technol. **26**(9), 1709–1721 (2016)
53. Y. Wu, J. Lim, M. H. Yang, *Online object tracking: A benchmark* (IEEE, 2013). https://doi.org/10.1109/cvpr.2013.312
54. Y. Wu, J. Lim, M. H. Yang, Object tracking benchmark. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1834–1848 (2015)

Publisher's Note