Aerospace Science and Technology ••• (••••) •••–•••

Contents lists available at ScienceDirect

Aerospace Science and Technology

www.elsevier.com/locate/aescte

# Edge-based target detection for unmanned aerial vehicles using competitive Bird Swarm Algorithm

Xiaohua Wang, Yimin Deng, Haibin Duan

*State Key Laboratory of Virtual Reality Technology and Systems, School of Automation Science and Electrical Engineering, Beihang University, Beijing 100083, PR China*

ARTICLE INFO

*Article history:*
Received 1 June 2017
Received in revised form 26 April 2018
Accepted 30 April 2018
Available online xxxx

*Keywords:*
Unmanned aerial vehicle
Target detection
Edge potential function
Bird Swarm Algorithm

ABSTRACT

Target detection for unmanned aerial vehicles is an important issue in autonomous formation flight. In this paper, a novel target detection approach for unmanned aerial vehicle formation is proposed based on edge matching. The windowed edge potential function is utilized to describe the attraction field for similar edges. Afterwards, the edge-based target detection problem can be formulated as an optimization problem. An improved version of the bird swarm algorithm, which is called competitive bird swarm algorithm, is proposed to find the location, rotation angle and scale of a given template on a specific image. A strategy named "disturbing the local optimum" is designed to help the original Bird Swarm Algorithm converge to the global optimal solution faster and more stably. Unmanned aerial vehicles moving in leader-follower pattern, which are called formation flight platforms, are used for our experiments. Images obtained by vision sensors embedded in the leaders are used to verify the effectiveness of the proposed method. The proposed algorithm is tested on both indoor and outdoor images to demonstrate the robustness. Comparative experiments with other state-of-the-art algorithms, including genetic algorithm, particle swarm optimization, artificial bee colony algorithm, pigeon-inspired optimization, and the basic bird swarm algorithm, are also conducted. The results prove the superiority and robustness of the proposed target detection algorithm.



## 1. Introduction

Unmanned Aerial Vehicles (UAVs) are important platforms in both civilian and industrial applications [1], for the advantages of zero casualties, good stealth performance, short operational preparation time, and relatively low life-cycle cost [2]. Multi-UAVs moving in formation have attracted much attention [3–8]. With visual sensors becoming more and more advanced, vision based formation has aroused much interest [9–12]. In leader-follower formation, the leader can ascertain the direction and distance of the following UAVs with information obtained by visual sensors. Therefore, target detection is an important task in vision based autonomous formation flight of UAVs. The aim of this study is to design an adaptive, efficient and robust target detection algorithm that can be applied to vision based UAV formation.

A wide variety of strategies have been established to deal with the target detection problem for UAVs in recent years [13]. A real-time detection algorithm for moving target from UAVs is proposed in [14]. A template matching approach is used in [15] to detect and track the run-way in image sequences. A target detection algorithm based on a visual attention model is given in [16]. A novel bio-inspired model is proposed in [17] via improved artificial bee colony and visual attention. An image registration algorithm is proposed in [18] for moving target detection. In addition, edge or template matching based methods, such as Charmfer matching [19] and Hausdorff distance matching [20], have also been extensively used for target detection [21].

In this paper, an edge matching based method is proposed for UAV target detection, which is much simpler and more efficient compared with feature-based algorithms. The edge of an image is detected and utilized in a matching procedure, which searches for the image patch with the highest similarity to a given edge template. A fast edge detection method based on structured forests is proposed in [22], which can obtain real-time performance and achieve state-of-the-art edge detection results. The edge matching method proposed in [23] utilizes the Canny edge detector. However, the edge map of a high-resolution UAV image extracted by Canny, would be enriched with textures, which will aggravate the computational complexity of edge matching. Therefore, the outstanding edge detection method based on structured forests is utilized to extract edges in this paper. Afterwards, the windowed edge potential function (WEPF) [23] is used to make a description

*E-mail address:* hbduan@buaa.edu.cn (H. Duan).

https://doi.org/10.1016/j.ast.2018.04.047
1270-9638/

of image edge. In terms of matching results and computational complexity, the WEPF based description is superior to traditional edge matching methods [24].

The edge-based matching problem can be considered as an optimization problem which can be resolved with various methods [13,21,23,24]. There are many advantages of bio-inspired optimized algorithms, such as high robustness, good distributed computing mechanisms, and extensive feasibility [25]. Thus a newly proposed bio-inspired optimization algorithm, Bird Swarm Algorithm (BSA) [26], is used for edge matching in aerial images. BSA is a novel optimization algorithm inspired by the social behaviors and social interactions in bird swarms. The searching space of BSA can be very large because of the high resolution of aerial images. In addition, the background of the scenes can be cluttered and complex, which means that there are many local optima in the edge matching problem. Thus, the basic BSA may be trapped into local optima or converge slowly. A strategy called "disturbing the local optimum" is designed and integrated into the basic BSA to overcome these shortcomings of it. The proposed BAS with the "disturbing the local optimum" strategy is called Competitive Bird Swarm Algorithm (CBSA). The "disturbing the local optimum" in CBSA can automatically check whether CBSA is trapped into local optima and add some disturbance to the local optimal solution if necessary. Therefore, the exploration ability of it is enhanced and the diversity of the swarm is improved. Furthermore, premature converge can be avoided.

The proposed target detection method is tested on aerial images obtained by the visual sensor embedded in UAVs. Experimental results demonstrate that the proposed edge matching algorithm can deal with the target detection problem for UAVs effectively. Furthermore, the performance of CBSA is compared with that of BSA, which demonstrates that the proposed strategy can improve the searching and converging ability of CBSA. Moreover, comparative experiments on CBSA and several state-of-the-art bio-inspired algorithms, including the particle swarm optimization (PSO) [27], artificial bee colony (ABC) algorithm [21], Genetic Algorithm (GA) [28], and Pigeon-inspired Optimization (PIO) algorithm [13] are conducted to demonstrate the advantage of the CBSA approach.

The remainder of this paper is organized as follows. In Section 2, the edge detection approach and the principle of WEPF is introduced. In Section 3, the BSA algorithm is presented. The "disturbing the local optimum" strategy is proposed in Section 4, where the detailed implementation procedure of the proposed edge-based target detection algorithm is provided. In Section 5, a series of experimental results are given to demonstrate the effectiveness of the proposed approach, followed by the concluding remarks given in Section 6.

## 2. Edge detection and the WEPF model

### 2.1. Edge detection method using structured forests

Edges generally exhibit patterns of local structure, such as straight lines or T-junctions [29]. The problem of predicting local edge masks can be formulated as a structured learning framework applied to random decision forests [22,30].

1) *Structured random forests*

Structured labels are utilized to determine the splitting function at each branch in the tree. For a $d \times d$ image patch, the annotation of it can be either a segmentation mask $y \in Y = \mathbb{Z}^{d\times d}$ or a binary edge map $y' \in Y' = \{0,1\}^{d\times d}$. Both representations are utilized in this approach. All the structured labels $y$ at a given node are robustly mapped to a discrete set of labels $c \in C$, $C = \{1,\ldots,k\}$. Similar structured labels are assigned to the same discrete label. Standard information gain measures can be evaluated on the discrete space. A mapping from $Y$ to an intermediate space $Z$ is defined to measure similarity over $Y$ and calculate information gain. $m$ dimensions of $Z$ are sampled to reduce dimensionality, and a reduced mapping $\Pi_\phi : Y \to Z$ is obtained. To further reduce the dimensionality, principal component analysis (PCA) [31] is utilized. Then a straightforward map from $Z$ to $C$ is utilized to obtain the discrete labels. PCA quantization can be used to obtain the discrete label set $C$. The top $\log_2(k)$ PCA dimensions can be used to quantize a discrete label $c$ as the assignment of $z$. In this paper we set $m = 256$ and $k = 2$. Each forest predicts a patch of edge pixel labels that are aggregated across the image to compute the final edge map.

2) *Input features*

In this paper, $32 \times 32$ image patches are used to predict $16 \times 16$ structured segmentation tasks. Two types of features are used: pixel lookups and pairwise differences. $x \in \mathbb{R}^{32\times32\times K}$ is the feature vector, where $K$ is the number of channels. Three channels in the CIE-LUV color space together with the normalized gradient magnitude at the original scale and half resolution scale are used. Four orientation channels are derived from the gradient magnitude channels. Thus the input feature has 13 channels. Each channel is blurred with a radius 2 triangle filter and down sampled by a factor of 2. A large triangle blur is used on each channel (8 pixel radius), and each channel is down sampled to a resolution of $5 \times 5$. Then candidate pairs are sampled and pairwise differences are computed. Thus the total dimension of a candidate feature is 7228.

3) *Mapping function*

A mapping $\Pi_\phi : Y \to Z$ is defined to train decision trees. $y(j)$ for $1 \le j \le 256$ denote the $j$th pixel of mask $y$. $z = \prod(y)$ is a large binary vector that encodes $[y(j_1) = y(j_2)]$ for each unique pair $j_1 \ne j_2$.

4) *Ensemble model*

The outputs of multiple trees in the random forests are combined to achieve robust results. The corresponding edge map $y'$ is stored at each leaf node together with the learned mask $y$. Multiple overlapping edge maps $y' \in Y'$ can be averaged to obtain a soft edge response.

5) *Multiscale detection and edge sharpening*

The structured edge detector can be implemented on a multiscale version to enhance the performance of it. Three versions of resolution (1/2, 1, and 2) are computed and the results of the three edge maps are averaged after resizing to the original image dimensions. The edge maps can be sharpened optionally using local image color and depth values, with which the edge maps are better aligned to the image data. Additionally, the edge values vary from 0 to 1, which is continuous-valued. However, binary-valued edge information should be imported to the EPF computation. Thus, edge values lower than 1/3 of the maximum value in the corresponding edge image are set as 0, while others are set as 1 in this paper.

### 2.2. The principle of EPF

The image edges are considered as charged elements in EPF, which can generate an attraction field over object with similar edges. The concept of EPF is derived from the physics of electricity, simulating the electric potential generated by the electrostatic field. It is utilized in this paper to model the potential generated by edge structures of images. An edge template of a particular target is attracted by a set of equivalent charged edge points, which maximize the potential in EPF.

In the electricity, a set of point charges in a homogeneous background $Q_i$ generates a potential, whose intensity can be calculated as follows.

$$v(\vec{r}) = \frac{1}{4\pi\varepsilon}\sum_i \frac{Q_i}{|\vec{r}-\vec{r}_i|} \tag{1}$$

where $\vec{r}$ and $\vec{r}_i$ are the observation point and the charge location, respectively. $\varepsilon$ is the electrical permittivity of the medium. Objects with opposite charge in the potential field will be attracted to the field point with the maximum differential potential.

In the edge-based target detection problem, the $i$-th edge point with the coordinate $(x_i, y_i)$ in the image is equivalent to the charge point $Q_{eq}(x_i, y_i)$. The edge potential produced by a set of edge points can be described as follows.

$$\mathrm{EPF}(x, y) = \frac{1}{4\pi\varepsilon_{eq}} \sum_i \frac{Q_{eq(x_i,y_i)}}{\sqrt{(x-x_i)^2 + (y-y_i)^2}} \tag{2}$$

where $\varepsilon_{eq}$ is the equivalent permittivity of image background. The edge points in the template image can be considered as an object in the equivalent edge potential field. Therefore, it would also be attracted to the field point with the maximum differential potential.

Windowed EPF (WEPF) is an improved version of the basic EPF [23], with the similarity measure and computation speed. A window $w$ is defined in WEPF to limit the influence radius of a charged element to the potential. Thus WEPF can be defined as follows.

$$\mathrm{WEPF}(x, y) = \frac{Q}{4\pi\varepsilon_{eq}} \sum_{(x_i,y_i)\in w} \frac{1}{\sqrt{(x-x_i)^2 + (y-y_i)^2}} \tag{3}$$

where $w$ is the window defined to center on $(x, y)$ in the image, the window size is set as $9 \times 9$ in this paper. $(x_i, y_i) \in w$ are edge points in the calculating window. All the edge points are modeled with an equal charge $Q$ for simplification. WEPF improves both the robustness and speed of edge matching process. In this paper, the edge maps are binarized in this paper, thus $Q = 1$. When $x = x_i$ and $y = y_i$, the value of $\frac{1}{\sqrt{(x-x_i)^2+(y-y_i)^2}}$ is set as 10 compulsorily. The equivalent permittivity of image background is set as $\varepsilon_{eq} = 0.05$ in this paper.

## 3. The BSA algorithm

BSA is a novel optimization algorithm inspired by the swarm intelligence extracted from the social behaviors and social interactions in bird swarms. While foraging, birds frequently raise their heads and scan their surroundings, which is interpreted as vigilance behavior [32]. Studies also suggest that birds on the periphery of a group try to move to the center to protect themselves from being attacked by predators [33]. Additionally, birds may fly to another site for foraging or for escaping from predators [34]. Therefore, in the BSA algorithm three behaviors are mainly extracted, i.e., foraging behavior, vigilance behavior and flight behavior.

### 3.1. Foraging behavior

Each bird searches for food according to its previous best experience and the swarms' previous best experience about food. Thus, the foraging behavior can be elaborated as:

$$X_{i,j}^{t+1} = X_{i,j}^t + S_1 \times \left(P_{i,j} - X_{i,j}^t\right) \times \mathrm{rand}(0,1) + S_2 \times \left(G_j - X_{i,j}^t\right) \times \mathrm{rand}(0,1) \tag{4}$$

where $j \in [1, \ldots, D]$, $D$ is the dimension of the problem. $S_1$ and $S_2$ are cognitive and social accelerated coefficient, respectively. $P_{i,j}$ is the best position of the $i$-th bird and $G_j$ is the global position, which is shared among the swarm.

### 3.2. Vigilance behavior

Each bird tries to move towards the center of the swarm while keeping vigilance. Birds among the swarm compete for the position in the center, where is much safer than the periphery of the swarm. This behavior can be simply modeled as:

$$X_{i,j}^{t+1} = X_{i,j}^t + A_1 \times \left(mean_j - X_{i,j}^t\right) \times \mathrm{rand}(0,1) + A_2 \times \left(P_{k,j} - X_{i,j}^t\right) \times \mathrm{rand}(-1,1) \tag{5}$$

$$A_1 = a_1 \times \exp\left(-\frac{pFit_i}{sumfit + \varepsilon} \times N\right) \tag{6}$$

$$A_1 = a_2 \times \exp\left(\left(\frac{pFit_i - pFit_k}{|pFit_k - pFit_i| + \xi}\right)\frac{N \times pFit_i}{sumfit + \xi}\right) \tag{7}$$

where $k \in \{1, 2, \ldots, N\}, k \neq i$, $N$ is the number of birds in the swarm. $a_1, a_2 \in [1, 2]$ are coefficient which can be adjusted, $mean_j$ is the $j$-th element of the average position of the whole swarm. $pFit_i$ is the best fitness of the $i$-th bird and $sumfit$ is the sum of the swarms' best fitness value. $\xi$ is the smallest constant in the computer.

### 3.3. Flight behavior

Birds fly to other sites to forage again in responding to predation or merely for foraging. A bird can be a producer or a scrounger. Behaviors can be formulated as follows for producers and scroungers, respectively.

$$X_{i,j}^{t+1} = x_{i,j}^t + \mathrm{rand}n(0,1) \times X_{i,j}^t \tag{8}$$

$$X_{i,j}^{t+1} = x_{i,j}^t + \mathrm{rand}(0,1) \times FL \times \left(X_{k,j}^t - X_{i,j}^t\right) \tag{9}$$

where $\mathrm{rand}n(0,1)$ is a Gaussian distributed random number with $\mu = 0, \sigma = 1$. $FL \in [0, 2]$ means that the scrounger will follow the producer to search for food. Each bird is assumed to fly to another place every $FQ$ unit interval.

## 4. Target detection using CBSA

The edge detection algorithm and the principle of EPF have been clarified and the basic BSA has also been discussed. However, the edge matching process is complex and challenging. Although the BSA algorithm is novel and superior, there are still some shortcomings when utilized to the edge matching problem. Thus an improving strategy called "disturbing the local optimum" is introduced to the original BSA to help it jump out of local optima. In this section, the proposed CBSA algorithm will be discussed and the framework of the edge-based target detection algorithm for UAVs using CBSA is given in detail.

### 4.1. "Disturbing the local optimum"

In the strategy of "disturbing the local optimum", an effective method that identifies premature stagnation is embedded to the basic BSA. The strategy can automatically check whether BSA is trapped into the local optimum. Once the premature stagnation happens, a randomized solution is used as a substitute for current optimum to change the current searching locus.

When the optimal solution of CBSA has no change during continuous $MAX$ iterations, the algorithm is assumed to have the potential of stagnation. A counter $CN$ is used to record the number of continuous iterations in which CBSA obtains the same optimal solution. If in current iteration CBSA obtains the same optimal solution to that in the last iteration, $CN = CN + 1$. Otherwise, $CN = 0$. When $CN = MAX$, a random value is used to replace a randomly

selected dimension of current optimal solution, which can be described as follows.

$$G' = (G_{i1}, G_{i2}, \ldots, G_{iD}) \tag{10}$$

where $G_{ik} \in [\min_k, \max_k], k \in [1, D]$ is randomly chosen. Compute the fitness value of $G'$ and compare it with the best fitness value of the swarm. If the fitness value of $G'$ is larger, current best solution $G$ is replaced with $G'$. This strategy can make a fine tuning for the global optimal solution with a random disturbance and help the algorithm jump out of the local optimal solution. The number of invalid iterations can be effectively reduced with the stagnation judgment and random disturbance. Thus the algorithm can converge to the global optimum faster and more stably.

### 4.2. Problem formulation

A binary edge image of a target is given as a template, which is utilized to detect the specific target in the tested aerial image. WEPF is utilized to calculate the edge potential for the edge map of the test image. Afterwards, the CBSA algorithm is utilized to find a patch in the test image that can arouse the highest attraction of the template. The matching process can be specifically described as finding the optimal translation parameters $(t_u, t_v)$, rotation angle $\theta$ and scaling factor $s$ of the template to make a match with the test image. A candidate solution of the problem can be defined as $\omega = [t_u, t_v, \theta, s]$. Afterwards, the matching function is defined as:

$$f(\omega) = \frac{1}{N^{(\omega)}} \sum_{n^{(\omega)}=1}^{N^{(\omega)}} \left\{ \mathrm{WEPF}\left(x_n^{(\omega)}, y_n^{(\omega)}\right) \right\} \tag{11}$$

where $n^{(\omega)}$ is the $n$-th pixel of the edge template, which is transformed with $\omega$. $N^{(\omega)}$ denotes the number of edge pixels in the transformed template. $(x_n^{(\omega)}, y_n^{(\omega)})$ represents the corresponding coordinate of a candidate patch in the test image. Therefore, $\mathrm{WEPF}(x_n^{(\omega)}, y_n^{(\omega)}) = \frac{Q}{4\pi\varepsilon_{eq}} \sum_{(x_i,y_i)\in w} \frac{1}{\sqrt{(x_n^{(\omega)}-x_i)^2+(y_n^{(\omega)}-y_i)^2}}$, where $w$ is the window defined to center on $(x_n^{(\omega)}, y_n^{(\omega)})$ in the test image. $(x_i, y_i) \in w$ are the edge points in the calculating window. Therefore, the matching function can be rewrited as:

$$f(\omega) = \frac{1}{N^{(\omega)}} \times \sum_{n^{(\omega)}=1}^{N^{(\omega)}} \left\{ \frac{Q}{4\pi\varepsilon_{eq}} \sum_{(x_i,y_i)\in w} \frac{1}{\sqrt{(x_n^{(\omega)}-x_i)^2+(y_n^{(\omega)}-y_i)^2}} \right\} \tag{12}$$

when $x_n^{(\omega)} = x_i$ and $y_n^{(\omega)} = y_i$, $\frac{1}{\sqrt{(x_n^{(\omega)}-x_i)^2+(y_n^{(\omega)}-y_i)^2}}$ is set as 10 compulsorily. All edge points are modeled with an equal charge $Q$ for simplification. In this paper, the edge maps are binarized, thus $Q = 1$. The value of $f(\omega)$ is maximized when the edge template with the certain transformation is matched accurately in the test image.

### 4.3. Implementation of CBSA for target detection

The implementation of the proposed edge-based target detection for UAVs using CBSA can be described as follows.

**Step 1)** Edge detection. The test image is obtained with the visual sensor embedded in UAV. The edge map of the test image is calculated.

**Step 2)** WEPF calculation. Calculate WEPF of the test image with the windowed edge potential field function model given in (3) based on the edge map obtained in **Step 1)**.

**Step 3)** Initialize parameters. Initialize parameters of the CBSA algorithm, including the number of birds $N$, the maximum number of iteration $M$, the frequency of birds' flight behaviors $FQ$, the probability of foraging for food $p$, five constant parameters $S_1, S_2, a_1, a_2$ and $FL$, and $MAX$ in the "disturbing the local optimum" strategy. Additionally, initialize the search space of the CBSA algorithm and initialize the position of birds. Set the number of iteration $Nc = 0$ and $CN = 0$.

**Step 4)** Evaluate the fitness of birds. Compute the fitness value of each bird using (12). Store the fitness of each bird and the global best fitness. Store the positions of the birds.

**Step 5)** Choose an operation. Set $Nc = Nc + 1$. Check whether the iteration number is larger than $M$. If so, go to **Step 10)**. Otherwise, check if $Nc > 1$ and current global best fitness equals to that of last iteration. If so, set $CN = CN + 1$, then go to **Step 9)** when $CN > MAX$ or go to **Step 5)** when $CN \leq MAX$. Otherwise, set $CN = 0$ and check if $Nc\%FQ = 0$. If so, go to **Step 8)**. Otherwise, go to Step 6) when $\mathrm{rand}(0, 1) < p$ or go to **Step 7)** when $\mathrm{rand}(0, 1) \geq p$.

**Step 6)** Conduct the foraging behavior. Use (4) to update each the position bird. Evaluate the fitness of birds. If the new solutions are better than their previous ones, update them. Go to **Step 5)**.

**Step 7)** Conduct the vigilance behavior. Use (5)–(7) to make the birds move to the center while competing with each other. Evaluate the fitness of birds. If the new solutions are better than their previous ones, update them. Go to **Step 5)**.

**Step 8)** Conduct the flight behavior. Use (8) or (9) to update each the position bird. Evaluate the fitness of birds. If the new solutions are better than their previous ones, update them. Go to **Step 5)**.

**Step 9)** Conduct the "disturbing the local optimum" strategy. Use (10) to make a disturbance of current best position and set $CN = 0$. If the fitness of $G'$ is better than that of current best fitness in the swarm, update the best position. Go to **Step 5)**.

**Step 10)** Output the optimal solution and its fitness value. Use the translation, rotation and scaling parameters given by CBSA to transform the edge template. Mark the transformed edge template on the original test image to make the result visualized.

The flow chart of CBSA for edge-based target detection is given in Fig. 1.

## 5. Simulation results and analysis

In our experiments, UAVs moving in the leader-follower formation are used as the platforms. In this formation structure, one of the UAVs is designated as the leader, with others treated as followers. The images obtained by the visual sensor embedded in the leader are utilized to test the feasibility and robustness of the proposed method in the edge-based target detection task. The information obtained by target detection will be used to make relative pose estimations among UAVs in further research.

The performance of CBSA is compared with those of the basic BSA and other state-of-the-art evolutionary algorithms, including the PSO [27], ABC [21], GA [28], and PIO [13]. The methods are tested on several images in different scenes, and the comparative results are given. The experiments are conducted on a PC with Intel Core i7, 3.6-GHz CPU, 4-GB memory, and 64-bit Windows 7.

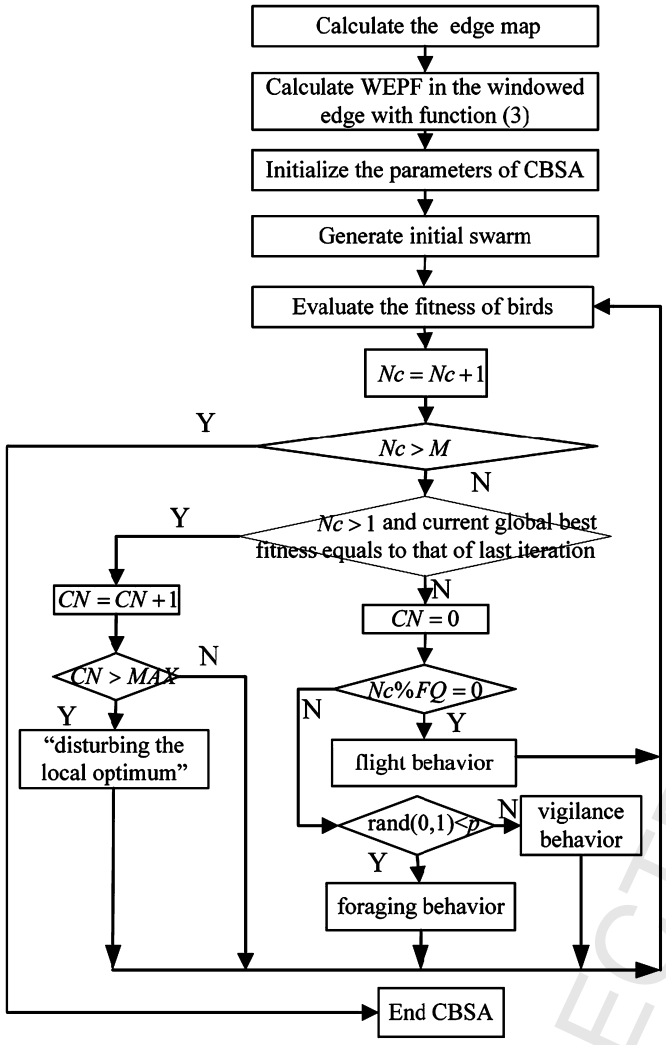


**Fig. 1.** Flow chart of edge-based target detection for UAVs using CBSA.

**Table 1**
Parameters of CBSA.

| Symbol | Quantity | Value |
|---|---|---|
| $N$ | Number of individuals | 200 |
| $M$ | Maximum number of iteration | 100 |
| $FQ$ | Frequency of birds' flight behaviors | 2 |
| $p$ | Probability of foraging for food | 0.8+ 0.2*rand |
| $S_1$ | Constant parameter 1 in (4) | 1.5 |
| $S_2$ | Constant parameter 2 in (4) | 1.5 |
| $a_1$ | Constant parameter in (6) | 1 |
| $a_2$ | Constant parameter in (7) | 1 |
| $FL$ | Constant parameter in (9) | 0.5 + 0.4*rand |
| $MAX$ | Parameter in "disturbing the local optimum" strategy | 5 |

The initial parameters of CBSA are selected based on tests and practical experience. The number of individuals and the maximum number of iteration for all the 6 algorithms are set as the same to make fair comparisons. The searching spaces of the algorithms are determined according to test images. The range of translation equals to the size of the test image. The value of the rotation varies from 0° to 360°. The scaling parameter varies from 0.8 to 1.5. Parameters of BSA and CBSA are set the same, except for MAX. Parameters of the CBSA method are presented in Table 1.

**Table 2**
Comparative results of 6 methods for Experiment I.

| | Algorithm | Best solution | Best fitness |
|---|---|---|---|
| Case I | GA | [204, 681, 102, 1.1] | 9.5570 |
| | PSO | [274, 683, 161, 1.0] | 14.7404 |
| | ABC | [270, 679, 305, 9] | 12.3714 |
| | PIO | [274, 683, 161, 1.0] | 14.7404 |
| | BSA | [214, 96, 32, 9] | 9.6743 |
| | **CBSA** | **[274, 690, 315, 1.0]** | **33.7330** |
| | **Truth** | **[274, 690, 315, 1.0]** | **33.7330** |
| Case II | GA | [254, 375, 297, 0.9] | 12.5584 |
| | PSO | [251, 288, 233, 0.9] | 15.4586 |
| | ABC | [252, 309, 167, 1.0] | 13.9585 |
| | PIO | [296, 307, 178, 0.9] | 17.9702 |
| | BSA | [248, 311, 46, 1.1] | 13.5968 |
| | **CBSA** | **[252, 308, 315, 1.0]** | **33.3646** |
| | **Truth** | **[252, 308, 315, 1.0]** | **33.3646** |

### 5.1. Experiment I

In this experiment the proposed CBSA algorithm and other five algorithms are tested on two indoor images. The resolution of the original image used in Case I is 1291*964. The high resolution of the image results in a large searching space for the optimization algorithm. The experimental results are given in Fig. 2.

The template used in Experiment I Case I is rotated with 45° in the original scale. In Fig. 2(c), the edge map obtained by the Canny algorithm is cluttered. The edge map given in Fig. 2(d) that obtained by the method introduced in Section 2.1 is much cleaner. Afterwards, the edge map in Fig. 2(d) is binarized and utilized to compute the WEPF. The experimental results given in Fig. 2(f) shows the best matching results of Case I obtained by the aforementioned 6 algorithms among 10 independent runs. The matching results show that only the proposed CBSA algorithm makes the optimal matching, while the other 5 algorithms are all failed. The convergence curves are given in Fig. 2(g). The green line given in Fig. 2(g) is the theoretical best fitness. The legend of it is "Truth". In Case I of Experiment I, CBSA obtains the best fitness value within 100 iterations. The evolution curves of CBSA and BSA show that the BSA algorithm is trapped into local optimum and the converging speed of it is rather slow. The performances of the other four algorithms are also dissatisfactory. The proposed CBSA method fulfilled the target detection task.

The resolution of the original image used in Experiment I, Case II is 1008*765. The target detected in Experiment I Case I is installed on the UAV as a mark. The template used in Experiment I Case II is rotated with 45° in the original scale. The results shown in Fig. 3 indicate that the CBSA method can make good performance for the indoor images. The evolution curves further demonstrated the advantage the CBSA, which detects the target accurately. The best solutions obtained with the 6 methods are given in Table 2, which show that CBSA find the accurate solutions. The row named "Truth" represents the theoretical optimal solution and the corresponding best fitness. Results in Table 2 show that the proposed CBSA algorithm converges to the theoretical optimum.

### 5.2. Experiment II

In Experiment II, the six algorithms are tested on two outdoor images, which are obtained by the UAV visual sensor. The original images used in Cases I and II are obtained with the Mercury camera from Daheng IMAVISION. The resolutions of them are both 1292*964. Two octorotors used as the leader-follower formation flight platform are S1000 from DJI technology Inc. The camera is embedded in the leader UAV. The formation flight platform is

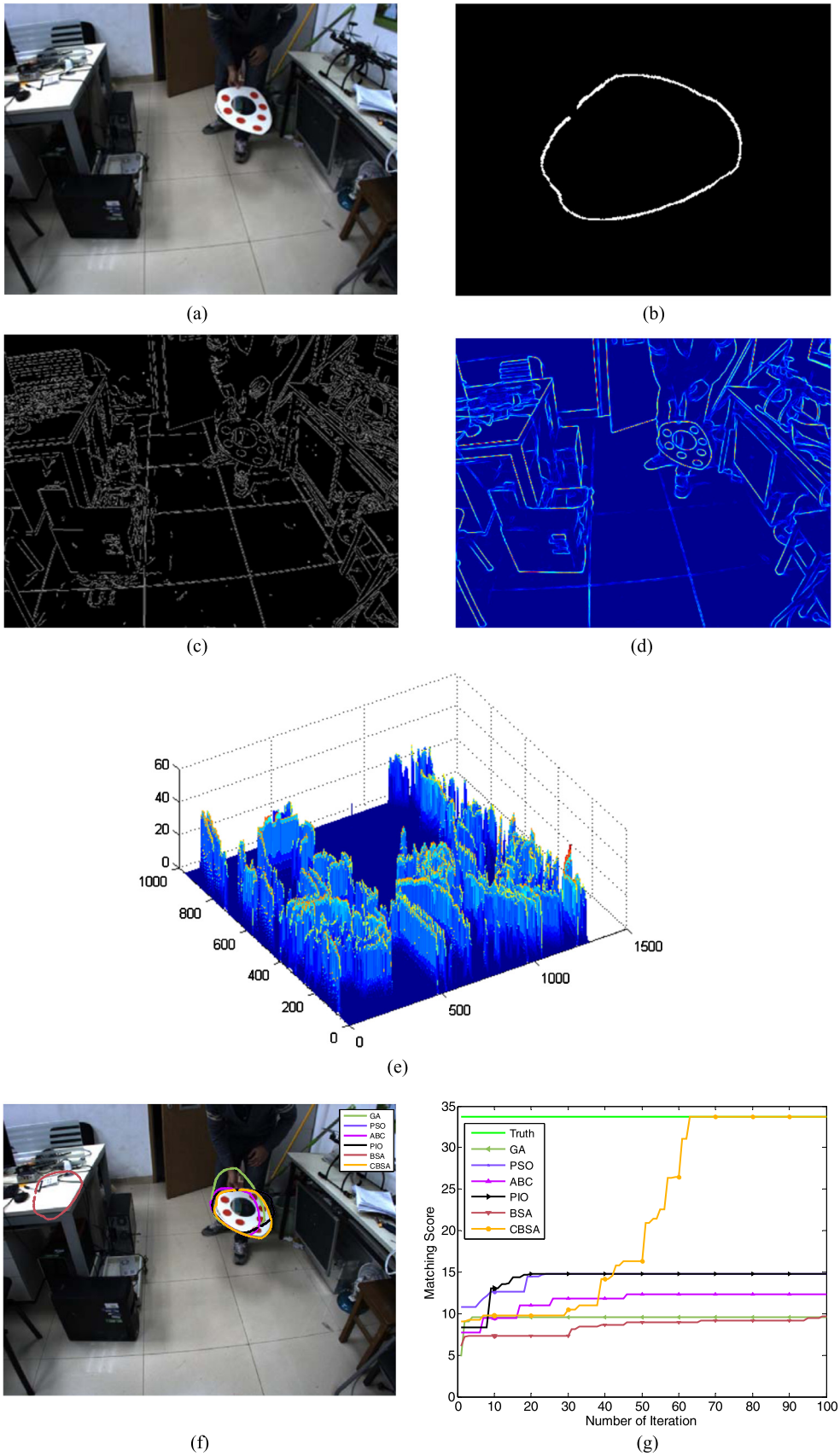


**Fig. 2.** Target recognition results obtained by the six evolutionary algorithms for Experiment I, Case I. (a) Test image, (b) edge template, (c) edge map of the test image extracted of by Canny, (d) edge map extracted by the method introduced in Section 2.1, (e) edge potential distribution of (d), (f) matching results, and (g) evolution curves. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

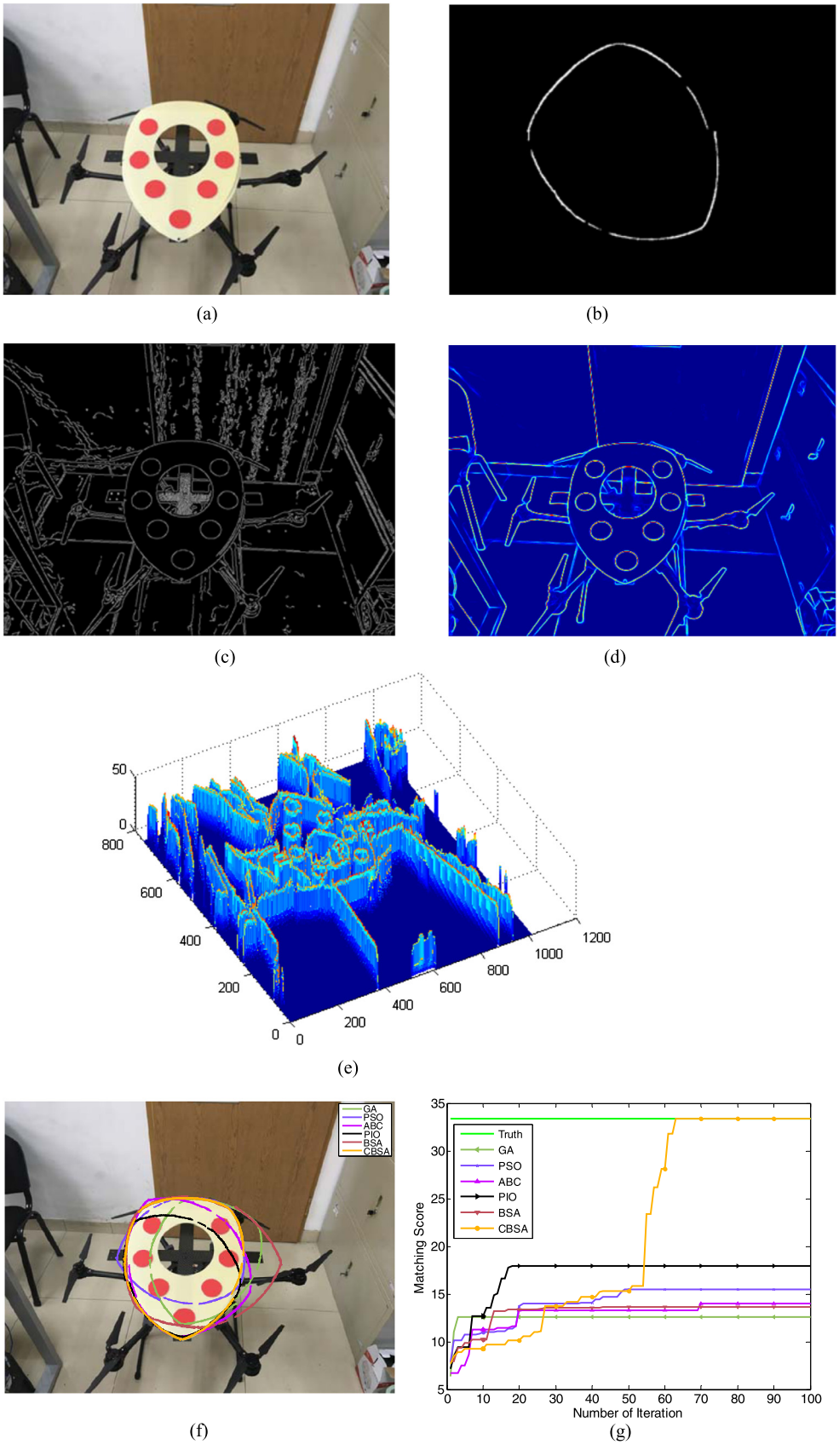


**Fig. 3.** Target recognition results obtained by the six evolutionary algorithms for Experiment I, Case II. (a) Test image, (b) edge template, (c) edge map of the test image extracted by Canny, (d) edge map extracted by the method introduced in Section 2.1, (e) edge potential distribution of (d), (f) matching results, and (g) evolution curves.

**Fig. 4.** Leader-follower formation flight platform used in Experiment II.

shown in Fig. 4. The target is the same to that detected in Experiment I. The experimental results of Cases I and II in Experiment II are given in Figs. 5 and 6.

The same edge template is used in both Cases I and II. To test the robustness of the proposed algorithm, the template size is reduced to 1/1.1 and 1/1.2 of the original size in Cases I and II, respectively. The edges extracted by Canny, given in Fig. 5(c) and Fig. 6(c) are redundant and cluttered because the texture of the background is complex. However, the method introduced in Section 2.1 can restrain the edge of the background well. Only some main edges are obtained, which can decrease the complexity of calculation obviously.

The comparative results of the six algorithms verified the superiority of CBSA. Matching results given in Figs. 5 and 6 show that GA, PSO, and the basic BSA all mistake the characters on the signboard with the target, while CBSA attains a precise matching. Moreover, the target is occluded partly with equipment in the leader UAV in Case II. The CBSA algorithm can also detect the target, which further demonstrates the robustness of the proposed method.

### 5.3. Experiment III

The six algorithms are tested with a different target in Experiment *III*. The resolution of the original image used in Cases I and II in Experiment III is 1024*526. The leader UAV is designed based on the quadrotor X650 pro from XAIRCRAFT Technology Co., Ltd. A Phantom 3 Standard quadrotor from DJI technology Inc is used as the follower UAV, which is also the target to detect. The leader-follower formation flight platform is shown in Fig. 7. The experimental results of Cases I and II are given in Figs. 8 and 9.

In the first case of Experiment III, CBSA finds the global optimum successfully, demonstrating the effectiveness of both the objective function and CBSA. In Case I of Experiment III, the edge template is rotated with 300°. Partial information of edge in the target template is lost. The PSO algorithm roughly detects the target, while other 4 comparative algorithms obtained unsatisfying results as shown in Fig. 8(f). The evolution curves in Fig. 8(g) show that the best fitness value obtained by CBSA is larger than those of the other 5 algorithms. It is obviously that the proposed "disturbing the local optimum" strategy strengthens the searching ability of CBSA.

In Experiment III, Case II, another image is utilized to test the performance of CBSA using an edge template with both rotation and scaling. The edge template is rotated with 100° and the size of it is set as 1/1.2 of the original size. Target detection results given in Fig. 9 have shown that CBSA can still deal with the matching problem well when both the orientation and scale of the edge template are changed. CBSA detects the following UAV accurately.

**Table 3**
Comparative results of 6 methods for Experiment III.

| | Algorithm | Best solution | Best fitness |
|---|---|---|---|
| Case I | GA | [265, 707, 243, 1.4] | 7.6877 |
| | PSO | [352, 775, 241, 1.0] | 16.0348 |
| | ABC | [261, 692, 313, 1.1] | 10.1732 |
| | PIO | [304, 622, 74, 1.0] | 13.0546 |
| | BSA | [273, 703, 197, 0.9] | 11.9890 |
| | **CBSA** | **[352, 784, 60, 1.0]** | **25.0994** |
| | **Truth** | **[352, 784, 60, 1.0]** | **25.0994** |
| Case II | GA | [322, 645, 86, 1.4] | 6.3535 |
| | PSO | [304, 592, 82, 1.0] | 12.4548 |
| | ABC | [271, 577, 0, 1.0] | 13.1675 |
| | PIO | [298, 634, 90, 0.9] | 11.5265 |
| | BSA | [304, 592, 82, 1.0] | 12.4548 |
| | **CBSA** | **[305, 592, 260, 1.2]** | **29.5452** |
| | **Truth** | **[305, 592, 260, 1.2]** | **29.5452** |

The best solutions obtained with the six methods in Experiment III are given in Table 3. In the first case the rotation and scaling parameters obtains by CBSA is 300 and 1.0, respectively. In the second case the rotation and scaling parameters are 100 and 1/1.2, respectively. Thus the best solutions of CBSA are all accurate. In addition, all the 10 evolution curves of CBSA for Experiment III, Case II are given in Fig. 10. In one of the ten runs, CBSA is trapped into a local optimum, while in other 9 runs it all converges to the optimal solution. The green line with legend "Truth" given in Fig. 10(g) is theoretical best fitness. The blue lines represent the 9 run that CBSA converges to the global optimum, while the red line indicates the run that CBSA is trapped into a local optimum.

The "disturbing the local optimum" makes a fine tuning for the global optimal solution with a random disturbance. Therefore, it cannot guarantee that a better solution can be found with the disturbance. In the condition that the algorithm trapped to a local optimum and all the four dimensions of the local optimum are different from that of the global optimum, it will be difficult for CBSA to jump out of the local optimum. The reason is that the "disturbing the local optimum" uses a random value to replace a randomly selected dimension of the current optimal solution. Therefore, only one dimension of the local optimum is modified in an operation. But it indeed enhances the exploration ability of BSA, improves the diversity of the swarm, and increase the probability of obtaining the optimal solution.

## 6. Conclusions

An edge-based target detection algorithm using CBSA is proposed for UAVs in formation flight. A fast edge extraction method based on structured forest is adopted to obtain discriminating edges in aerial images. The concept of WEPF is adopted in the proposed algorithm to build an attraction pattern for image edges to attract the edge template. The original BSA algorithm is improved with a "disturbing the local optimum" strategy, which can improve the global optimizing ability. The experimental results have demonstrated the superiority of CBSA. There is still some further work to do to reduce the computation time consumption. Furthermore, the implementation of the proposed novel approach in embedded processors is also a focus of the future study.

## Conflict of interest statement

None declared.

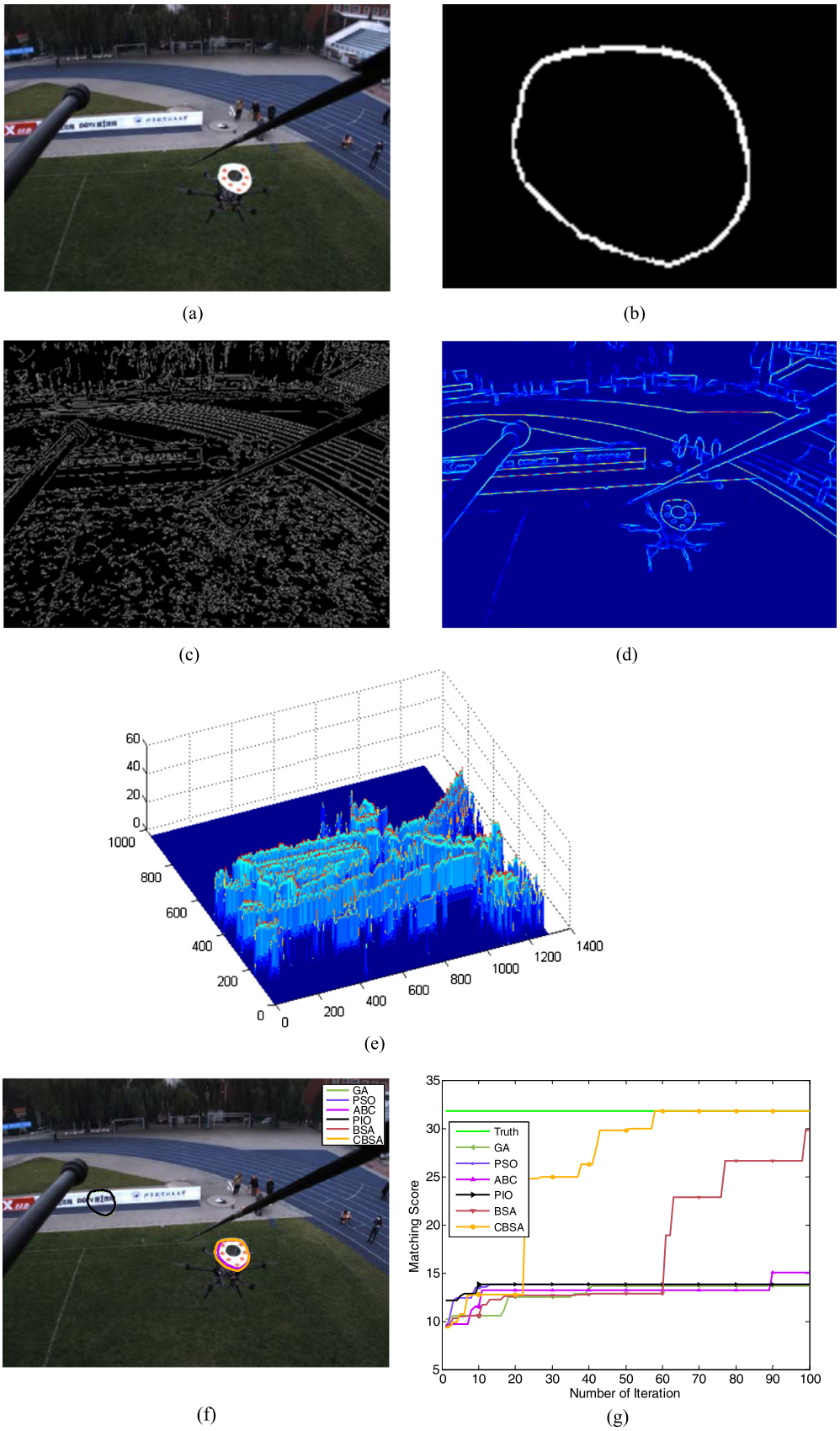


**Fig. 5.** Target recognition results obtained by the six evolutionary algorithms for Experiment II, Case I. (a) Test image, (b) edge template, (c) edge map of the test image extracted by Canny, (d) edge map extracted by the method introduced in Section 2.1, (e) edge potential distribution of (d), (f) matching results, and (g) evolution curves.

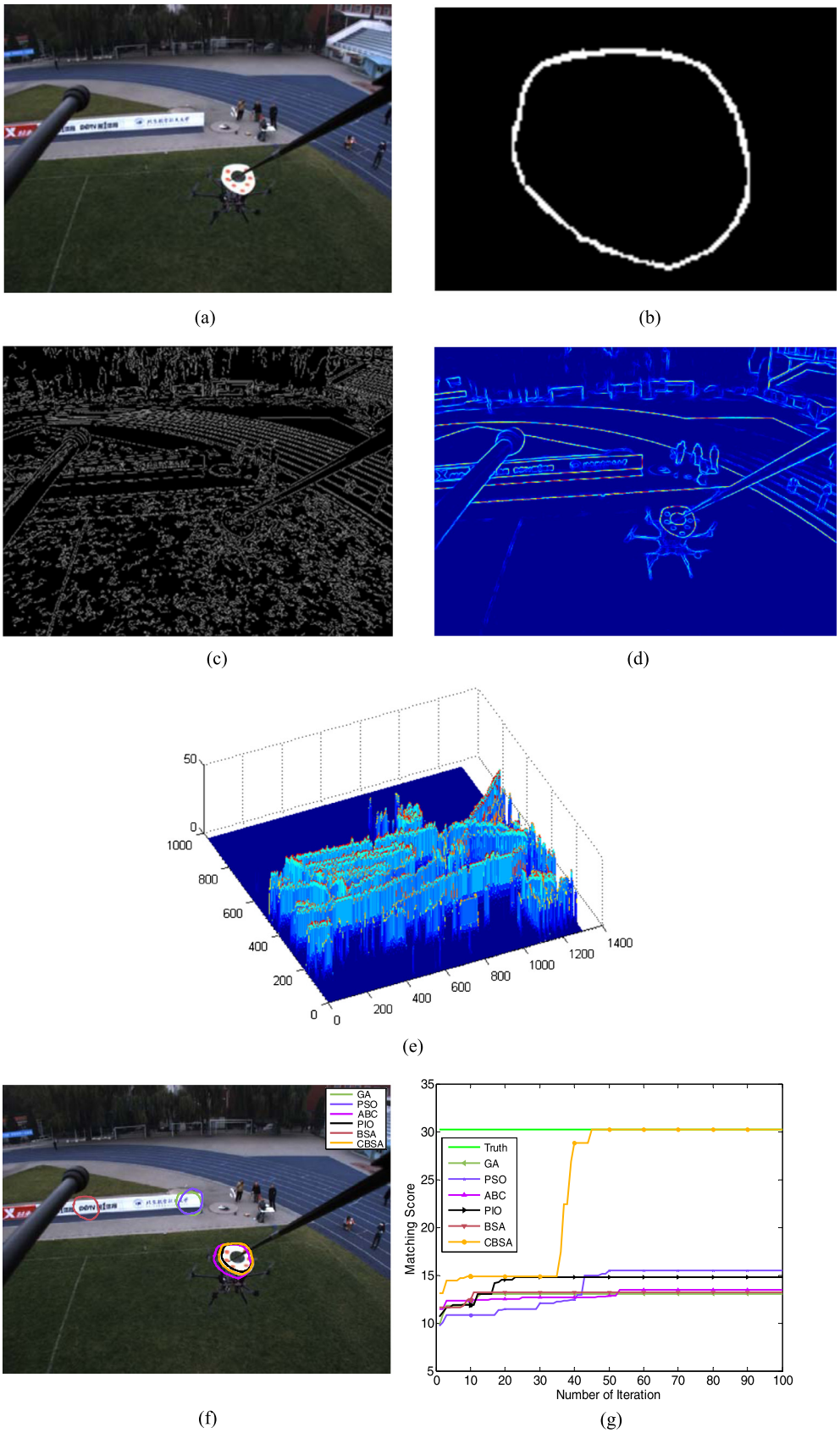


**Fig. 6.** Target recognition results obtained by the six evolutionary algorithms for Experiment II, Case II. (a) Test image, (b) edge template, (c) edge map of the test image extracted by Canny, (d) edge map extracted by the method introduced in Section 2.1, (e) edge potential distribution of (d), (f) matching results, and (g) evolution curves.

**Fig. 7.** Leader-follower formation flight platform used in Experiment III.

## Acknowledgements

This work was partially supported by National Natural Science Foundation of China under grant #61425008, #91648205, #61333004, Aeronautical Foundation of China under grant #2015ZA51013, and the Academic Excellence Foundation of BUAA for PhD Students. The authors would like to thank the editors and reviewers for their critical review of this manuscript.

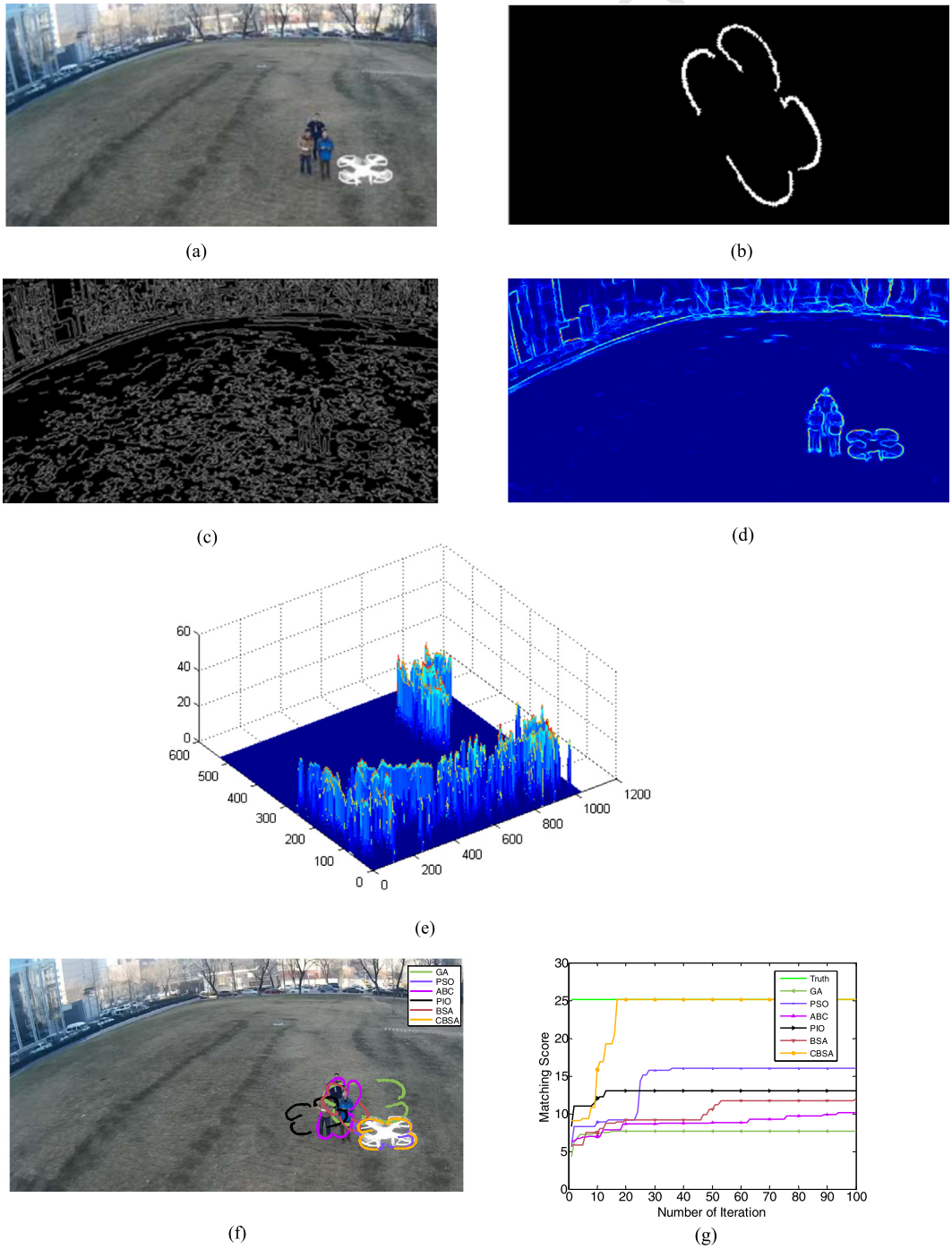


**Fig. 8.** Target recognition results obtained by the six evolutionary algorithms for Experiment III, Case I. (a) Test image, (b) edge template, (c) edge map of the test image extracted by Canny, (d) edge map extracted by the method introduced in Section 2.1, (e) edge potential distribution of (d), (f) matching results, and (g) evolution curves.

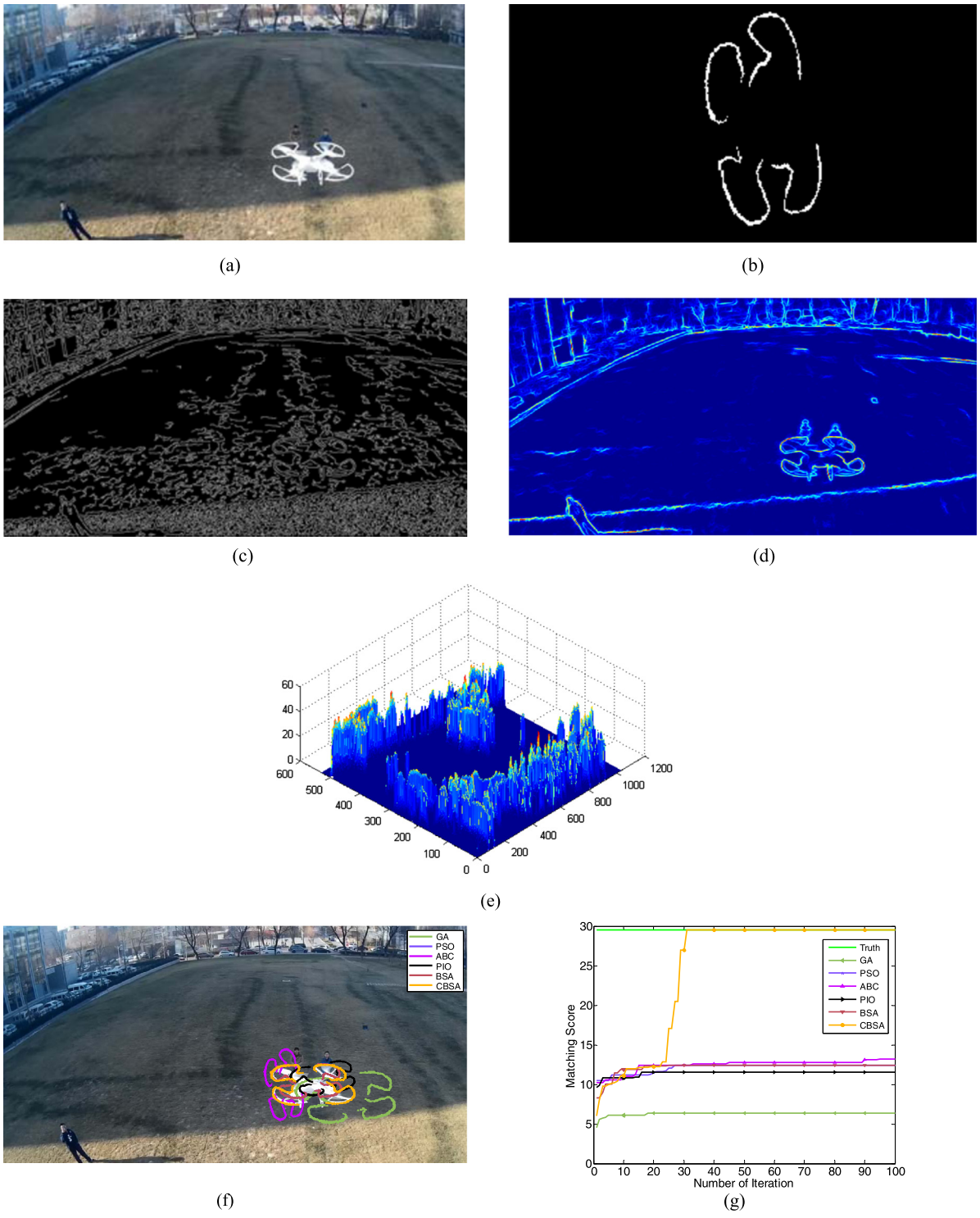


**Fig. 9.** Target recognition results obtained by the six evolutionary algorithms for Experiment III, Case II. (a) Test image, (b) edge template, (c) edge map of the test image extracted by Canny, (d) edge map extracted by the method introduced in Section 2.1, (e) edge potential distribution of (d), (f) matching results, and (g) evolution curves.

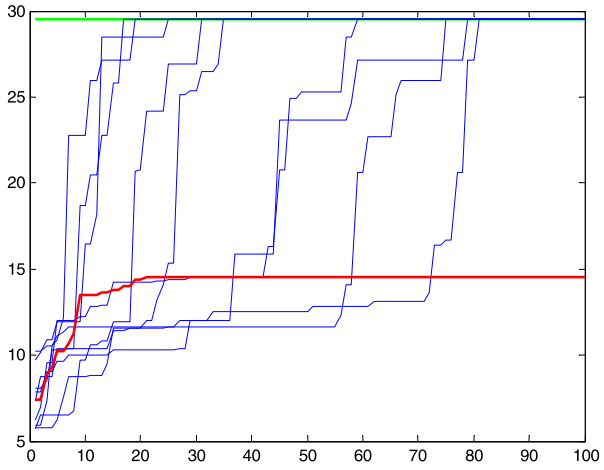


**Fig. 10.** Evolution curves of CBSA for Experiment III, Case II (10 times).

## References

[1] R. Beard, T. McLain, Multiple UAV cooperative search under collision avoidance and limited range communication constraints, in: Proceedings of 42nd IEEE Conference on Decision and Control, vol. 1, December 2003, pp. 25–30.
[2] F. Lin, K. Lum, B. Chen, T. Lee, Development of a vision-based ground target detection and tracking system for a small unmanned helicopter, Sci. China, Ser. F, Inf. Sci. 52 (11) (2009) 2201–2215.
[3] H. Duan, Y. Zhang, S. Liu, Multiple UAVs/UGVs heterogeneous coordinated technique based on Receding Horizon Control (RHC) and velocity vector control, Sci. China, Technol. Sci. 54 (4) (2011) 869–876.
[4] X. Zhang, H. Duan, Y. Yu, Receding horizon control for multi-UAVs close formation control based on differential evolution, Sci. China, Ser. F, Inf. Sci. 53 (2) (2010) 223–235.
[5] H. Duan, S. Liu, Nonlinear dual-mode receding horizon control for multiple UAVs formation flight based on chaotic particle swarm optimization, IET Control Theory Appl. 4 (11) (2010) 2565–2578.
[6] P. Sarunic, R. Evans, Hierarchical model predictive control of UAVs performing multitarget–multisensor tracking, IEEE Trans. Aerosp. Electron. Syst. 50 (3) (2014) 2253–2268.
[7] H. Duan, Q. Luo, Y. Yu, Trophallaxis network control approach to formation flight of multiple unmanned aerial vehicles, Sci. China, Technol. Sci. 56 (5) (2013) 1066–1074.
[8] H. Duan, Q. Luo, Y. Shi, G. Ma, Hybrid particle swarm optimization and genetic algorithm for multi-UAV formation reconfiguration, IEEE Comput. Intell. Mag. 8 (3) (2013) 6–27.
[9] A. Das, R. Fierro, V. Kumar, J. Ostrowski, J. Spletzer, C. Taylor, A vision-based formation control framework, IEEE Trans. Robot. Autom. 18 (5) (2002) 813–825.
[10] G. Mariottini, F. Morbidi, D. Prattichizzo, N. Valk, N. Michael, G. Pappas, K. Daniilidis, Vision-based localization for leader-follower formation control, IEEE Trans. Robot. 25 (6) (2009) 1431–1438.
[11] P. Vela, A. Betser, J. Malcolm, A. Tannenbaum, Vision-based range regulation of a leader-follower formation, IEEE Trans. Control Syst. Technol. 17 (2) (2009) 442–448.
[12] S. Lange, N. Sünderhauf, P.A. Protzel, A vision based onboard approach for landing and position control of an autonomous multirotor UAV in GPS-denied environments, in: Proceedings of International Conference on Advanced Robotics, June 2009, pp. 1–6.
[13] C. Li, H. Duan, Target detection approach for UAVs via improved pigeon-inspired optimization and edge potential function, Aerosp. Sci. Technol. 39 (2014) 352–360.
[14] M. Pouzet, P. Bonnin, J. Laneurit, C. Tessier, A robust real-time image algorithm for moving target detection from unmanned aerial vehicles (UAV), in: Proceedings of 11th International Conference on Informatics in Control, Automation and Robotics, vol. 1, IEEE, September 2014, pp. 266–273.
[15] M. Ding, Y. Cao, L. Guo, A method to recognize and track runway in the image sequences based on template matching, in: Proceedings of IEEE 1st International Symposium on Systems and Control in Aerospace and Astronautics, January 2006.
[16] G. Wang, Target detection based on a model of visual attention for UAV, in: Proceedings of IEEE 26th Chinese Control and Decision Conference, May 2014, pp. 3890–3894.
[17] Y. Deng, H. Duan, Biological edge detection for UCAV via improved artificial bee colony and visual attention, Aircr. Eng. Aerosp. Technol., Int. J. 86 (2) (2014) 138–146.
[18] M. Pouzet, P. Bonnin, J. Laneurit, C. Tessier, Moving targets detection from UAV based on a robust real-time image registration algorithm, in: Proceedings of IEEE International Conference on Image Processing, October 2014, pp. 2378–2382.
[19] G. Borgefors, Hierarchical chamfer matching: a parametric edge matching algorithm, IEEE Trans. Pattern Anal. Mach. Intell. 10 (6) (1988) 849–865.
[20] D. Huttenlocher, G. Klanderman, W. Rucklidge, Comparing images using the Hausdorff distance, IEEE Trans. Pattern Anal. Mach. Intell. 15 (9) (1993) 850–863.
[21] C. Xu, H. Duan, Artificial bee colony (ABC) optimized edge potential function (EPF) approach to target recognition for low-altitude aircraft, Pattern Recognit. Lett. 31 (13) (2010) 1759–1772.
[22] P. Dollar, C. Zitnick, Fast edge detection using structured forests, IEEE Trans. Pattern Anal. Mach. Intell. 37 (8) (2015) 1558–1570.
[23] M. Dao, F. De Natale, A. Massa, Edge potential functions (EPF) and genetic algorithms (GA) for edge-based matching of visual objects, IEEE Trans. Multimed. 9 (1) (2007) 120–135.
[24] H. Duan, L. Gan, Elitist chemical reaction optimization for contour-based target recognition in aerial images, IEEE Trans. Geosci. Remote Sens. 53 (5) (2015) 2845–2859.
[25] H. Duan, S. Liu, J. Wu, Novel intelligent water drops optimization approach to single UCAV smooth trajectory planning, Aerosp. Sci. Technol. 13 (8) (2009) 442–449.
[26] X. Meng, X. Gao, L. Lu, Y. Liu, H. Zhang, A new bio-inspired optimisation algorithm: Bird Swarm Algorithm, J. Exp. Theor. Artif. Intell. (2015) 1–15.
[27] R. Poli, J. Kennedy, T. Blackwell, Particle swarm optimization, Swarm Intell. 1 (1) (2007) 33–57.
[28] C. Im, H. Jung, Y. Kim, Hybrid genetic algorithm for electromagnetic topology optimization, IEEE Trans. Magn. 39 (5) (2003) 2163–2169.
[29] P. Dollár, C.L. Zitnick, Structured forests for fast edge detection, in: Proceedings of IEEE International Conference on Computer Vision, December 2013, pp. 1841–1848.
[30] P. Kontschieder, S. Rota Buló, H. Bischof, M. Pelillo, Structured class-labels in random forests for semantic image labelling, in: Proceedings of IEEE International Conference on Computer Vision, November 2011, pp. 2190–2197.
[31] I. Joliffe, Principal Component Analysis, Springer-Verlag, New York, NY, USA, 1986.
[32] T. Anderson, Biology of the Ubiquitous House Sparrow: From Genes to Populations, Oxford University Press, Oxford, 2006.
[33] H. Pulliam, On the advantages of flocking, J. Theor. Biol. 38 (2) (1973) 419–422.
[34] L. Giraldeau, T. Caraco, Social Foraging Theory, Princeton University Press, Princeton, NJ, 2000.