

# Salient object detection using biogeography-based optimization to combine features

Zhicheng Wang<sup>1</sup> · Xiaobei Wu<sup>1</sup>

Published online: 8 January 2016  
© Springer Science+Business Media New York 2016

**Abstract** Salient object detection aims to automatically localize a foreground object with respect to its background in an image. It plays a crucial role in a wide range of computer vision and multimedia applications. In this work, we propose an improved salient object detection method based on biogeography-based optimization, a relatively new bio-inspired metaheuristic algorithm that searches for the global optimum using a migration model. Our approach consists of two steps. In the first step, a set of local (multi-scale contrast), regional (center-surround histogram), and global (color spatial distribution) salient feature maps are extracted and normalized. In the second step, an optimal weight vector for combining these feature maps into one saliency map is determined using biogeography-based optimization and improved variants of this algorithm. As a result, a salient objects were identified and labeled as distinct from the image background. We implemented our method using three biogeography-based optimization variants, and compared our results for three popular databases against two other state-of-the-art approaches. The experimental results demonstrate that our method exhibits refined and consistent detection of salient objects.

**Keywords** Salient object detection · Biogeography-based optimization · Ecogeography-based optimization · Saliency map · Migration model · Metaheuristic algorithm

---

✉ Zhicheng Wang  
zhichengwang@tongji.edu.cn  
Xiaobei Wu  
xb.wu@ymail.com

<sup>1</sup> College of Electronics and Information Engineering, Tongji University, Shanghai 201804, China

## 1 Introduction

Salient object detection (SOD) refers to detecting the location of the foreground object that captures most human perceptual attention in a scene [1, 2]. It aims at tagging both figural and background regions, and has been widely investigated in neurobiology and psychophysics [3]. SOD plays a crucial role in a broad range of computer vision and multimedia applications, involving image compression, image cropping, and image retrieval. It equips applications with a basic technique to access key information in an image. In mobile robotics localization systems, SOD endows mobile robots with the capability to navigate to specific household objects and to interact with people [4]. In intelligent traffic systems, SOD can be used for finding navigational landmarks, as well as distinguishing traffic signs, vehicles, and pedestrians [5]. In medical-imaging applications [6], SOD facilitates the delineation of pathological structures and other regions of particular interest for further segmentation of anatomical images. In surveillance systems, SOD can interpret activities in a scene automatically, and detect unusual events, such as traffic violations or unauthorized entry [7]. However, even now human perception is not been fully understood; thus, detecting salient objects in an image in an efficient and effective way is still a challenge.

From an information-processing point of view, the existing methods for modeling visual saliency can be classified into two categories: bottom-up methods [5] and top-down methods [8]. In bottom-up methods, the differences between pixels or pitches on the low-level features, such as color (intensity) and edge (texture) are calculated, and saliency maps are extracted by assigning pixels that are more “different” from their surrounding areas with higher saliency values. In top-down methods, high-level knowledge about what types of objects are to be detected is used to process

saliency information in a task-dependent manner. Although the bottom-up methods work well for low-level saliency, they are neither sufficient nor necessary for images where salient regions are unique and related to the human perception [9]. In contrast, the top-down methods often require prior knowledge of the visual system. Because both methods have their disadvantages, it is likely that through integrating lower level features and higher level prior assumptions, salient objects will be detected more effectively and efficiently.

In a recent advance, Liu et al. [10] developed a SOD method that combines bottom-up and top-down approaches. In this method, the SOD model is formulated as a binary labeling problem, while generic salient objects are calculated by combining high-level concepts with a set of novel local (multi-scale contrast), regional (center-surround histogram), and global (color spatial distribution) salient features with a conditional random field from maximum likelihood estimation criteria. However, this model uses only a single common linear weight vector based on a training set to combine the feature maps for all test images. Thus, it loses generalizations and its performance degrades dramatically, when it is applied across multiple types of images, e.g., input images with high variations with respect to training images. In addition, although maximum likelihood estimation is a well-known parameter estimation technique, it is very sensitive to the choice of initial values and does not always converge.

Biogeography-based optimization (BBO) [11] is a relatively new bio-inspired and population-based metaheuristic algorithm for global optimization based on mathematical models of biogeography. In BBO, each individual solution to the problem is modeled as a “habitat” or “island” with a given habitat suitability index (HSI). The algorithm evolves a population of solutions by continuously migrating features between them, until a global optimum or an acceptable near-optimal solution is reached. According to such migration models, high HSI solutions tend to share their features with low HSI solutions, while low HSI solutions are likely to accept many new features from high HSI solutions. BBO has proven itself as a competitive method to many other well-known evolutionary algorithms on a wide set of benchmarks and practical problems [11–19].

In this paper, we develop a novel method to tag salient objects from the image background. Our method consists of two steps. In the first step, multi-scale contrast, center-surround histogram, and color spatial distribution feature maps are extracted and normalized based on the Liu et al. model [10]. In the second step, a BBO [11], including its later improvements [20, 21] are used to determine an optimal weight vector to combine these features into one saliency map to label the salient object from the image background. To verify the efficacy of the method, we carried

out experiments on three well-known public data sets. Its performance is evaluated in terms of precision, recall, the F-measure, area under curve (AUC), computation time, and peak memory use. We compared our method with the other two state-of-the-arts techniques: the original approach described in Liu et al. [10] and a revised method from Singh et al. [22]. Our experimental results demonstrate that our method has a significant performance advantage over these other methods.

In general, the important contributions of our new method are:

- To the best of our knowledge, it is the first time that a BBO metaheuristic algorithm has been successfully applied for SOD.
- Our method outputs uniformly highlighted salient regions with well-defined boundaries.
- Our method is robust and computationally efficient.

The remainder of this paper is organized as follows. Section 2 surveys related work on SOD. Section 3 describes the considered optimization problem in SOD. Section 4 introduces the basic BBO and its variants used for SOD in this work. Section 5 evaluates the proposed method on three public datasets with computational experiments. Section 6 presents the conclusion and further work perspectives.

## 2 Related work

In the literature, the bottom-up methods comprise the greatest proportion of publications. These methods only consider low-level features based on image distinctiveness, making them intuitional and fast.

In 1998, based on the behavior and neuronal architecture of the early primate visual system, Itti et al. [5] proposed a seminal framework for SOD, where multi-scale image features were combined into a single topographical saliency map, and attended locations were selected using a dynamical neural network to decrease saliency. Cheng et al. [23] proposed a regional contrast-based saliency extraction approach, named the histogram-based contrast method, to evaluate global contrast differences and spatial coherence simultaneously. However, this approach is targeted towards natural scenes, and is suboptimal for extracting saliency of highly textured scenes. Rahtu et al. [24] proposed a simple method for detecting visually salient areas, using searching image segments and sliding windows. It employs an integral histogram approach to efficiently evaluate semi-local intensity histograms, estimating the distributions of objects and surroundings.

Harel et al. [25] presented a biologically plausible, bottom-up visual saliency model, which uses novel concepts from graph theory to concentrate mass on activation maps,

forming activation maps from raw features. These maps highlight conspicuity. Hou et al. [26] proposed a fast front-end method that simulates the behavior of a pre-attentive visual search to construct a saliency map. The spectral residual of a corresponding image in the spectral domain is extracted and then transformed to obtain the saliency map. These maps suggest positions of proto-objects. Achanta et al. [27] presented a frequency-tuned approach to compute saliency in images. They used low-level features of color and luminance to estimate center-surround contrast. This approach is able to output full-resolution saliency maps, with well-defined boundaries to salient objects.

Based on contrast analysis, Ma et al. [28] presented a feasible algorithm to extract attention areas from images. They describe in abstract how to include such information into the saliency computation. They also present an image attention analysis framework that simulates two types of human cognition processes, and three levels of attention, including attended view, attended areas, and attended points.

In comparison, the top-down SOD methods require a priori knowledge to achieve high-quality saliency detection; hence, they are often slower than bottom-up methods.

Navalpakkam et al. [29] proposed a method that defines complex targets and distracting objects as a conjunction of different features across multiple-feature dimensions. It uses top-down components to accumulate statistical knowledge of the local features of the target, while computing visual saliencies of scene locations for different local visual features at multiple spatial scales using bottom-up components. The model is not only applicable to natural scenes, but also to artificial search arrays. Moreover, it allows realistic observers with different beliefs. Zhang et al. [8] proposed an adaptive Bayesian framework to detect salient objects. This method involves human observation behaviors and scalable subtractive clustering techniques to construct an attention Gaussian-mixture model and background Gaussian-mixture model, respectively. It uses a Bayesian framework to maximize the saliency difference. Goferman et al. [30] proposed a novel algorithm to detect context-aware saliency based on the fundamental idea that the salient object is distinctive to its local and global surroundings. It adopts four principles observed in the psychological literature, i.e., local low-level considerations, global considerations, visual organizational rules, and high-level factors.

Shen [9] proposed a unified model based on low rank matrix recovery, where a color image is represented as a low-rank matrix plus sparse noise in a learned feature space. In this method, low-level features and higher level guidance are used to detect salient objects from color images. Thus, the higher level guidance knowledge is integrated as pixel-wise priors in the method. Following a decision-theoretic formulation for saliency, previously developed for top-down processing, Gao et al. [31] proposed a bottom-up

visual saliency detector that combines top-down discriminant principles with bottom-up saliency selection to detect salient objects. In [22], Singh et al. proposed a new objective function to obtain a weight vector for combining low-level features to generate salient objects. It used a constrained particle swarm optimization (C-PSO) algorithm to optimize the vector, and employed an adaptive thresholding strategy that first uses the Canny edge detector (an edge detection operator) to generate edge and silhouette information. It uses this information for classifying pixels. However, it was found that both the C-PSO and the adaptive thresholding strategy were computationally expensive, and the C-PSO can be easily trapped in local optima.

### 3 Weight optimization in salient object detection

In this paper, we consider three key features for detecting salient object(s) in a color image.

- $f_1$ : Multi-scale contrast, which is a widely used local feature simulating human visual receptive fields for attention, because contrast is the most important factor that influences human visual perception [10, 23, 32].
- $f_2$ : Center-surround histogram, which is an important regional salient feature simulating the way the human vision system extracts edges, because a salient object has a large center-surround histogram distance [10].
- $f_3$ : Color spatial distribution, which is a well-known global feature describing the spatial distribution of specific colors, because the more widely a color is distributed in the image, the less likely it is occurs in salient objects [10].

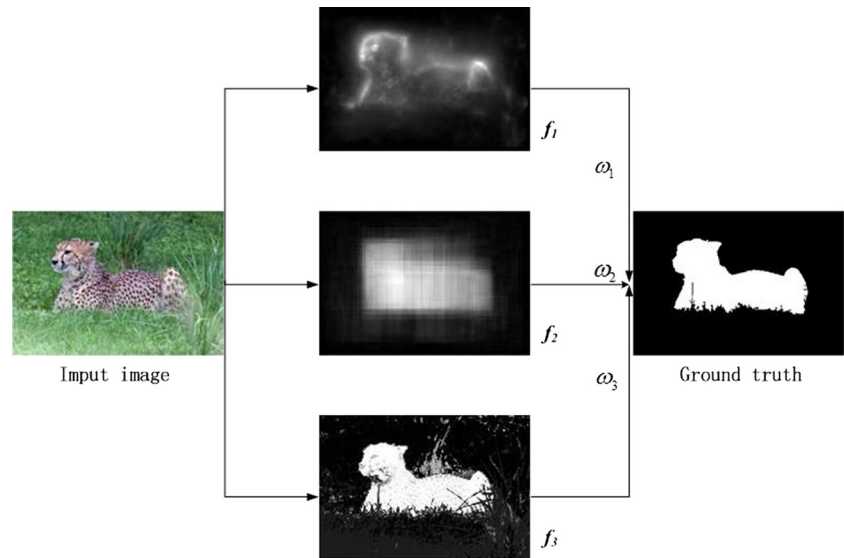
There are many ways to combine these features to obtain saliency results for images. A simple way is to give equal weight to all three features. Here, we solve an optimization problem to combine them in the most optimal way to detect salient objects from the background [22]. In a pre-processing step, the three feature maps ( $f_1, f_2, f_3$ ) of a given image are extracted and normalized according to the model in Liu et al. [10]. As shown in Fig. 1, let  $\vec{\omega} = \{\omega_1 \omega_2 \omega_3\}$  denote a weight vector to combine the three feature maps ( $f_1, f_2, f_3$ ) to tag a saliency result that is as close to the ground truth image as possible. The saliency value of every pixel  $p$  for this saliency map  $S$  is calculated as:

$$S(p, \vec{\omega}) = \sum_{k=1}^3 (\omega_k \times f_k(p)), \quad (1)$$

where  $\sum_{k=1}^3 \omega_k = 1$ .

The saliency map  $S$  is normalized between [0, 1], such that 0 represents black and 1 represents white. We recount

**Fig. 1** A sample input image, showing maps  $f_1$ ,  $f_2$ , and  $f_3$  and its ground truth image



the saliency value of every pixel  $p$  for the normalized saliency map  $S_N$  as follows:

$$S_N(p, \vec{\omega}) = \frac{S(p, \vec{\omega}) - \min_{p \in P} S(p, \vec{\omega})}{\max_{p \in P} S(p, \vec{\omega}) - \min_{p \in P} S(p, \vec{\omega})}, \quad (2)$$

where  $\min_{p \in P} S(p, \vec{\omega})$  denotes the minimum value among the set  $P$  of all pixels, and  $\max_{p \in P} S(p, \vec{\omega})$  denotes the maximum value among set  $P$ .

Using an iterative approach to find the desired optimal weight vector, a fitness function is needed to assess the candidate weight vectors at each iteration. For optimally highlighting salient objects, the saliency values of attention pixels should be maximized, i.e., they should approach 1 (white). In contrast, the saliency values of background pixels should be minimized, i.e., they should approach 0 (black). Hence, we used a fitness function similar to that used by Singh et al. [22], but instead we chose a fixed thresholding strategy that is more computationally efficient than the originally adaptive strategy. In our approach, the threshold is set to the half of the maximum value of the normalized saliency map  $S_N$  for classifying each pixel  $p$  as an attention pixel or a background pixel:

$$\tau(\vec{\omega}) = \frac{\max_{p \in P} S_N(p, \vec{\omega})}{2} \quad (3)$$

Thus, the binarized attention mask  $A$  of salient object is given by:

$$A(p, \vec{\omega}) = \begin{cases} 1, & S_N(p, \vec{\omega}) \geq \tau(\vec{\omega}) \\ 0, & S_N(p, \vec{\omega}) < \tau(\vec{\omega}) \end{cases}, \quad (4)$$

where 1 denotes attention pixels and 0 denotes background pixels.

Because the saliency contribution from attention and background pixels should be maximized and minimized

simultaneously, the fitness function to assess the candidate weight vector is given by:

$$Fitness(\vec{\omega}) = \sum_{p \in P_{att}} (1 - S_N(p, \vec{\omega})) + \sum_{p \in P_{bkg}} S_N(p, \vec{\omega}), \quad (5)$$

where  $P_{att}$  and  $P_{bkg}$  denoting the set of attention pixels and the set of background pixels, are defined as:

$$\begin{aligned} P_{att}(\vec{\omega}) &= \{p : p \in P \wedge A(p, \vec{\omega}) = 1\} \\ P_{bkg}(\vec{\omega}) &= \{p : p \in P \wedge A(p, \vec{\omega}) = 0\} \end{aligned} \quad (6)$$

Clearly, our aim is to minimize the fitness function value, and thereby the optimization problem, to find an optimal weight vector  $\vec{\omega}^*$ , such that:

$$\vec{\omega}^* = \min_{\vec{\omega} \in W} Fitness(\vec{\omega}), \quad (7)$$

where  $W$  denotes the set of three-dimensional real-value vectors satisfying  $\sum_{k=1}^3 \omega_k = 1$ .

## 4 Biogeography-based optimization for salient object detection

In this section, we introduce basic BBO and its improved variants for solving the SOD optimization problem.

### 4.1 Basic biogeography-based optimization

In BBO [11], each solution is modeled as a habitat, while each solution component is modeled as a habitat feature, with a suitability index variable (SIV). The algorithm maintains a population of habitats, ranked by their HSI or fitness values. A higher (lower) HSI value signifies that the habitat has more (less) energetic species, better (worse) living conditions, and more (less) active interaction with neighboring

habitats. We compute for each habitat  $H_i$ , an immigration rate  $\lambda_i$ , and an emigration rate  $\mu_i$ , which are functions of its HSI value. High HSI habitats tend to share their features with low HSI habitats, and low HSI habitats are likely to accept new features from high HSI habitats.

Suppose habitats are sorted in increasing order of their fitness values, then the immigration and emigration rates of the  $i$ th habitat can be calculated as:

$$\lambda_i = I \left( \frac{i}{n} \right), \tag{8}$$

$$\mu_i = E \left( 1 - \frac{i}{n} \right), \tag{9}$$

where  $I$  is the maximum possible immigration rate (occurring when the habitat has no species) and  $E$  is the maximum possible emigration rate (occurring when the habitat has maximum biodiversity). The linear relationship between the HSI of a habitat and its migration rate is shown in Fig. 2.

In BBO, there are two main probabilistic operators: migration, which allows information sharing between candidate solutions; and mutation, which allows candidate solutions to exchange information themselves.

The migration operator works on immigration and emigration rates of the habitats. For each generation, each SIV of each habitat  $H_i$  has a probability  $\lambda_i$  to be immigrated; once selected, this SIV is exchanged with the corresponding SIV of the emigrating habitat  $H_j$ , selected with a probability proportional to  $\mu_j$ .

For the mutation operator, we define a probability  $P_i$  for each habitat  $H_i$  with respect to its immigration rate  $\lambda_i$  and

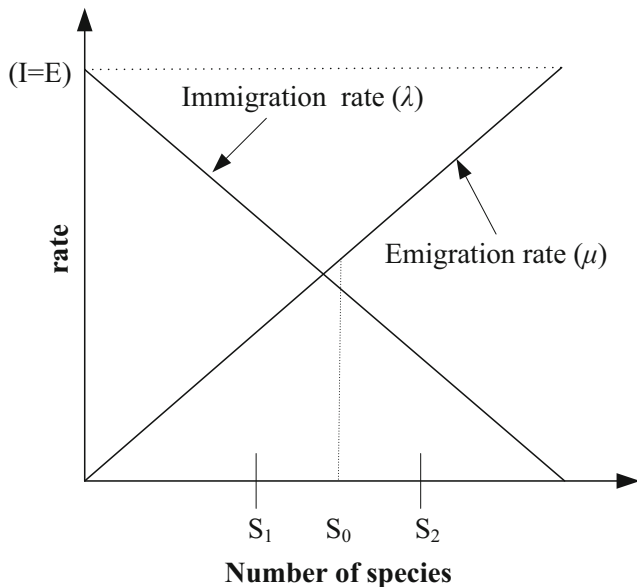


Fig. 2 The relationship between HSI and migration rates

emigration rate  $\mu_i$ , and then compute a mutation rate  $\pi_i$  for  $H_i$ , as follows:

$$\pi_i = \pi_{\max} \left( 1 - \frac{P_i}{P_{\max}} \right), \tag{10}$$

where  $\pi_{\max}$  is the maximum mutation rate given by the user, and  $P_{\max}$  is the maximum habitat probability for the population.

Algorithm 1 presents the framework of the basic BBO for optimizing the weight vector used to combine map features for SOD, where function  $rand()$  produces a random number uniformly distributed in  $[0, 1]$ .

**Algorithm 1** Outline of the basic biogeography-based optimization algorithm

- 1: Randomly initialize a population of  $n$  habitats (weight vectors) for the problem;
- 2: **while** stop criterion is not satisfied **do**
- 3:     **for**  $i = 1$  to  $n$  **do**
- 4:         Evaluate the HSI of  $H_i$ , and calculate its  $\lambda_i$ ,  $\mu_i$ , and  $\pi_i$ ;
- 5:     **end for**
- 6:     **for**  $i = 1$  to  $n$  **do**
- 7:         **for**  $d = 1$  to 3 **do**
- 8:             **if**  $rand() < \lambda_i$  **then** //migration
- 9:                 Select a habitat  $H_j$  with probability  $\propto \mu_j$ ;
- 10:                 Replace the  $d$ th SIV (weight) of  $H_i$  with the corresponding one for  $H_j$ ;
- 11:             **end if**
- 12:         **end for**
- 13:     **end for**
- 14:     **for**  $i = 1$  to  $n$  **do**
- 15:         **for**  $d = 1$  to 3 **do**
- 16:             **if**  $rand() < \pi_i$  **then** //mutation
- 17:                 Replace the  $d$ th SIV of habitat  $H_i$  with a random value within  $[0, 1]$ ;
- 18:             **end if**
- 19:         **end for**
- 20:     **end for**
- 21: **end while**
- 22: **return** the best solution found so far.

In the optimization problem for SOD, each candidate weight vector is represented by a habitat and each vector component is represented by a SIV (Fig. 3). Thus, in each iteration we use (5) to assess each candidate weight vector of the population.

#### 4.2 Localized biogeography-based optimization with ring topology

The basic BBO is based on a global topology, as shown in Fig. 4a. Hence, whenever a habitat is selected for immigration, any other habitat has the chance to share information



SIV	SIV1	SIV2	SIV3
Habitat $H_1$	0.32	0.28	0.40

**Fig. 3** An example of candidate weight vectors in our optimization problem. SIV = suitability index variable

with it. This often causes premature convergence, because habitats may be strongly attracted by a habitat trapped in a local optimum. Zheng et al. [20] suggested using local topology to tackle this issue. One simple local topology is the ring topology, where each habitat is only connected to two other habitats and migration can only occur between neighboring habitats (Fig. 4b). Although this ring topology is quite easy to implement, it improves the search capability very effectively.

The framework of the localized BBO using local ring topology is described in Algorithm 2.

---

**Algorithm 2** The localized biogeography-based optimization algorithm with local ring topology

---

```

1: Randomly initialize a population of  $n$  habitats (weight
   vectors) for the problem;
2: while stop criterion is not satisfied do
3:   for  $i = 1$  to  $n$  do
4:     Evaluate the HSI of  $H_i$ , and calculate its  $\lambda_i$ ,  $\mu_i$ ,
       and  $\pi_i$ ;
5:   end for
6:   for  $i = 1$  to  $n$  do
7:     for  $d = 1$  to 3 do
8:       if  $\text{rand}() < \lambda_i$  then //migration
9:         let  $i1 = (i - 1) \% n$ ,  $i2 = (i + 1) \% n$ ;
10:        if  $\text{rand}() < \mu_{i1}$  then Select  $H_{i1}$  as  $H_j$ ;
11:        else Select  $H_{i2}$  as  $H_j$ ;
12:        end if
13:      end if
14:      Replace the  $d$ th SIV (weight) of  $H_i$  with the
       corresponding one for  $H_j$ ;
15:    end for
16:  end for
17:  for  $i = 1$  to  $n$  do
18:    for  $d = 1$  to 3 do
19:      if  $\text{rand}() < \pi_i$  then //mutation
20:        Replace the  $d$ th SIV of habitat  $H_i$  with a
         random value within  $[0, 1]$ ;
21:      end if
22:    end for
23:  end for
24: end while
25: return the best solution found so far.

```

---

### 4.3 Localized biogeography-based optimization with random topology

There are many other local topologies, such as triangle and square topologies. Another extraordinary topology is the local random topology, where the neighbors of any habitat can be randomly set during the search process. Random topology is much more complex, but it avoids premature convergence better and is more effective than static topologies.

The simplest way to generate a random topology is to randomly select  $K$  (where  $0 < K < n$ , and  $n$  is the size of population) neighbors for each habitat. A more effective way is to make sure that each habitat has  $K$  probable neighbors. Thus, the probability of any two habitats being connected is  $K/(n-1)$ . The parameter  $K$  can be fine tuned to improve algorithm performance for specific problems.

The framework of the localized BBO with random topology is described in Algorithm 3, where *Link* is a matrix giving the neighborhood of the solutions.

---

**Algorithm 3** The localized biogeography-based optimization algorithm with random topology

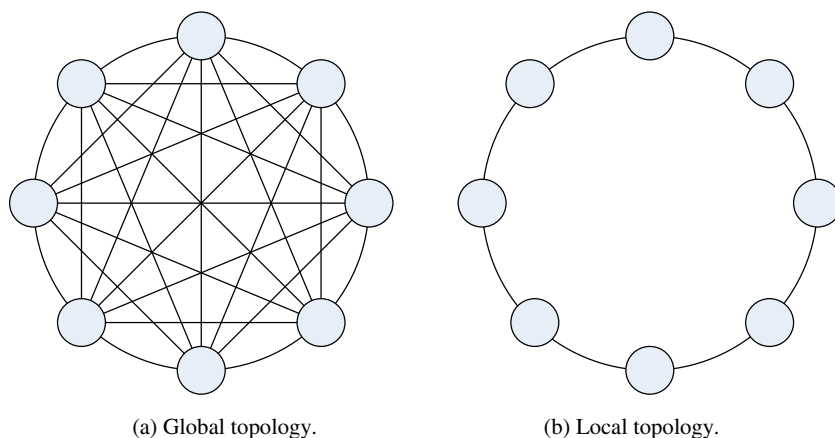
---

```

1: Randomly initialize a population of  $n$  habitats (weight
   vectors) for the problem;
2: Initialize the matrix Link;
3: while stop criterion is not satisfied do
4:   for  $i = 1$  to  $n$  do
5:     Evaluate the HSI of  $H_i$ , and calculate its  $\lambda_i$ ,  $\mu_i$ ,
       and  $\pi_i$ ;
6:   end for
7:   for  $i = 1$  to  $n$  do
8:     for  $d = 1$  to 3 do
9:       if  $\text{rand}() < \lambda_i$  then //migration
10:        Select a habitat  $H_j$  satisfying
          $\text{Link}(i, j) = 1$  with probability  $\propto \mu_i$ ;
11:        Replace the  $d$ th SIV (weight) of  $H_i$  with
         the corresponding one for  $H_j$ ;
12:      end if
13:    end for
14:  end for
15:  for  $i = 1$  to  $n$  do
16:    for  $d = 1$  to 3 do
17:      if  $\text{rand}() < \pi_i$  then //mutation
18:        Replace the  $d$ th SIV of habitat  $H_i$  with a
         random value within  $[0, 1]$ ;
19:      end if
20:    end for
21:  end for
22: end while
23: return the best solution found so far.

```

---

**Fig. 4** Global versus local ring topologies used in models

#### 4.4 Ecogeography-based optimization

Using local topologies, Zheng et al. [21] improved the BBO metaheuristic algorithm by differentiating between migration across neighboring habitats and the migration between non-neighboring habitats. This mimics the principle of immigration and emigration of species between habitats in ecogeographical distributions. This modification is known as ecogeography-based optimization (EBO).

The original BBO had a unique migration operator, while EBO differentiates between global and local migration. Global migration puts more emphasis on exploration, while the local migration tends to perform more exploitation. These are controlled by a linearly decreasing immaturity index  $\eta$  given by:

$$\eta = \eta_{\max} - \frac{t}{t_{\max}}(\eta_{\max} - \eta_{\min}) \quad (11)$$

where  $t$  is the current generation,  $t_{\max}$  is the total number of generations of the algorithm, and  $\eta_{\max}$  and  $\eta_{\min}$  are the upper and lower limits of  $\eta$ .

When a solution is to be immigrated, we generate a random number uniformly distributed between 0 and 1. If this number is smaller than  $\eta$ , then the global migration operator is adopted, otherwise the local migration operator is adopted.

Both global and local migrations operate on each selected  $d_{\text{th}}$  dimension of habitat  $H_i$ , as shown in (12) and (13):

$$H_i(d) = \begin{cases} H_F(d) + \alpha(H_N(d) - H_i(d)), & f(H_F) \leq f(H_N) \\ H_N(d) + \alpha(H_F(d) - H_i(d)), & f(H_F) > f(H_N) \end{cases}, \quad (12)$$

$$H_i(d) = H_i(d) + \alpha(H_N(d) - H_i(d)), \quad (13)$$

where  $\alpha$  is a coefficient ranging from 0 to 1;  $H_N$  and  $H_F$  are neighbor and non-neighbor solutions of  $H_i$ , selected according to the emigration rate  $\mu_i$ ; and  $f$  denotes the fitness function.

The framework of the EBO is described in Algorithm 4.

---

#### Algorithm 4

The ecogeography-based optimization algorithm

---

- 1: Randomly initialize a population of  $n$  habitats (weight vectors) to the problem;
  - 2: **while** stop criterion is not satisfied **do**
  - 3:   Calculate  $\eta$ ;
  - 4:   **for**  $i = 1$  **to**  $n$  **do**
  - 5:     Evaluate the HSI of  $H_i$ , and calculate its  $\lambda_i$ ,  $\mu_i$ , and  $\pi_i$ ;
  - 6:   **end for**
  - 7:   **for**  $i = 1$  **to**  $n$  **do**
  - 8:     Clone  $H_i$  to a new island  $H'_i$ ;
  - 9:     **for**  $d = 1$  **to** 3 **do**
  - 10:      **if**  $\text{rand}() < \lambda_i$  **then**
  - 11:       Select a neighboring habitat  $H_N$  with probability  $\propto \mu_i$ ;
  - 12:       **if**  $\text{rand}() < \eta$  **then**
  - 13:          Select a non-neighboring habitat  $H_F$  with probability  $\propto \mu_i$ ;
  - 14:          **if**  $H_F$  is fitter than  $H_N$  **then**  $H_{i,d} = H_{F,d} + \alpha(H_{N,d} - H_{i,d})$ ;
  - 15:          **else**  $H_{i,d} = H_{N,d} + \alpha(H_{F,d} - H_{i,d})$ ;
  - 16:          **end if**
  - 17:       **else**
  - 18:           $H_{i,d} = H_{i,d} + \alpha(H_{N,d} - H_{i,d})$ ;
  - 19:       **end if**
  - 20:      **end if**
  - 21:     **end for**
  - 22:     **if** at least one dimension of  $H'_i$  is changed **then**
  - 23:       Evaluate the new  $H'_i$ ;
  - 24:       **if**  $H'_i$  is better than  $H_i$  **then** Replace  $H_i$  with  $H'_i$  **end if**;
  - 25:     **end if**
  - 26:   **end for**
  - 27: **end while**
  - 28: **return** the best solution found so far.
-

## 5 Computational experiments

We carried out experiments to evaluate the performance of our method on the following three widely-used image data sets:

- MSRA [10], which includes two image sets. Image set A contains 20843 pictures with their ground truth images manually labeled by three users. Image set B contains 5000 pictures with their ground truth images manually labeled by nine users. The ground truths of MSRA images are presented in a bound box. They use user-averaged results to construct their ground truths.
- iCoSeg [33, 34], is a publicly available co-segmentation database, which contains 643 images along with their pixel ground-truth hand annotations.
- SED [35, 36], which contains 2 subsets of 200 gray level images, along with ground truth segmentation. Each image is segmented by three users. The ground truth is constructed by declaring a pixel as foreground, if it was marked as foreground by at least two users.

We selected 300 images from above three data sets for salient object detection testing. These images differed in attributes, such as category, color, shape, and size. Sample images and their ground truths from each data set are shown in Fig. 5.

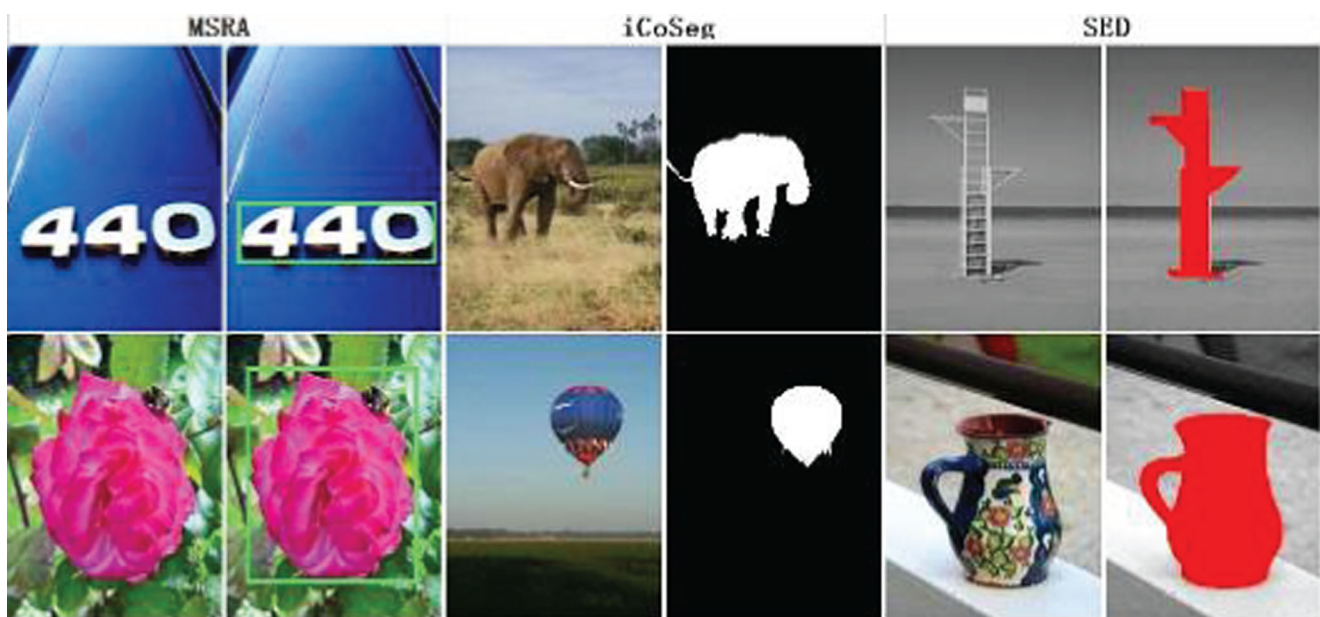
We compared our four BBO algorithms, i.e., basic BBO, localized BBO with ring topology (denoted by LBBO-Ring), localized BBO with random topology (denoted by

LBBO-Random), and EBO, against both the original model from Liu et al. [10] (for which we used a matlab implementation from [37]) and the C-PSO algorithm from Singh et al. [22]. In our algorithms, the fixed threshold  $\tau$  was chosen to be half of the maximum saliency value, the population size was 20, and the iteration number was 10. In LBBO-Random and EBO, the parameter  $K$  used to compute the probability that any two habitats are connected was set to 3. In EBO, the coefficient  $\alpha$  was a random value between (0, 1), and the upper and lower limit of  $\eta$  were 0.7 and 0.4. Parameter settings for the models of Singh et al. [22] and Liu et al. [10] were set, as suggested in their publications. The experiments were conducted on a computer with an Intel Core i5-2400 processor and 4GB memory.

As a first step, we ran each BBO algorithm on each test image five times. Based on the outputs of these algorithms, we divide the test images into three groups:

- Group A: test images where all four BBO algorithm outputs were optimal;
- Group B: test images where there is some probability for BBO algorithms to achieve optimal salient results;
- Group C: test images where none of the four BBO algorithm outputs was optimal.

The values for the first group were averaged over five computational runs. For the other two groups, we carried out 20 additional simulations on each image, and computed their average values for the top 20 of the total 25 runs.



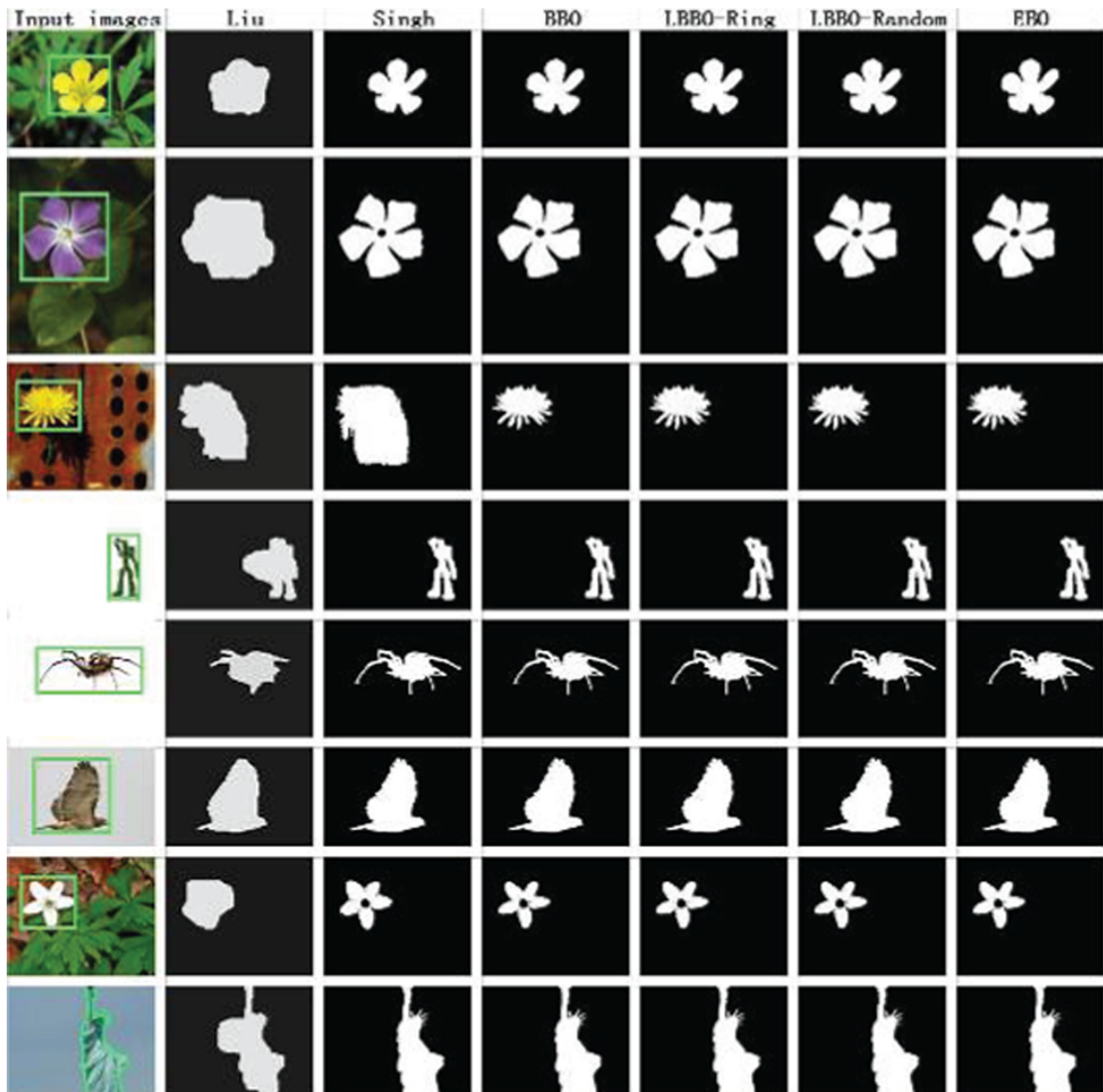
**Fig. 5** Sample images and their ground truths from the three data sets used for salient object detection tests: MSRA, iCoSeg, and SED



### 5.1 Qualitative comparisons

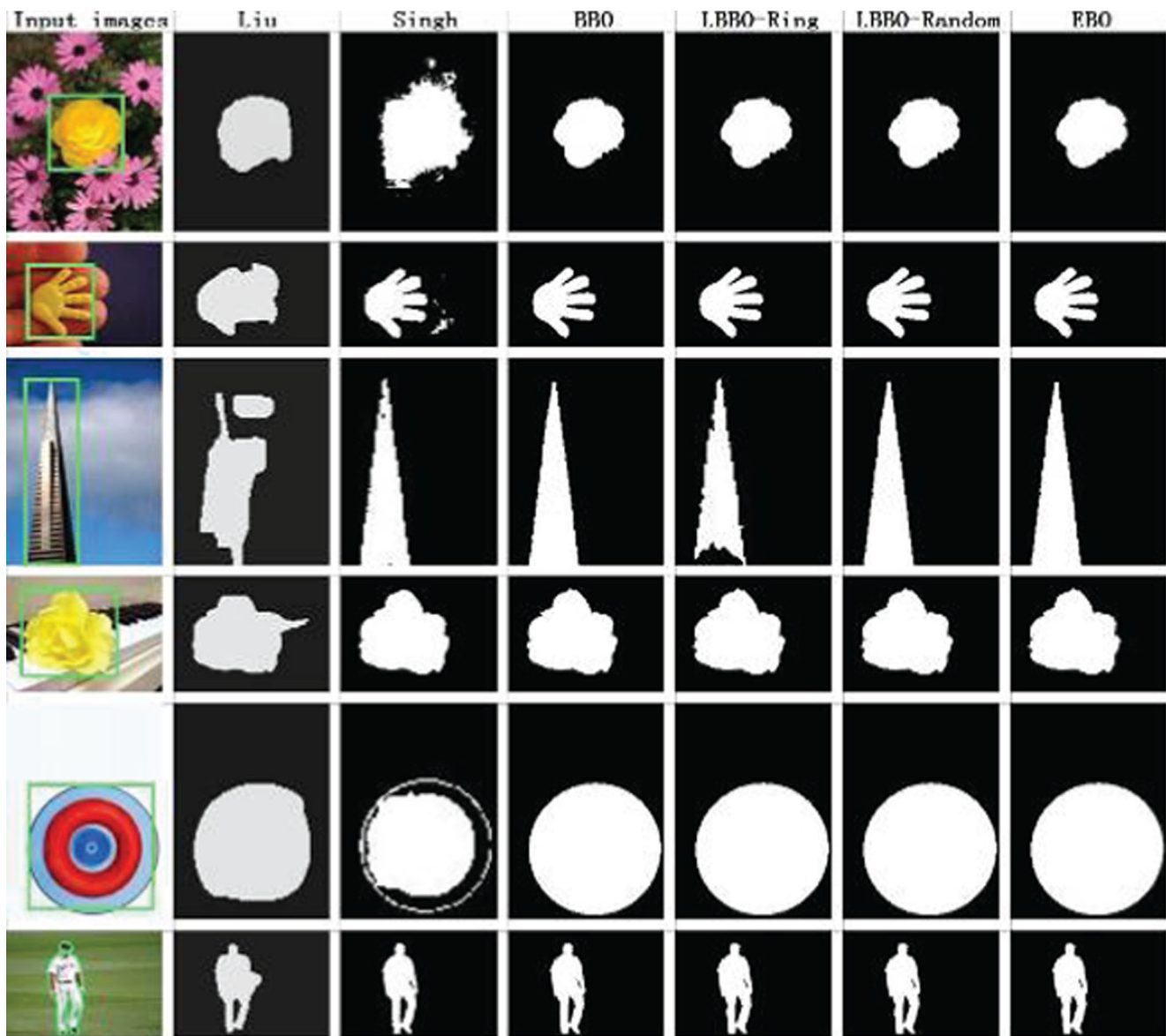
Visual comparisons of the best results for the six different algorithms on images in Groups A–C are shown in Figs. 6, 7 and 8 respectively. The ground truth is marked on the input image. For Group B, we also present the average probability of achieving optimal salient results for various algorithms in Table 1; the best result among these algorithms is marked in bold.

Clearly, Liu et al.'s model localized the objects accurately, but it gave unnecessary information on the salient object, which reduced its performance. Singh et al.'s model obtained saliency results with clear and fine shape information at times, but with a probability that was much smaller than for our method. However, as the complexity of the testing images increased, e.g., the image in the second row of Fig. 7 has only small texture differences between salient object and background, then the Singh et al. model was



**Fig. 6** Visual performance comparison of salient object detection in Group A images using various algorithms. Liu = algorithm from Liu et al. [10]; Singh = algorithm from Singh et al. [22]; BBO = basic

biogeography-based optimization; LBBO-Ring = local BBO with ring topology; LBBO-Random = local BBO with random topology; and EBO = ecogeography-based optimization



**Fig. 7** Visual performance comparison of salient object detection in Group B images using various algorithms. Liu = algorithm from Liu et al. [10]; Singh = algorithm from Singh et al. [22]; BBO = basic

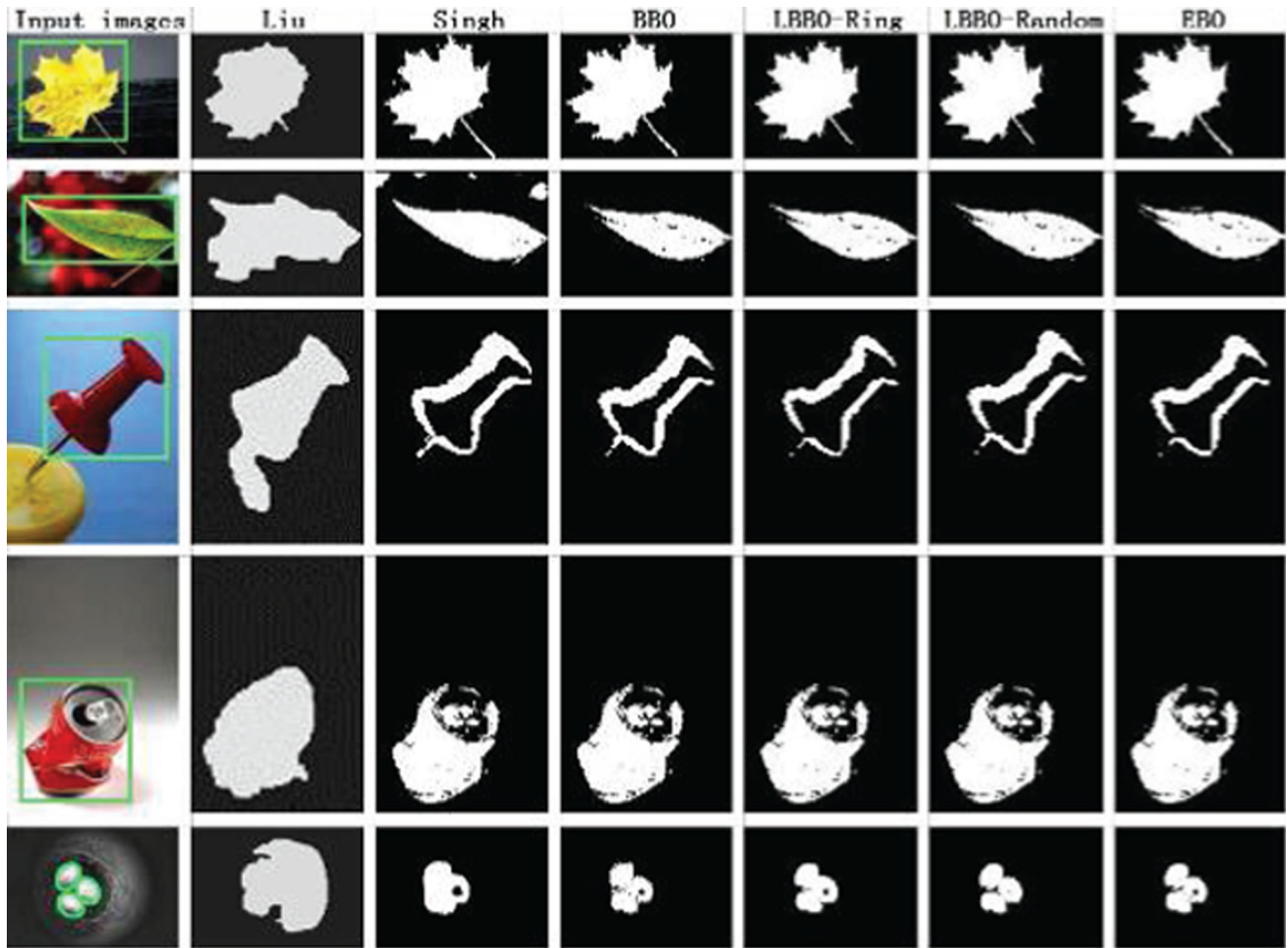
biogeography-based optimization; LBBO-Ring = local BBO with ring topology; LBBO-Random = local BBO with random topology; and EBO = ecogeography-based optimization

more likely to be trapped in local optima. Our method deals better with challenging images, where the background was cluttered. For example, the image in the third row of Fig. 6 as well as images in the first two rows of Fig. 7 show that the other two approaches were distracted by the textures in the background, while our four BBO algorithms successfully output an accurate salient object. In Group C, our method still produced the best results among all algorithms, confirming that our BBO algorithms are more robust to changes in color, object size, and object location in images with different background types.

## 5.2 Quantitative comparisons

We used precision, recall, the F-Measure, the AUC computation time and peak memory usage to quantitatively evaluate and compare the performances of these various algorithms.

Because the ground truths for all MSRA images are shown within a bound box, we surrounded our saliency results with rectangles as well [22]. Based on both the ground truth rectangle (*GTRec*) and the saliency result rectangles (*RSRec*) for each image we calculated *TP* (true



**Fig. 8** Visual performance comparison of salient object detection in Group C images for various algorithms. Liu = algorithm from Liu et al. [10]; Singh = algorithm from Singh et al. [22]; BBO = basic

biogeography-based optimization; LBBO-Ring = local BBO with ring topology; LBBO-Random = local BBO with random topology; EBO = ecogeography-based optimization

positives; the total number of attention pixels that are detected as salient), *FP* (false positives; the total number of background pixels that are detected as salient), *TN* (true negatives; the total number of background pixels that are detected as background) and *FN* (false negatives; the number of attention pixels that are detected as background) values for the various algorithms as follows:

$$TP = \sum_{GTRec(x,y)=1} RSRec(x, y), \tag{14}$$

$$FP = \sum_{GTRec(x,y)=0} RSRec(x, y), \tag{15}$$

$$TN = \sum_{RSRec(x,y)=0} (1 - GTRec(x, y)), \tag{16}$$

$$FN = \sum_{RSRec(x,y)=0} GTRec(x, y), \tag{17}$$

where  $GTRec(x,y)=1$  gives attention pixels in the ground truth rectangle  $GTRec$  while  $RSRec(x,y)=1$  gives attention pixels in the saliency result rectangles  $RSRec$ .

Because the provided ground truths for images from iCoSeg [33] and SED [35] are not defined by a rectangular shape, they can be generated as a binary mask. Thus, we

**Table 1** Probability of achieving optimal salient results in Group B images for various algorithms (Liu = algorithm from Liu et al. [10]; Singh = algorithm from Singh et al. [22]; BBO = basic biogeography-based optimization; LBBO-Ring = local BBO with ring topology; LBBO-Random = local BBO with random topology; and EBO = ecogeography-based optimization)

Index	Liu	Singh	BBO	LBBO-Ring	LBBO-Random	EBO
Probability	0 %	23 %	<b>61 %</b>	47 %	56 %	54 %

Best result is shown in bold

compared the attention mask  $A$  generated by different algorithms with the binary ground truth mask pixel by pixel to obtain  $TP$ ,  $FP$ ,  $TN$  and  $FN$  values for images from these two datasets.

Hence, the precision, recall and F-measure, were calculated as follows:

$$Precision = \frac{TP}{TP + FP}, \quad (18)$$

$$Recall = \frac{TP}{TP + FN}, \quad (19)$$

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall}. \quad (20)$$

We drew a receiver operator characteristic (ROC) curve between the true-positive rate ( $TPR$ ) and the false-positive rate ( $FPR$ ) to obtain the AUC.  $TPR$  and  $FPR$  were calculated using:

$$TPR = \frac{TP}{\sum_{(x,y)} GTRec(x, y)}, \quad (21)$$

$$FPR = \frac{FP}{w \times h - \sum_{(x,y)} GTRec(x, y)}, \quad (22)$$

where  $w$  and  $h$  represent the width and height of the image.

A quantitative comparison of Groups A–C is given in Tables 2, 3 and 4. The best result for each metric among the various algorithms is marked in bold.

We obtained the peak memory use during processing from Windows task manager for each of the six algorithms. These values are listed in Table 5. Because Liu et al.'s model [10], Singh et al.'s model [22], and our method used a threshold to generate the attention mask  $A$  of the salient object from the saliency map, we adjusted this threshold to draw ROC curves. Figure 9 presents ROC curves for  $TPR$  and  $FPR$  for the various algorithms. The AUC values of these algorithms are listed in Table 5; the best result is marked in bold.

For images in Groups A–C, our four BBO algorithms had precision values that were much better than Singh et al. and Liu et al. models. For images in Groups A–C, the Singh et al. model gave the highest recall rates. Our four BBO algorithms had highest F-measures among the six algorithms for images in Groups A–C.

Our four BBO algorithms had least computation time, without requiring more memory for images in Groups A–C. They all ran much faster than the Singh et al. model. The peak memory use for our four BBO algorithms was no larger than either Singh et al. or Liu et al. models.

AUC is a measure used to rank quality. Algorithms covering larger AUC are better in terms of their performance. Our four BBO algorithms had values of AUC that were much better than either the Singh et al. or Liu et al. model.

**Table 2** Quantitative metrics for Group A images (Liu = algorithm from Liu et al. [10]; Singh = algorithm from Singh et al. [22]; BBO = basic biogeography-based optimization; LBBO-Ring = local

BBO with ring topology; LBBO-Random = local BBO with random topology; and EBO = ecogeography-based optimization)

Index	Results	Liu	Singh	BBO	LBBO-Ring	LBBO-Random	EBO
Precision	average	0.843	0.913	<b>0.991</b>	<b>0.991</b>	<b>0.991</b>	<b>0.991</b>
	max	1.000	1.000	1.000	1.000	1.000	1.000
	min	0.535	0.357	0.963	0.963	0.963	0.963
	std	0.204	0.225	0.014	0.014	0.014	0.014
Recall	average	0.896	<b>0.955</b>	0.949	0.949	0.949	0.949
	max	0.961	0.987	1.000	1.000	1.000	1.000
	min	0.659	0.909	0.889	0.889	0.889	0.889
	std	0.098	0.024	0.038	0.038	0.038	0.038
F-measure	average	0.850	0.914	<b>0.969</b>	<b>0.969</b>	<b>0.969</b>	<b>0.969</b>
	max	0.974	0.982	0.982	0.982	0.982	0.982
	min	0.682	0.499	0.941	0.941	0.941	0.941
	std	0.118	0.168	0.016	0.016	0.016	0.016
Time (in sec) per image	average	8.110	16.809	6.944	7.035	<b>6.928</b>	6.988
	max	8.837	46.189	7.785	8.127	7.663	7.858
	min	7.064	8.100	5.928	5.868	5.886	5.917
	std	0.594	13.449	0.632	0.738	0.604	0.633

Best result is shown in bold for each metric



**Table 3** Quantitative metrics for Group B images (Liu = algorithm from Liu et al. [10]; Singh = algorithm from Singh et al. [22]; BBO = basic biogeography-based optimization; LBBO-Ring = local

BBO with ring topology; LBBO-Random = local BBO with random topology; and EBO = ecogeography-based optimization)

Index	Results	Liu	Singh	BBO	LBBO-Ring	LBBO-Random	EBO
Precision	average	0.792	0.700	0.921	0.906	<b>0.930</b>	<b>0.930</b>
	max	0.960	0.994	1.000	0.999	1.000	1.000
	min	0.488	0.203	0.700	0.631	0.714	0.751
	std	0.170	0.328	0.114	0.145	0.109	0.093
Recall	average	0.903	<b>0.945</b>	0.924	0.919	0.911	0.918
	max	0.986	1.000	0.989	0.991	0.990	0.992
	min	0.784	0.765	0.800	0.781	0.764	0.779
	std	0.077	0.090	0.067	0.078	0.087	0.079
F-measure	average	0.837	0.750	0.916	0.897	0.914	<b>0.919</b>
	max	0.973	0.979	0.983	0.983	0.981	0.983
	min	0.593	0.332	0.725	0.687	0.716	0.753
	std	0.129	0.243	0.099	0.113	0.104	0.089
Time (in sec) per image	average	8.005	18.260	6.725	6.697	<b>6.674</b>	6.687
	max	8.383	29.197	7.452	7.360	7.184	7.117
	min	7.782	11.760	6.310	6.192	6.260	6.339
	std	0.271	7.081	0.431	0.425	0.345	0.294

Best result is shown in bold for each metric

### 5.3 Discussion

Recall is a measure of how much of the ground truth is detected. Clearly, the Singh et al. model had the highest

recall rates of all methods tested. This is because the Singh et al. model always contains unnecessary information on regions surrounding the salient objects. A high recall rate can be achieved by simply selecting an attention region as

**Table 4** Quantitative metrics for Group C images (Liu = algorithm from Liu et al. [10]; Singh = algorithm from Singh et al. [22]; BBO = basic biogeography-based optimization; LBBO-Ring = local

BBO with ring topology; LBBO-Random = local BBO with random topology; and EBO = ecogeography-based optimization)

Index	Results	Liu	Singh	BBO	LBBO-Ring	LBBO-Random	EBO
Precision	average	0.720	0.668	0.878	0.858	<b>0.879</b>	0.866
	max	0.995	0.843	0.975	0.975	0.973	0.966
	min	0.221	0.447	0.784	0.781	0.801	0.766
	std	0.295	0.186	0.072	0.071	0.064	0.076
Recall	average	0.904	<b>0.957</b>	0.831	0.840	0.844	0.820
	max	0.986	1.000	0.976	0.975	0.958	0.984
	min	0.823	0.862	0.702	0.694	0.744	0.611
	std	0.071	0.056	0.125	0.119	0.101	0.152
F-measure	average	0.760	0.759	0.824	0.838	<b>0.851</b>	0.831
	max	0.900	0.896	0.913	0.896	0.898	0.904
	min	0.359	0.582	0.749	0.731	0.776	0.675
	std	0.228	0.130	0.071	0.074	0.059	0.096
Time (in sec) per image	average	8.163	15.374	6.859	<b>6.778</b>	6.787	6.786
	max	8.434	16.366	7.182	7.180	7.167	7.127
	min	7.901	14.607	6.365	6.420	6.403	6.450
	std	0.277	0.737	0.351	0.328	0.329	0.304

Best result is shown in bold for each metric



**Table 5** Area under curve (AUC) values and peak memory use of different algorithms (Liu = algorithm from Liu et al. [10]; Singh = algorithm from Singh et al. [22]; BBO = basic

biogeography-based optimization; LBBO-Ring = local BBO with ring topology; LBBO-Random = local BBO with random topology; and EBO = ecogeography-based optimization)

Index	Liu	Singh	BBO	LBBO-Ring	LBBO-Random	EBO
AUC	0.858	0.846	<b>0.965</b>	0.935	<b>0.965</b>	0.959
Peak memory usage (in Mb) per image	399.641	400.500	398.543	397.078	395.988	<b>395.387</b>

Best result is shown in bold for each metric

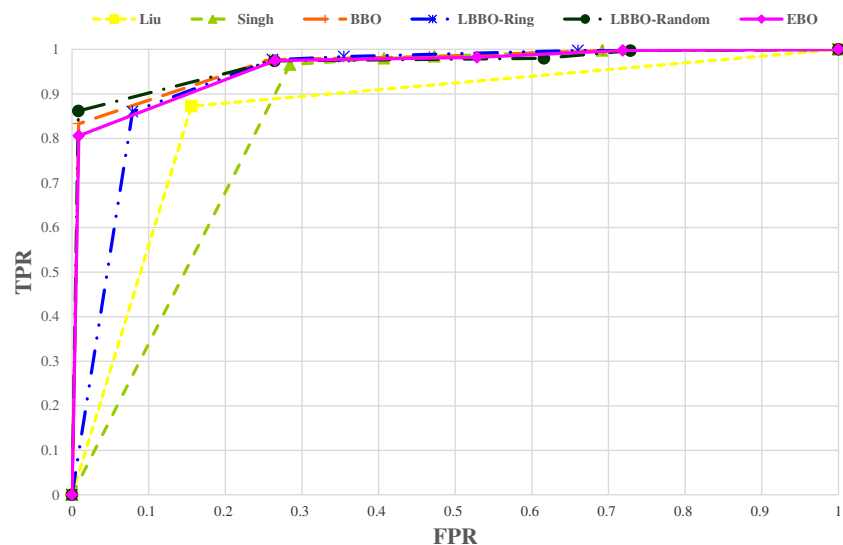
large as possible. Therefore, it is not a very useful measure in SOD [10]. In contrast, precision is a measure of how much noise is in the salient result. High precision denotes accurate demarcation of a salient object, which is a main objective of SOD. Although for an image with a large salient object, a high precision can be achieved by simply selecting a large attention region, it is difficult to achieve high precision on an image with a small salient object. We selected images containing small objects to test this tactic, e.g., the image in the second row of Fig. 7; our method yielded high precision for all such examples. As the weighted harmonic mean of both precision and recall, the F-measure is an overall performance measure. F-measures for our experiments show that all four of our BBO algorithms consistently outperformed other algorithms in this study.

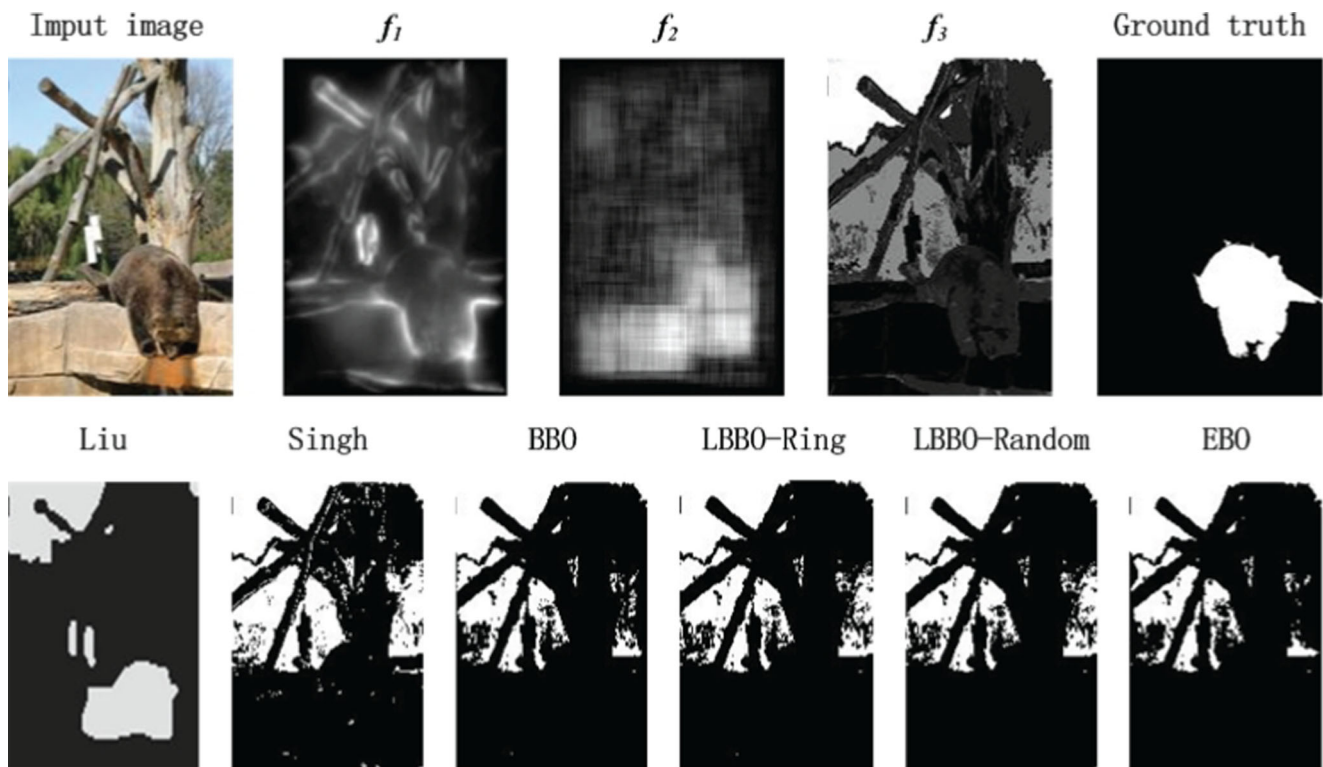
Although the Singh et al. model and our four BBO algorithms require an optimization algorithm involving an iterative process that is computationally expensive, our BBO algorithms ran much faster than the Singh et al. model. Because the capability of global optimization in the four BBO algorithms depresses the influence of using a threshold, a fixed threshold was selected. Thus, we saved the computational effort required for calculating an adjusted

threshold in the Singh et al. model, without any loss of performance. In fact, even when we adopt the adjusted threshold method in the Singh et al. model, our computation time per image is around 7 s, which is still faster than both the Singh et al. and Liu et al. models. There were no statistically significant differences between peak memory use among the different algorithms; this reflects that the calculation of three feature maps requires the largest memory storage in each algorithm.

Because the Liu et al. model uses a single common linear weight vector that is based on the training set to combine the feature maps for all test images; there is a slight probability for it to output an optimal salient object. In contrast, the C-PSO algorithm in the Singh et al. model often gets trapped in local optima. Thus, its ability to find a global optimal salient object is much lower than our method. There were no statistically significant differences among the four versions of the BBO algorithms. For Group A images, our four BBO algorithms yielded an optimal salient object in each run; thus, their values were the same. However, in the other two groups (B, C), the precision values and F-measures for the LBBO-Random and EBO were slightly better than for BBO and LBBO-Ring. This is because local random topology enhances the exploitation capability in BBO.

**Fig. 9** The receiver operator characteristic (ROC) curves for various algorithms. TPR = true-positive rate; FPR = false-positive rate; Liu = algorithm from Liu et al. [10]; Singh = algorithm from Singh et al [22]; BBO = basic biogeography-based optimization; LBBO-Ring = local BBO with ring topology; LBBO-Random = local BBO with random topology; and EBO = ecogeography-based optimization





**Fig. 10** Results showing failure to identify the salient object based on three feature maps ( $f_1$ ,  $f_2$ ,  $f_3$ ) in various algorithms. Liu = algorithm from Liu et al. [10]; Singh = algorithm from Singh et al. [22];

BBO = basic biogeography-based optimization; LBBO-Ring = local BBO with ring topology; LBBO-Random = local BBO with random topology; and EBO = ecogeography-based optimization

## 6 Conclusion

In this paper, we proposed four BBO algorithms to combine multi-scale contrast, center-surround histogram, and on color spatial distribution feature maps to create a saliency mask. Experiments carried out on three popular image databases showed that our proposed approach has a significant performance advantage over other state-of-the-art methods. The LBBO-Random algorithm exhibited the best performance. Moreover, our method is not only computational efficient, but also has robust and accurate saliency estimation.

Some issues require further investigation. When the multi-scale contrast, center-surround histogram, and color spatial distribution features fail to describe the salient object accurately, it is very difficult for our method, Liu et al. and Singh et al. models to output a good result based on these three feature maps. Figure 10 illustrates a case where these algorithms fail. Currently we are adding more sophisticated visual features to improve the performance of our method. Such work will aim to develop a SOD-specific BBO variant. We are also adapting some very recent and efficient meta-heuristics including water wave optimization [38] and optics

inspired optimization [39] for solving the SOD optimization problem.

## References

1. Borji A, Sihite DN, Itti L (2014) What/where to look next? Modeling top-down visual attention in complex interactive environments. *IEEE Trans Syst Man Cyber A* 44(5):523–538. doi:[10.1109/TSMC.2013.2279715](https://doi.org/10.1109/TSMC.2013.2279715)
2. Hayhoe M, Ballard D (2005) Eye movements in natural behavior. *Trends Cogn Sci* 9(4):188–194. doi:[10.1016/j.tics.2005.02](https://doi.org/10.1016/j.tics.2005.02)
3. Pashler HE, Sutherland S (1998) *The psychology of attention*, vol 15. MIT press, MA, Cambridge
4. Frintrop S (2010) General object tracking with a component-based target descriptor. In: *IEEE International Conference on Robotics and Automation*, pp 4531–4536. doi:[10.1109/ROBOT.2010.5509638](https://doi.org/10.1109/ROBOT.2010.5509638)
5. Itti L, Koch C, Niebur E (1998) A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans Pattern Anal* 20(11):1254–1259. doi:[10.1109/34.730558](https://doi.org/10.1109/34.730558)
6. Pham DL, Xu C, Prince JL (2000) Current methods in medical image segmentation 1. *Annu Rev Biomed Eng* 2(1):315–337. doi:[10.1146/annurev.bioeng.2.1.315](https://doi.org/10.1146/annurev.bioeng.2.1.315)
7. Talha AM, Junejo IN (2014) Dynamic scene understanding using temporal association rules. *Image Vision Comput* 32(12):1102–1116. doi:[10.1016/j.imavis.2014.08.010](https://doi.org/10.1016/j.imavis.2014.08.010)

8. Zhang W, Wu QJ, Wang G, Yin H (2010) An adaptive computational model for salient object detection. *IEEE Trans Multimedia* 12(4):300–316. doi:[10.1109/TMM.2010.2047607](https://doi.org/10.1109/TMM.2010.2047607)
9. Shen X, Wu Y (2012) A unified approach to salient object detection via low rank matrix recovery. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp 853–860. doi:[10.1109/CVPR.2012.6247758](https://doi.org/10.1109/CVPR.2012.6247758)
10. Liu T, Yuan Z, Sun J, Wang J, Zheng N, Tang X, Shum H-Y (2011) Learning to detect a salient object. *IEEE Trans Pattern Anal* 33(2):353–367. doi:[10.1109/TPAMI.2010.70](https://doi.org/10.1109/TPAMI.2010.70)
11. Simon D (2008) Biogeography-based optimization. *IEEE Trans Evolut Comput* 12(6):702–713. doi:[10.1109/TEVC.2008.919004](https://doi.org/10.1109/TEVC.2008.919004)
12. Song Y, Liu M, Wang Z (2010) Biogeography-based optimization for the traveling salesman problems. In: *3rd international joint conference on computational science and optimization*, pp 295–299. doi:[10.1109/CSO.2010.79](https://doi.org/10.1109/CSO.2010.79)
13. Boussaïd I, Chatterjee A, Siarry P, Ahmed-Nacer M (2012) Biogeography-based optimization for constrained optimization problems. *Comput Oper Res* 39:3293–3304
14. Zheng Y-J, Ling H-F, Shi H-H, Chen H-S, Chen S-Y (2014) Emergency railway wagon scheduling by hybrid biogeography-based optimization. *Comput Oper Res* 43:1–8. doi:[10.1016/j.cor.2013.09.002](https://doi.org/10.1016/j.cor.2013.09.002)
15. Hadidi A (2015) A robust approach for optimal design of plate fin heat exchangers using biogeography based optimization (BBO) algorithm. *Appl Energy* 150:196–210. doi:[10.1016/j.apenergy.2015.04.024](https://doi.org/10.1016/j.apenergy.2015.04.024)
16. Zheng X-W, Lu D-J, Wang X-G, Liu H (2015) A cooperative coevolutionary biogeography-based optimizer. *Appl Intell* 43:95–111. doi:[10.1007/s10489-014-0627-9](https://doi.org/10.1007/s10489-014-0627-9)
17. Zheng Y-J, Ling H-F, Chen S-Y, Xue J-Y (2015) A hybrid neuro-fuzzy network based on differential biogeography-based optimization for online population classification in earthquakes. *IEEE Trans Fuzzy Syst* 23:1070–1083. doi:[10.1109/TFUZZ.2014.2337938](https://doi.org/10.1109/TFUZZ.2014.2337938)
18. Zheng Y-J, Ling H-F, Xu X-L, Chen S-Y (2015) Emergency scheduling of engineering rescue tasks in disaster relief operations and its application in China. *Int Trans Oper Res* 22:503–518. doi:[10.1111/itor.12148](https://doi.org/10.1111/itor.12148)
19. Zhou X, Liu Y, Li B, Sun G (2015) Multiobjective biogeography based optimization algorithm with decomposition for community detection in dynamic networks. *Physica A* 436:430–442. doi:[10.1016/j.physa.2015.05.069](https://doi.org/10.1016/j.physa.2015.05.069)
20. Zheng Y-J, Ling H-F, Wu X-B, Xue J-Y (2014) Localized biogeography-based optimization. *Soft Comput* 18(11):2323–2334. doi:[10.1007/s00500-013-1209-1](https://doi.org/10.1007/s00500-013-1209-1)
21. Zheng Y-J, Ling H-F, Xue J-Y (2014) Ecogeography-based optimization: enhancing biogeography-based optimization with ecogeographic barriers and differentiations. *Comput Oper Res* 50:115–127. doi:[10.1016/j.cor.2014.04.013](https://doi.org/10.1016/j.cor.2014.04.013)
22. Singh N, Arya R, Agrawal R (2014) A novel approach to combine features for salient object detection using constrained particle swarm optimization. *Pattern Recogn* 47(4):1731–1739. doi:[10.1016/j.patcog.2013.11.012](https://doi.org/10.1016/j.patcog.2013.11.012)
23. Cheng M-M, Zhang G-X, Mitra NJ, Huang X, Hu S-M (2011) Global contrast based salient region detection. *IEEE Conf Comput Vis Pattern Recognit* 409–416. doi:[10.1109/CVPR.2011.5995344](https://doi.org/10.1109/CVPR.2011.5995344)
24. Rahtu E, Heikkilä J (2009) A simple and efficient saliency detector for background subtraction. In: *IEEE 12th international conference on computer vision workshops*, pp 1137–1144. doi:[10.1109/ICCVW.2009.5457577](https://doi.org/10.1109/ICCVW.2009.5457577)
25. Harel J, Koch C, Perona P (2006) Graph-based visual saliency. *Adv Neur In*:545–552
26. Hou X, Zhang L (2007) Saliency detection: A spectral residual approach. *IEEE Conf Comput Vis Pattern Recognit* 1–8. doi:[10.1109/CVPR.2007.383267](https://doi.org/10.1109/CVPR.2007.383267)
27. Achanta R, Hemami S, Estrada F, Susstrunk S (2009) Frequency-tuned salient region detection. *IEEE Conf Comput Vis Pattern Recognit* 1597–1604. doi:[10.1109/CVPR.2009.5206596](https://doi.org/10.1109/CVPR.2009.5206596)
28. Ma Y-F, Zhang H-J (2003) Contrast-based image attention analysis by using fuzzy growing. In: *Proceedings of the 11th ACM international conference on multimedia*, pp 374–381. doi:[10.1145/957013.957094](https://doi.org/10.1145/957013.957094)
29. Navalpakkam V, Itti L (2006) An integrated model of top-down and bottom-up attention for optimizing detection speed. In: *IEEE computer society conference on computer vision and pattern recognition*, pp 2049–2056. doi:[10.1109/CVPR.2006.54](https://doi.org/10.1109/CVPR.2006.54)
30. Goferman S, Zelnik-Manor L, Tal A (2012) Context-aware saliency detection. *IEEE Trans Pattern Anal* 34(10):1915–1926. doi:[10.1109/TPAMI.2011.272](https://doi.org/10.1109/TPAMI.2011.272)
31. Gao D, Vasconcelos N (2007) Bottom-up saliency is a discriminant process. In: *IEEE 11th international conference on computer vision*, pp 1–6. doi:[10.1109/ICCV.2007.440885](https://doi.org/10.1109/ICCV.2007.440885)
32. Martin DR, Malik J (2004) Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Pattern Anal Machine Intell* 26:530–549. doi:[10.1109/TPAMI.2004.1273918](https://doi.org/10.1109/TPAMI.2004.1273918)
33. Batra D, Kowdle N, Parikh D, Luo J, Chen T (2010) Icoseg: interactive co-segmentation with intelligent scribble guidance. *IEEE Conf Comput Vis Pattern Recognit* 1597–1604. doi:[10.1109/CVPR.2010.5540080](https://doi.org/10.1109/CVPR.2010.5540080)
34. Batra D, Kowdle N, Parikh D, Luo J, Chen T (2011) Interactively co-segmenting topically related images with intelligent scribble guidance. *Int J Comput Vision* 93(3):273–292. doi:[10.1007/s11263-010-0415-x](https://doi.org/10.1007/s11263-010-0415-x)
35. Alpert S, Galun M, Basri R, Brandt A (2007) Image segmentation by probabilistic bottom-up aggregation and cue integration. *IEEE Conf Comput Vis Pattern Recognit* 1–8. doi:[10.1109/CVPR.2007.383017](https://doi.org/10.1109/CVPR.2007.383017)
36. Jiang H, Wang J, Yuan Z, Wu Y, Zheng N, Li S (2013) Salient Object Detection: A Discriminative Regional Feature Integration Approach. *IEEE Conf Comput Vis Pattern Recognit* 2083–2090. doi:[10.1109/CVPR.2013.271](https://doi.org/10.1109/CVPR.2013.271)
37. Dhar S, Ordonez V, Berg TL (2011) High level describable attributes for predicting aesthetics and interestingness. *IEEE Conf Comput Vis Pattern Recognit* 1657–1664. doi:[10.1109/CVPR.2011.599546](https://doi.org/10.1109/CVPR.2011.599546)
38. Zheng YJ (2015) Water wave optimization: a new nature-inspired metaheuristic. *Comput Oper Res* 55:1–11. doi:[10.1016/j.cor.2014.10.008](https://doi.org/10.1016/j.cor.2014.10.008)
39. Kashan AH (2015) A new metaheuristic for optimization: optics inspired optimization (OIO). *Comput Oper Res* 55:99–125. doi:[10.1016/j.cor.2014.10.011](https://doi.org/10.1016/j.cor.2014.10.011)



**Zhicheng Wang** received Ph.D. degree from Huazhong University of Science and Technology in 2006. He is currently the Director of CAD Research Center of Tongji University. His research topics include pattern recognition and computer vision.



**Xiaobei Wu** received the bachelor's degree of software engineering from Zhejiang University of Technology, China, in 2013. Currently, she is a master student of computer science and technology in Tongji University, China. Her current research interests include evolutionary algorithms, integer programming and image processing.