Multimed Tools Appl (2018) 77:23699–23727
https://doi.org/10.1007/s11042-018-5637-x

Meta-heuristic moth swarm algorithm for multilevel thresholding image segmentation

Yongquan Zhou[1,2] · **Xiao Yang**[1,2] · **Ying Ling**[1,2] · **Jinzhong Zhang**[1,2]

Received: 1 July 2017 / Revised: 15 December 2017 / Accepted: 9 January 2018 /
Published online: 1 February 2018

Abstract Multilevel thresholding is a very important image processing technique in the field of image segmentation. However, the computational complexity of determining the optimal threshold grows exponentially with increasing thresholds. To overcome this drawback, in this paper, we propose a multi-threshold image segmentation method based on the moth swarm algorithm. The meta-heuristic algorithm uses Kapur's entropy method to optimize the thresholds for eight standard test images. When compared with other state-of-the-art evolutionary algorithms, the proposed method proved to be robust and effective according to numerical experimental results and image segmentation results. This indicates the high performance of the method for the segmentation of digital images.

Keywords Multilevel thresholding · Moth swarm algorithm · Image segmentation · Meta-heuristic · Kapur's entropy

1 Introduction

Image segmentation is the technology and process of dividing an image into several specific and unique areas and presenting the object of interest. The process involves dividing an image into different segmentation pixel classes with features such as grayscale, color, and texture.

[23]. There is some higher-level processing, that is, image analysis, object recognition, and computer vision. Image segmentation is often used as the preprocessing stage of this higher-level processing. Different types of methods exist for image segmentation. Thresholding is one

✉ Yongquan Zhou
yongquanzhou@126.com

[1] College of Information Science and Engineering, Guangxi University for Nationalities, Nanning 530006, China

[2] Key Laboratories of Guangxi High Schools Complex System and Computational Intelligence, Nanning, Guangxi 530006, China

of the main methods and has the ability to search for the optimal threshold. The thresholding technique consists of bi-level thresholding and multilevel thresholding [1, 25]. Bi-level thresholding separates the pixels in the image into two parts, whereas multilevel thresholding separates the pixels into several parts [8]. Image processing has many aspects, such as image enhancement, image homogenization, and image segmentation. Pattern recognition is sometimes included in image processing. The main content of machine learning is generalization, which separates two or more items according to their characteristics [16, 17]. Machine learning is a science of artificial intelligence [18, 19]. The main research area in this field is artificial intelligence, in particular, how to improve the performance of specific algorithms in experience learning [14]. In image segmentation, it is often possible to manually mark the work; however, it is difficult to write a complete rule for automatic processing. Sometimes there is an entire set of algorithms; however, there are too many parameters, and it is too tedious to manually adjust and determine the correct parameters. We can use the machine learning method to extract a certain number of features and manually mark a batch of results, and then use machine learning to determine a set of automatic judgment criteria [15]. Machine learning is more effective in developing such software.

Many computer scientists and scholars have studied image segmentation for many years and have proposed some innovations that have been explored and expected in the literature. An automatic threshold selection method based on the information theory entropy criterion proposed by Otsu in 1979 caused great concern, and both theoretical research and practical applications have made significant breakthroughs [22]. Pun and Kapur proposed the method of using the maximum a priori entropy to estimate the classification rationality to select the threshold in 1980 [24] and 1985 [9], respectively. Yen proposed a method of selecting the threshold using the principle of maximum relativity instead of the commonly used principle of maximum entropy [31]. In 1999, Yin designed an improved genetic algorithm and embedded learning strategies to enhance its multi-threshold search capability [32]. This method greatly reduced the computational cost of the multilevel threshold and had good segmentation results. In 2004, Lai studied Gaussian smoothing in detail, proposed a genetic algorithm based on it, and applied the algorithm to image segmentation [13]. In 2008, Maitra introduced a cooperative learning operator and comprehensive learning operator in particle swarm optimization (PSO), which enhanced the image segmentation ability of the algorithm. The comprehensive learning operator reduced the risk of the premature algorithm. The cooperative learning operator was devoted to solving the problem of the dimensionality disaster [20]. In 2012, the differential evolution (DE) algorithm based on the Gaussian distribution function was applied in multi-threshold segmentation. The calculation complexity of the algorithm was reduced, but it was still high [3]. In 2014, Bhandari proposed a method called ELR-CS to solve multi-threshold image segmentation and compared it in a large number of experiments with the cuckoo search (CS) algorithm and wind driven optimization (WDO) algorithm, which indicated that it had good segmentation performance [2]. In 2015, Wang applied the flower pollination algorithm (FPA) based on a modified randomized location to multi-threshold medical image segmentation [28]. In 2017, Khairuzzaman used the gray wolf algorithm to optimize Otsu's objective function and Kapur's method to solve multi-threshold image segmentation. The experimental results demonstrated that the proposed gray wolf optimization (GWO) was more stable and obtained higher-quality solutions than the PSO and BFO algorithms

[12]. Aziz applied the whale optimization algorithm (WOA) and moth-flame optimization (MFO) for multilevel thresholding image segmentation using the two algorithms in Otsu's method and Kapur's method [5], respectively. The WOA and MFO algorithms were better than the other compared algorithms for almost all the test images. WOA was superior to MFO under the peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) tests.

Recently, the entropy-based method for multilevel thresholding image segmentation has been quite popular, in particular, Ostu's criterion [12], cross entropy, Tsallis entropy [26], and Kapur's entropy [12]. Kapur's entropy is a nonparametric threshold technique that maximizes entropy to calculate the homogeneity of classes. Among these thresholding methods, Kapur's entropy has attracted the attention of researchers and has been shown to be more superior than other thresholding methods. However, to select the optimal threshold, an exhaustive search using Kapur's entropy requires much more execution time for increasing thresholds. To overcome this problem, researchers have used swarm intelligent algorithms that are inspired by nature. The most popular algorithms are the genetic algorithm (GA) [6], (PSO) [11], (DE) [27], artificial bee colony (ABC) [10], and firefly algorithm (FA) [30]. Researchers have also used swarm optimization with thresholding methods to solve multilevel thresholding image segmentation. Akay used PSO and ABC for multilevel thresholding to maximize Kapur's entropy [14]. In [4], Sathyaet et al. used a modified PSO with minimum cross entropy to search for the optimal threshold. These studies have demonstrated the power of combining the swarm algorithm with thresholding methods to manage image segmentation.

Determining the optimal thresholding for image segmentation has gained more attention in recent years. It is a valuable foundational technology for digital image processing and machine vision and it is often used as a preprocessing stage for applications such as pattern recognition. However, traditional multilevel thresholding methods are computationally expensive because they involve exhaustively searching the optimal thresholds to optimize the objective functions. In this paper, we propose an algorithm that is inspired by the orientation of moths toward moonlight called the moth swarm algorithm (MSA), [21] which is used to solve multilevel thresholding for image segmentation, thereby overcoming some deficiencies of other algorithms. The proposed method selects the optimal set of thresholds using Kapur's entropy function. The simulation results show that the MSA obtained better results when compared with the WOA [5], bat algorithm (BA) [29], GWO [12], and FPA [28] in terms of the PSNR [2], SSIM [2], computational times, and Kapur's entropy fitness function. Multi-threshold image segmentation is a powerful image processing technique that is used for the preprocessing of pattern recognition and computer vision. The performance of the higher-level processing system depends on the accuracy of the segmentation technique used. The MSA has the advantage that it is simple and universal, has strong robustness, is suitable for parallel processing, and has a wide application range. The MSA is an evolutionary method that is inspired by the phototropism of moths and transverse orientation. Different to other evolutionary algorithms, the MSA exhibits interesting search capabilities while maintaining a low computational overhead. Segmentation is one of the most important tasks in image processing that endeavors to identify whether the pixel intensity corresponds to a predefined

class. Applying the MSA to image segmentation is a new method of image segmentation.

The remainder of this paper is structured as follows: In Section 2, we introduce the multi-threshold and Kapur entropy. In Section 3, we briefly present the concept of the MSA. In Section 4, we present the proposed MSA-based multilevel thresholding method and its pseudocode. In Section 5, we present the experimental results and discussions. In the final section, we conclude the study and suggest some directions for future studies.

2 Multilevel thresholding

Bi-level thresholding divides an image into two parts: one is the object and the other is the background. Bi-level thresholding is effective if the image is simple, that is, it contains only an object; however, if the image is complex and involves many objects, bi-level thresholding may fail to provide the appropriate performance [12]. As a result, multilevel thresholding is often used instead of bi-level thresholding to segment complex images. To obtain a good segmentation result, it is essential to choose the prober values of these thresholds. Optimal threshold selection methods search for thresholds by optimizing an objective function. To obtain a good thresholding function, entropy-based methods have been found to be efficient and feasible. Kapur's entropy is one of the most popular techniques used for optimal thresholding techniques. The concept of Kapur's entropy is briefly introduced in the following subsection.

2.1 Concept of Kapur's entropy

Kapur et al. supposed that there are double probability distributions that are the object and background. Therefore, maximizing all the entropy of the partitioned image is the major step to obtain the best threshold level. In the situation in which the optimal thresholds for segmenting the classes are assigned appropriately, then only the maximum entropy is required and the most suitable. Thus, the major purpose in this paper is to search for the optimal threshold (the best fitness value) that yields the maximum entropy using Kapur's entropy technique and the MSA.

Let there be K gray levels in a given image, which is in the range of $\{0, 1, 2\ldots(K-1)\}$. Let N be the whole number of pixels in the image and n_i denote the number of pixels at gray level i. Then $p_i = n_i/N$ represents the probability of the occurrence of gray level i in image K. Kapur's entropy defines an image completely represented by its corresponding gray-level histogram. Consider that there exist m thresholds $[t_1, t_2, ..t_m]$ to be chosen that divide the image into many parts: C_0, C_1, C_2, $\ldots C_m$. Therefore, Kapur's entropy is obtained using.

$$J_2(t_1, t_2, \ldots t_m) = H_0 + H_1 + H_2 + \ldots + H_m \tag{1}$$

where

$$H_1 = -\sum_{i=0}^{t_1-1}(p_i/\omega_0)\ln(p_i/\omega_0),\ \ \omega_0 = \sum_{i=0}^{t_1-1} p_i;\quad H_2 = -\sum_{i=t_1}^{t_2-1}(p_i/\omega_1)\ln(p_i/\omega_1),\ \ \omega_1 = \sum_{i=t_1}^{t_2-1} p_i;$$
$$H_3 = -\sum_{i=t_2}^{t_3-1}(p_i/\omega_2)\ln(p_i/\omega_2),\ \ \omega_2 = \sum_{i=t_2}^{t_3-1} p_i;\quad H_m = -\sum_{i=t_m}^{K-1}(p_i/\omega_m)\ln(p_i/\omega_m),\ \ \omega_m = \sum_{i=t_m}^{t_m-1} p_i;$$

where $H_1, H_2, \ldots H_m$ represent Kapur's entropies and $\omega_0, \omega_1, \omega_2, \ldots, \omega_m$ denote the class probabilities of the segmented classes $C_0, C_1, C_2, \ldots C_m$, respectively. The purpose of the present study is to use the MSA to maximize Kapur's objective function, which is defined in Eq. (1).

3 Moth swarm algorithm

The MSA is a meta-heuristic proposed by Al-Attar Ali Mohamed [21] that is inspired by the special behavior of moths. A brief mathematical model of the MSA is provided in the following subsections.

3.1 Basic concepts

In the basic MSA, the position of the light source represents a possible solution to the problem to be optimized and the luminescence intensity of the light source represents the fitness of this solution. These assumptions are used to approximate the characteristics of the proposed algorithm. In the MSA, the entire moth population is divided into three groups of moths, which are defined as follows:

Pathfinders: This group of moths (n_p) can discover new areas in the optimization space according to the first-in, last-out principle. Pathfinders distinguish the best position of light sources to guide the movement of the main groups.
Prospectors: This group of moths flies into a stochastic spiral path in the neighborhood of the light sources that have been marked by the group of pathfinders.
Onlookers: This group of moths flies directly to the best global solution (moonlight) that has been determined by the prospectors.

3.2 Mathematical model

For each iteration, each moth x_i is integrated into the optimization problem to determine the luminescence intensity of its corresponding light source $f(x_i)$. The pathfinders' positions are considered as the best fitness in the swarm and they provide guidance for the next update iteration. Therefore, the prospectors and onlookers in the swarm are the second the third-best groups. The MSA is executed as follows:

3.2.1 Initialization

At the beginning of the flight, a set of moths that are the candidate solution are randomly generated as follows:

$$x_{i,j} = rand[0,1]\cdot\left(x_j^{\max} - x_j^{\min}\right) + x_j^{\min} \quad \forall \; i\in\{1,2\ldots,n\};\;\; j\in\{1,2,\ldots d\} \tag{2}$$

where $x_j^{\max}$ represents the upper limit and $x_j^{\min}$ represents the lower limit.

After executing the initialization, the moths are divided into types according to their calculated fitness. Hence, the best moths are defined as light sources (pathfinders), the second best are defined as prospectors, and the worse are defined as onlookers.

3.2.2 Reconnaissance stage

n the MSA, the quality of the swarm for exploration may decrease over the course of iterations. The moths can become stagnant in an area, which appears to be good and easily achieves the local optimum. To eliminate premature convergence and enhance the diversity of solutions, pathfinder moths search for less-crowded places to guide other groups. Pathfinder moths update their positions by interacting with each other (crossover operations) and with the ability to fly for long distances (Lévy mutation), thereby using the adaptive crossover with Lévy mutation, which is presented in the following four steps.

Diversity index for crossover points To improve the diversity of the solutions, a new strategy is used to select the crossover points. First, for iteration t, normalized dispersal degree σ_j^t of the individuals in the *jth* dimension can be calculated as follows:

$$\sigma_j^t = \frac{\sqrt{\frac{1}{n_p}\sum_{i=1}^{n_p}\left(x_{ij}^t - \overline{x_j^t}\right)^2}}{\overline{x_j^t}} \tag{3}$$

where $\overline{x_j^t} = \frac{1}{n_p}\sum_{i=1}^{n_p} x_{ij}^t$, n_p represents the number of pathfinder moths and μ^t is a variation coefficient used to measure the relative dispersion and can be calculated as follows:

$$\mu^t = \frac{1}{d}\sum_{j=1}^{d}\sigma_j^t \tag{4}$$

Any element among the pathfinder moths that has a low degree of disposal is accepted in group c_p, which is the cross point group defined as follows:

$$j\in c_p \qquad if\, \sigma_j^t \le \mu^t \tag{5}$$

Thus, with the new strategy, the group of crossover points can change dynamically over the course of the iterations.

Lévy flights A Lévy flight is a class of random walk that is based on a power-law distribution called α - *stable* distribution, which can travel a large distance using different types of steps. For Lévy flights, Mantegna's algorithm [7] is used to emulate the α - *stable* distribution by generating random samples L_i that have the same behavior as Lévy flights, which is defined as follows:

$$L_i \sim step \oplus Levy(\alpha) \sim 0.01\frac{u}{|y|^{1/\alpha}} \tag{6}$$

where *step* denotes the scaling size that corresponds to the scales of the interest scales, $\oplus$

represents the dot product (entrywise multiplication), $\mu = N\left(0, \sigma_\mu^2\right)$ and $\nu = N\left(0, \sigma_y^2\right)$ are both normal stochastic distributions with $\sigma_\mu = \left[\frac{\Gamma(1+\beta)\times\sin(\pi\times\beta/2)}{\Gamma(((1+\beta/2))\times\beta\times 2^{(\beta-1)/2}}\right]^{1/\alpha}$, and $\sigma_y = 1$.

Difference vectors Lévy mutation For $n_c \in n_p$ crossover execution points, the MSA addresses the sub-trail vector $\overrightarrow{v_p} = \left[v_{p1}, v_{p2}, \ldots, v_{pn_c}\right]$ by disturbing the constituents that have been selected from the host vector $\overrightarrow{x_{r^1}} = \left[x_{r^1 1}, x_{r^2 2}, \ldots x_{r^3 n_c}\right]$, with the corresponding constituents in the donor vectors $\overrightarrow{x_{r^1}} = \left[x_{r^1 1}, x_{r^2 2}, \ldots x_{r^3 n_c}\right]$. The mutation mechanism can be used for synthesis as a sub-trail vector defined as follows:

$$\overrightarrow{v_p^t} = \overrightarrow{x_{r^1}^t} + L_{p1}^t \cdot \left(\overrightarrow{x_{r^2}^t} - \overrightarrow{x_{r^3}^t}\right) + L_{p2}^t \cdot \left(\overrightarrow{x_{r^4}^t} - \overrightarrow{x_{r^5}^t}\right) \quad \forall \; r^1 \neq r^2 \neq r^3 \neq r^4 \neq r^5 \neq p \in \{1, 2, \ldots, n_p\} \tag{7}$$

In Eq. (6), L_{p1}^t and L_{p2}^t are the two equivalent variables used as the mutation scaling factor and they are both generated by the power-law Lévy flights using (L_p~*random*(n_c) ⊕ *Levy*(α)). The series of indices ($r^1, r^2, r^3, r^4, r^5, p$), which are mutually indices, are selected exclusively from the pathfinder solutions.

Adaptive crossover operation based on population diversity To obtain the completed trail solution, each pathfinder solution, also called a host vector, updates its position using the crossover operation by integrating the mutated variables of the sub-trail vectors (low degree of dispersal) with the related variables of the host vector. The main trail solutions are defined as follows:

$$V_{pj}^t = \begin{cases} v_{pj}^t & if \;\; j \in c_p \\ x_{pj}^t & if \;\; j \notin c_p \end{cases} \tag{8}$$

Note that μ^t is the variation coefficient that is used to control the rate of crossover.

Roulette wheel selection After completing all the preceding steps, the fitness of the completed trail solution is calculated and compared with the related host solution. The better fitness solutions are chosen to survive for the next iteration, which is calculated as follows:

$$\overrightarrow{x_p^{t+1}} = \begin{cases} \overrightarrow{x_p^t} & if \;\; f\left(\overrightarrow{V_p^t}\right) \geq f\left(\overrightarrow{x_p^t}\right) \\ \overrightarrow{v_p^t} & if \;\; f\left(\overrightarrow{V_p^t}\right) < f\left(\overrightarrow{x_p^t}\right) \end{cases} \tag{9}$$

where P_p is the probability of the luminescence intensity fit_p, which is modeled as follows:

$$P_p = \frac{fit_p}{\sum_{p=1}^{n_p} fit_p} \tag{10}$$

The luminescence intensity is computed from the objective function value f_p for minimization problems and defined as follows:

$$fit_p = \begin{cases} \dfrac{1}{1+f_p} & for \;\; f_p \geq 0 \\ 1 + f_p & for \;\; f_p < 0 \end{cases} \tag{11}$$

3.2.3 Transverse orientation

The prospector moths are the second-best luminescence intensity group of moths. The number of prospector moths decreases during the process of iteration T, which is modeled as follows:

$$n_f = round\left(\left(n - n_p\right) \times \left(1 - \frac{t}{T}\right)\right) \tag{12}$$

After the pathfinder moths complete their search, they share information about the luminescence intensity with the prospectors, which attempt to update their positions to locate new light sources. Each prospector moth x_i flies into a logarithmic spiral path to perform a deep search around the related artificial light source x_p, which is selected probability is P_p in Eq. (10). The new position of the ith prospector moth is modeled as follows:

$$x_i^{t+1} = \left|x_i^t - x_p^t\right| \cdot e^{\theta} \cdot \cos 2\pi\theta + x_p^t \qquad \forall\ \mathrm{p} \in \left\{1, 2, \ldots, \mathrm{n_p}\right\};\ \ i \in \left\{n_p + 1, n_p + 2, \ldots, n_f\right\} \tag{13}$$

where θ is a random number range in $[r, 1]$ for defining the shape of the logarithmic spiral and $r = -1 - t/T$. In the model of the MSA, each moth dynamically changes its type. Therefore, if any prospectors search a solution that is more luminescent than the existing light sources, then they attempt to become a pathfinder moth; that is, the new lighting sources and moonlight are put forward at the end of this stage.

3.2.4 Celestial navigation

During the process of optimization, as the number of prospectors decreases, the number of onlookers increases ($n_o = n - n_f - n_p$), which may result in accelerating the convergence rate of the MSA to achieve a global solution. The moths with the lowest fitness value are onlooker moths. For this stage, the onlookers are divided into the two following parts.

Gaussian walks In this stage, the onlookers are forced to search for more promising areas in the search place: the first part, with size $n_G = round(n_o/2)$ flying with Gaussian distribution $q \sim N\left(\mu, \sigma_G^2\right)$ with density using

$$f(q) = \frac{1}{\sqrt{2\pi}\sigma G} \exp\left(-\frac{(q-\mu)^2}{2\sigma^2 G}\right) \qquad -\infty < q < \infty \tag{14}$$

The new onlookers in this subgroup x_i^{t+1} fly with the set of Gaussian walks as follows:

$$x_i^{t+1} = x_i^t + \varepsilon_1 + \left[\varepsilon_2 \times gbest^t - \varepsilon_3 \times x_i^t\right] \ \forall\ i \in \{1, 2, .., n_G\} \tag{15}$$

$$\varepsilon_1 \sim random(size(d)) \oplus N\left(best_g^t, \frac{\log t}{t} \times \left(x_i^t - best_g^t\right)\right) \tag{16}$$

where ε_1 denotes a random sample drawn from the Gaussian walks scaled to the size of this group, $best_g^t$ denotes the global best solution, and both ε_2 and ε_3 are random numbers in the range [0, 1].

Associative learning mechanism with immediate memory In this stage, the left part of the onlookers with size $n_A = n_o - n_G$ are used to drift to the moonlight taking into account the associative learning operators with an immediate memory to simulate the real behavior of moths in nature. The immediate memory is initialized from the continuous Gaussian distribution on the intervals of $x_i^t - x_i^{\min}$ and $x_i^{\max} - x_i^t$. The mathematical model of this stage can be defined as follows:

$$\begin{aligned} x_i^{t+1} = {} & x_i^t + 0.001 \cdot G\left[x_i^t - x_i^{\min}, x_i^{\max} - x_i^t\right] + (1 - g/G) \cdot r_1 \cdot \left(best_p^t - x_i^t\right) \\ & + \left(2g/G \cdot r_2 \cdot \left(best_g^t - x_i^t\right)\right. \end{aligned} \tag{17}$$

where $i \in \{1, 2, \ldots n_A\}$, and $2g/G$ and $1 - g/G$ denote the social factor and cognitive factor, respectively. Both r_1 and r_2 are random numbers in the range [0, 1]. $best_p$ denotes a light source stochastic selected from the new pathfinder group according to the probability value of its related solution.

4 Methodology

4.1 Proposed MSA-based multilevel thresholding method

Moths represent the search agents and their positions represent the thresholds to be optimized. Therefore, depending on the number of thresholds, the moths move in one-dimensional, two-dimensional, three-dimensional, or hyper-dimensional space by changing their position vectors. The positions of the moths are first initialized randomly. Then the fitness of all the moths is determined using Eq. (1). The positions of the moths are updated if better positions are determined. This process is repeated until the maximum number of iterations is completed. The best position of the moths provides the desired thresholds. The pseudocode for the proposed multilevel thresholding method is presented in the following subsection.

4.2 Pseudocode for MSA-based multilevel thresholding

Step 1: Initialize the moth groups (x_i)

Step 2: Calculate the objective function value of each moth using Eq. (1) for the Kapur-based method

Step 3: While t < the number of iterations **do**

For each pathfinder moth

Using Eq. (9), update x_i

Calculate the fitness using Eq. (1)

Sort according to fitness values; choose the better fitness values

End For

For each prospector moth

Update the position using Eq. (13)

Calculate the new fitness

Created the new light sources

End For

For each onlooker moth

If ($i \in n_0$)

Generate Gaussian walk steps

Update position x_i with the Gaussian walk using Eq. (15)

Else

Update position $x_i, i \in n_m$ using Eq. (17)

End If

Calculate the fitness

End For

Redefine the type of each moth

End While

Get thresholding.

4.3 Flowchart of MSA-based multilevel thresholding (see Fig. 1)

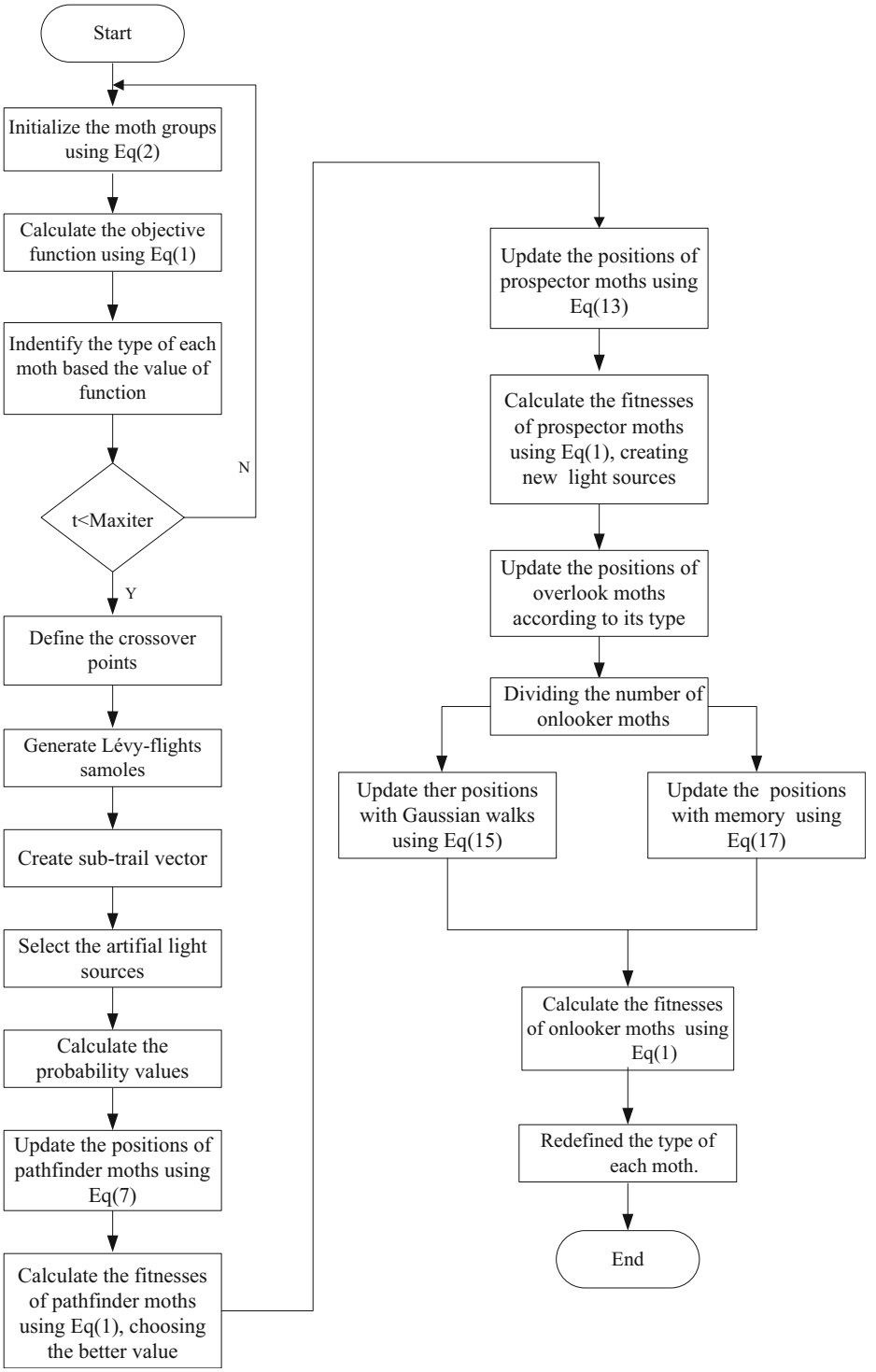


Fig. 1 Flowchart of the MSA

5 Experiments and discussion

The experimental setup for the proposed algorithm is briefly introduced in this section. First, we introduce the test images. Additionally, we present the parameter settings of each algorithm. Finally, we present the descriptions of the segmentation validation metrics.

5.1 Test images

In our study, the experiments were conducted on eight images that were carefully selected from the database of Berkeley University and shown in Fig. 2.

5.2 Segmented image quality metrics

The MSA used five methods to evaluate the performance of segmented images as follows:

(1) The fitness function value using Eq. (1). The larger the objective function value, the more information the segmented image contained. (2) The execution time of the MSA and other algorithms. The average execution time was used to compare the computational complexity of a multi-threshold approach. Less time indicated that an algorithm was faster than other algorithms. (3) The PSNR measured the difference between the segmented image and reference image based on the intensity values in the image. The larger the PSNR value, the fewer distortions were represented. Because the visual acuity of the human eye is not absolute, it is possible that a higher PSNR value may appear to be worse than a lower PSNR value. (4) The SSIM is a measure of the similarity of two images. When two images are identical, the value of the SSIM equals one. (5) A statistical analysis using the Wilcoxon rank sum test was performed at a 5% significance level. It demonstrated whether there was a meaningful difference among the five algorithms. A value of less than 0.05 indicated that it was maintained at the significance level.

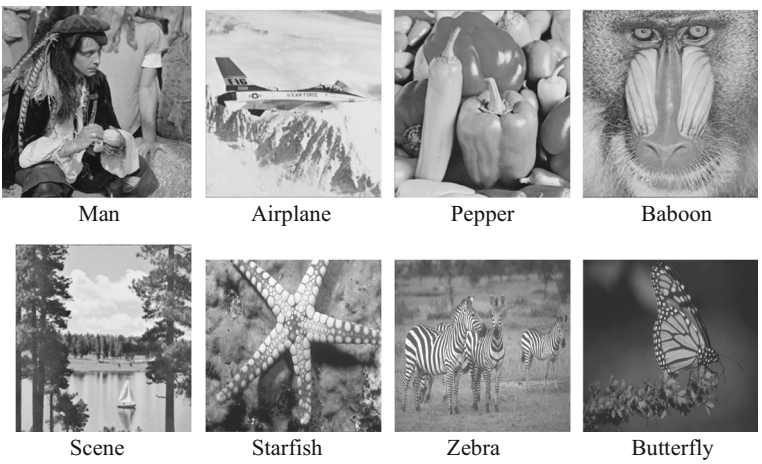


Fig. 2 Original images

Table 1 Comparison of the best fitness values for all the algorithms

Images	K	Fitness values					
		WOA	GWO	FPA	BA	MSA	Rank
Man	2	12.4285	12.6357	12.6215	12.6357	12.6582	1
	3	15.8115	15.8115	15.7602	15.6183	15.8725	1
	4	18.7429	18.6810	18.5069	18.5009	18.7143	2
	5	21.6896	21.4749	20.8053	21.431	21.5975	2
	6	23.037	18.154	17.9648	23.39	23.9376	1
Airplane	2	12.2125	12.2115	12.2048	12.2115	12.2231	1
	3	15.5039	15.5039	15.4653	15.4871	15.5446	1
	4	18.3121	18.3121	17.5397	18.0306	18.3402	1
	5	20.8305	20.9088	18.8862	20.5132	21.0935	1
	6	24.4089	24.4196	16.7703	22.7782	25.1055	1
Pepper	2	12.6346	12.6346	12.6251	12.6346	12.6398	1
	3	15.6887	15.6887	15.667	15.6843	15.7023	1
	4	18.5394	18.7338	18.3012	18.441	18.8129	1
	5	21.7712	21.4016	20.0285	20.0706	22.5113	1
	6	24.8297	18.2818	22.271	24.4284	25.5685	1
Baboon	2	12.2178	12.2178	12.2058	12.2178	12.2277	1
	3	15.2792	15.2792	15.213	15.2516	15.2950	1
	4	18.1227	18.1304	17.8802	18.1205	18.4594	1
	5	15.2792	20.7896	20.0949	20.538	21.0589	1
	6	23.1959	15.2791	16.5763	23.1312	23.3318	1
Scene	2	12.1446	12.3466	12.3363	12.3463	12.3553	1
	3	15.3183	15.3183	15.3135	15.2854	15.3380	1
	4	18.0118	18.0122	17.6827	17.7206	18.0523	1
	5	21.2878	20.6093	20.016	21.027	20.7145	3
	6	24.6921	24.6904	21.7677	22.4968	24.7524	1
Starfish	2	12.9682	12.9682	12.9649	12.9677	12.9690	1
	3	16.1254	16.1254	16.0434	15.7992	16.2150	1
	4	18.6202	19.0579	18.433	18.9514	19.0691	1
	5	22.3432	21.8090	21.5018	18.9726	22.7734	1
	6	25.9227	25.9317	23.9174	24.8966	25.1927	3
Zebra	2	12.1279	12.1279	12.0997	12.1235	12.1279	1
	3	15.0804	15.0804	14.9954	15.0764	15.0974	1
	4	17.8964	17.8857	17.5738	17.8644	17.8739	3
	5	21.1936	20.4511	19.3559	20.2401	21.2891	1
	6	24.4155	24.4509	19.9767	23.4194	24.801	1
Butterfly	2	12.4273	12.4273	12.4211	12.4273	12.4273	1
	3	15.6523	15.6524	15.5933	15.6053	15.6534	1
	4	18.6376	18.6376	17.261	18.6351	18.7785	1
	5	21.3965	21.3965	19.4727	20.3154	21.4259	1
	6	24.8441	24.8489	22.1475	23.8042	24.878	1

5.3 Experiment settings

The results obtained using MSA-based multilevel thresholding for image segmentation were compared with the algorithms using the WOA [5], BA [29], GWO [12], and FPA [28]. All the algorithms were run 30 times for each test image to ensure the credibility of the statistics. The parameter settings for all the algorithms are presented as follows. All the algorithms were run on a computer with an AMD Athlon (tm) II X4 640 processor and 4 GB of RAM using MATLAB R2012a.

Table 2 Best threshold values obtained from the algorithms for all test images

Images	K	WOA	GWO	FPA	BA	MSA
Man	2	112,211	92,172	143,211	83,186	85,172
	3	56,135,208	63,121,181	66,121,207	78,135,171	67,104,169
	4	44,110,170 200	40,66,138 177	67,101,162 226	41,80,156 226	80,124,170 233
	5	53,115,157 171,223	65,101,160 187,233	88,128,174 218,233	38,81,135 171,213	38,92,121 158,178
	6	73,104,149 184,208,229	39,83,135 161,205,236	76,103,168 188,217,236	35,86,156 179,222,235	37,67,88 114,129,172
Airplane	2	85,177	78,180	64,136	69,170	91,156
	3	53,97,139	50,89,171	91,145,195	60,139,168	78,134,166
	4	42,84,128 177	72,101,128 185	70,109,182 207	47,84,157 196	56,92,112 153
	5	58,83,111 147,190	160,232,18 72,174	65,104,136 182,208	62,121,157 190,212	49,93,101 143,184
	6	130,182.231 20,70,140	45,66,84 131,156,201	44,75,95 145,181,200	35,66,81 109,157,195	59,78,90 124,147,169
Pepper	2	74,146	72,134	85,159	78,143	69,157
	3	57,121,168	43,86,152	80,135,198	36,98,159	73,116,168
	4	48,101,165 194	48,74,123 191	49,89,128 169	37,89,127 175	58,111,142 197
	5	33,97,129, 149,201	41,73,100 158,208	46,99,126 155,178	45,77,98 147,205	59,70,110 142,184
	6	33,60,93 165,185,200	86,229,8 67,116,166	35,45,68 100,141,180	21,42,81 109,133,180	37,88,110 156,172,191
Baboon	2	170,300	61,138	122,221	61,126	90,131
	3	92,135,168	46,99,153	72,112,159	38,82,166	37,101,157
	4	25,75,106 169	26,74,125 156	36,106,154 189	27,70,103 149	37,102,160 189
	5	28,66,88 113,181	108,230,8 84,145	23,61,98 156,195	17,68,89 120,164	30,81,122 157,184
	6	21,39,58 84,122,167	22,53,82 128,159,177	10,37,65 139,163,193	29,52,79 111,164,185	23,54,78 122,158,190
Scene	2	91,162	92,163	113,168	103,171	83,171
	3	73,120,170	86,132,165	74,122,164	78,127,177	77,141,200
	4	81,114,152 199	67,92,138 194	31,76,115 187	40,78,126 182	59,98,129 158
	5	61,95,124 154,171	129,237,16 93,166	83,115,133 175,202	25,77,105 148,182	28,42,89 161,192
	6	126,239,2 239,5,125	93,165,240 15,94,165	31,52,70 104,151,190	34,63,105 136,168,189	39,65,91 119,159,201
Starfish	2	92,170	92,174	86,167	85,161	92,161
	3	63,127,213	75,131,183	65,154,185	92,172,215	91,130,203
	4	65,94,141, 176	59,94,151 207	96,132,168 197	62,85,120 164	50,81,117 186
	5	89,166,251 16,156	92,167,254 13,157	52,89,156 184,203	61,136,167 188,225	34,85,128 187,222
	6	38,65,89 157,174,200	155,251,14 251,15,256	33,58,80 141,195,222	78,102,120 157,196,224	44,88,107 129,149,215
Zebra	2	110,164	99,161	33,232	92,158	107,142
	3	94,151,177	113,177,218	85,135,171	108,153,211	74,129,165
	4	85,124,158 202	163,254,36 157	86,141,154 187	80,144,174 207	128,170,199 235
	5	88,126,153 181,222	77,95,133 168,207	84,118,142 199,223	91,153,173 197,240	82,96,132 167,210
	6	133,178,241 32,92,150	165,253,44 97,146,180	91,127,150 162,190,234	81,113,137, 169,189,236	82,103,163 185,223,238

Table 2 (continued)

Images	K	WOA	GWO	FPA	BA	MSA
Butterfly	2	95,190	85,165	100,162	100,172	83,167
	3	76,118,169	88,167,217	94,170,213	80,171,213	78,140,198
	4	44,87,129, 167	88,140,171, 217	100,135,193 220	75,149,180 214	54,87,176 215
	5	93,167,251 17,93	48,99,120 145,198	64,76,126 165,206	71,91,130 168,217	57,79,106 160,188
	6	94,249,23 68,112,172	102,242,23 85,164,235	71,87,128 165,180,234	35,68,98, 160,194,221	54,75,113, 143,163,193

Table 3 Average execution time for WOA, GWO, FPA, BA, and MSA

Images	K	Execution Times					
		WOA	GWO	FPA	BA	MSA	Rank
Man	2	3.670863	3.448892	3.988095	3.131518	3.40444	2
	3	4.465535	4.12334	4.12129	3.667006	3.3485	1
	4	4.685268	4.175399	4.241257	4.253551	2.9371	1
	5	4.659986	4.284073	4.367275	3.88244	4.34483	3
	6	5.329206	4.182162	4.48593	4.223635	4.02116	1
Airplane	2	3.487193	3.919529	3.617176	3.012253	4.11772	5
	3	4.08142	5.150806	3.816773	3.29666	3.253075	1
	4	3.913284	5.164029	3.964833	3.546674	2.856625	1
	5	4.386073	4.175707	4.198728	3.5552	2.890941	1
	6	4.931973	3.2004	4.225263	3.638575	3.266104	2
Pepper	2	3.462353	3.125778	3.729245	2.961756	4.31241	5
	3	4.091961	4.187989	3.978572	3.430852	2.949055	1
	4	4.340664	4.168819	4.05002	3.706896	2.988203	1
	5	4.614163	4.372566	4.207559	3.828993	3.044517	1
	6	5.047521	4.200213	4.333037	4.210472	3.017497	1
Baboon	2	3.618338	4.002072	3.725428	3.002515	4.23175	5
	3	3.779759	4.311182	3.810672	3.560505	2.918715	1
	4	4.36631	4.131047	4.028702	3.668449	5.25713	5
	5	4.697688	3.164794	4.262683	3.750542	2.995468	1
	6	4.896725	4.222328	4.213316	4.041591	3.006378	1
Scene	2	3.577844	4.432115	3.804555	3.188479	3.220119	2
	3	3.910137	4.170601	3.988137	3.198707	3.73633	2
	4	4.509571	4.269453	4.193845	3.579683	2.979141	1
	5	4.68619	4.173255	4.262395	3.752403	3.221293	1
	6	5.490817	5.188378	4.246695	3.919293	3.223279	1
Starfish	2	3.820036	4.434566	3.846841	3.073652	3.35736	2
	3	3.852187	4.039655	4.01888	3.300292	2.945067	1
	4	4.631377	4.165514	4.182721	3.782456	4.170151	2
	5	5.105683	4.176874	4.268152	3.456672	3.038873	1
	6	5.68723	4.202795	4.355798	3.750132	3.33344	1
Zebra	2	3.707141	3.339728	3.803877	3.047237	3.239583	1
	3	3.706577	3.160429	3.988334	3.332276	2.831891	1
	4	4.048737	4.461268	4.004779	3.538828	4.34538	4
	5	4.722355	3.174109	4.114894	3.531405	3.359156	2
	6	5.165684	4.168981	4.284822	3.811084	3.304698	1
Butterfuly	2	3.403841	4.041209	3.819823	3.08736	3.012461	1
	3	3.509875	4.228153	3.875453	3.279045	2.89213	1
	4	4.043335	4.474957	4.110491	3.547001	4.3508	1
	5	4.59514	4.269676	4.183948	3.564486	4.17714	2
	6	5.01741	4.169025	4.312276	3.789602	3.10434	1

Table 4 PSNR metrics for the WOA, GWO, FPA, BA, and MSA

Images	K	PSNR Values					
		WOA	GWO	FPA	BA	MSA	Rank
Man	2	53.7726	53.2051	52.9055	51.2809	53.2425	2
	3	54.3707	53.4659	56.1219	54.8625	56.7725	1
	4	53.5451	56.9898	57.5505	56.2640	57.1124	2
	5	54.6784	55.6627	55.3788	55.3784	55.8923	1
	6	55.2505	53.6396	55.3159	54.9181	55.8222	1
Airplane	2	66.5131	64.8908	63.4145	58.3139	63.5843	3
	3	65.0480	66.3855	67.8344	67.8274	67.2994	3
	4	64.5227	65.3550	65.4851	65.3573	65.0583	4
	5	74.2541	74.2806	66.5102	62.9149	75.8773	1
	6	66.5878	68.9624	67.8294	67.8194	67.7704	3
Pepper	2	57.2505	57.7542	57.5820	56.8069	57.9018	1
	3	58.4293	58.1319	56.9017	59.8393	59.0871	2
	4	58.1319	61.0690	59.9862	60.6185	60.3525	2
	5	60.5243	59.2694	61.5043	64.1002	60.9427	3
	6	59.9844	60.8243	62.4818	60.6151	62.6164	1
Baboon	2	58.6858	58.2691	59.2628	56.6931	59.7869	1
	3	63.7893	65.3874	63.7806	63.5322	65.9174	1
	4	61.7041	62.1236	63.0405	69.6367	68.9698	2
	5	72.9392	60.7170	70.4242	64.8301	71.5022	2
	6	67.9216	68.8279	67.6497	69.3415	71.2438	1
Scene	2	53.8531	55.3725	54.1780	55.2700	56.5951	1
	3	56.4279	56.4274	56.7021	57.0626	57.7968	1
	4	57.4868	74.1238	56.3608	57.8096	57.6879	2
	5	74.1241	55.3975	56.3785	64.6414	66.6272	2
	6	81.5320	74.1113	57.1272	57.5625	58.9090	2
Starfish	2	54.2938	55.6718	58.7019	56.9892	59.0406	1
	3	56.2416	56.3619	58.2857	56.1241	60.1509	1
	4	56.8627	58.5771	59.4716	58.4476	59.7764	1
	5	56.3619	59.4564	62.2667	60.5812	68.9798	1
	6	63.4761	63.8652	59.4495	61.6943	63.3476	3
Zebra	2	54.3997	53.8870	53.8774	52.7959	54.498	1
	3	62.0994	55.3611	52.6135	57.2759	57.7785	2
	4	54.1490	71.9094	53.2185	59.6263	53.5185	4
	5	71.9203	71.5815	56.9877	55.6917	59.1082	3
	6	61.2758	71.8496	71.5559	56.6667	57.2671	4
Butterfly	2	50.8261	50.3011	50.7850	50.7849	50.9601	1
	3	51.0820	50.8983	50.8365	51.0025	51.0361	2
	4	53.9354	50.8983	50.7850	51.1158	54.5711	1
	5	79.6272	77.0979	51.5751	51.2670	79.9178	1
	6	75.3326	75.3326	56.6446	56.6446	75.5848	1

BA setting The maximum pulse intensity A was 0.9, maximum pulse frequency r was 0.5, pulse attenuation coefficient *alpa* was 0.95, and pulse frequency increase factor *gamma* was 0.05. The population size was 25 and maximum iteration number was 100.

WOA setting The probability of p was 0.5, the number of iterations and population number were 100 and 25, respectively.

Table 5 SSIM metrics for the WOA, GWO, FPA, BA, and MSA

Images	K	SSIM	Measure				
		WOA	GWO	FPA	BA	MSA	Rank
Man	2	0.8691	0.8691	0.8610	0.8389	0.8643	3
	3	0.8738	0.8738	0.8839	0.8771	0.8891	1
	4	0.8670	0.8738	0.8692	0.8845	0.8786	2
	5	0.8738	0.8759	0.8801	0.8801	0.8828	1
	6	0.8691	0.8794	0.8796	0.8774	0.8825	1
Airplane	2	0.8973	0.8973	0.8898	0.8907	0.8973	1
	3	0.8984	0.8984	0.8992	0.8992	0.8988	2
	4	0.8942	0.8935	0.8921	0.8935	0.8945	1
	5	0.8966	0.8966	0.8982	0.8972	0.8987	1
	6	0.8989	0.8973	0.8976	0.8930	0.8991	1
Pepper	2	0.8839	0.8839	0.8821	0.8843	0.8846	1
	3	0.8898	0.8889	0.8846	0.8985	0.8852	3
	4	0.8889	0.8932	0.8932	0.8933	0.8938	1
	5	0.8942	0.8942	0.8939	0.8980	0.8949	3
	6	0.8932	0.8932	0.8928	0.8941	0.8943	1
Baboon	2	0.8938	0.8930	0.8847	0.8889	0.8952	1
	3	0.8976	0.8992	0.8986	0.8985	0.8976	3
	4	0.8875	0.8843	0.8983	0.8998	0.8943	3
	5	0.8873	0.8966	0.8961	0.8951	0.8973	1
	6	0.8997	0.8998	0.8996	0.8998	0.8999	1
Scene	2	0.8681	0.8781	0.8693	0.8775	0.8826	2
	3	0.8832	0.8832	0.8829	0.8858	0.8833	2
	4	0.8875	0.8875	0.8829	0.8873	0.8875	1
	5	0.8832	0.8832	0.8829	0.8991	0.8841	2
	6	0.8835	0.8835	0.8861	0.8877	0.8890	1
Starfish	2	0.8743	0.8833	0.8828	0.8888	0.8885	2
	3	0.8860	0.8865	0.8919	0.8854	0.8955	1
	4	0.8884	0.8931	0.8942	0.8928	0.8949	1
	5	0.8865	0.8925	0.8978	0.8966	0.8998	1
	6	0.8901	0.8991	0.8943	0.8978	0.8958	3
Zebra	2	0.8770	0.8734	0.8774	0.8627	0.8420	5
	3	0.8772	0.8721	0.8721	0.8893	0.8791	2
	4	0.8753	0.8753	0.8703	0.8943	0.8803	2
	5	0.8911	0.8899	0.8821	0.8835	0.8934	1
	6	0.8864	0.8869	0.8874	0.8874	0.8893	1
Butterfuly	2	0.8246	0.8178	0.8235	0.8235	0.8277	1
	3	0.8304	0.8263	0.8248	0.8287	0.8294	2
	4	0.8659	0.8263	0.8235	0.8311	0.8730	1
	5	0.8959	0.8970	0.8393	0.8338	0.8971	1
	6	0.8991	0.8968	0.8827	0.8827	0.8993	1

FPA setting The population number was 25, number of iterations was 100, and probability of p was 0.8.

GWO setting $\overrightarrow{\alpha}$ was linearly decreased from two to zero. The population and number of iterations were the same as those for the FPA.

Table 6 $p-values$ of the Wilcoxon test over 30 runs ($p-values > 0.5$ have been highlighted in bold)

Images	K	Wilcoxon test			
		MSA VS BA	MSA VS GWO	MSA VS FPA	MSA VS FPA
Man	2	0.006369	4.56E-11	0.003758	4.56E-11
	3	0.000571	1.24E-09	6.77E-07	1.24E-09
	4	0.005824	1.21E-12	0.002001	1.21E-12
	5	0.001174	4.71E-06	**0.501144**	4.71E-06
	6	0.000149	4.26E-06	0.006097	4.26E-06
Airplane	2	0.004425	1.24E-09	0.04841	1.24E-09
	3	2.96E-06	2.47E-08	0.002755	2.47E-08
	4	1.81E-05	6.02E-11	0.004855	6.02E-11
	5	5.47E-10	**0.07647**	3.96E-08	**0.07647**
	6	1.69E-09	**0.660726**	1.31E-08	**0.660726**
Pepper	2	8.14E-07	0.001792	3.81E-07	0.001792
	3	3.32E-11	**0.376612**	1.1E-08	**0.376612**
	4	0.007956	**0.07698**	5.46E-09	**0.07698**
	5	0.000253	**0.660735**	1.85E-08	**0.660735**
	6	1.61E-10	**0.075751**	8.1E-10	0.075751
Baboon	2	2.15E-10	0.007905	3.8E-10	0.007905
	3	0.00795	1.62E-09	3.32E-11	1.62E-09
	4	**0.387093**	1.06E-06	4.2E-10	1.06E-06
	5	**0.185645**	2.6E-08	4.08E-11	2.6E-08
	6	0.002051	7.62E-06	5.49E-11	7.62E-06
Scene	2	0.027065	0.000951	4.08E-11	0.000951
	3	0.027044	0.000178	3.02E-11	0.000178
	4	**0.318237**	0.018366	4.07E-11	0.018366
	5	0.007958	9.3E-06	3.69E-11	9.3E-06
	6	**0.153667**	8.88E-06	3.02E-11	8.88E-06
Starfish	2	**0.185687**	7.04E-07	3.02E-11	7.04E-07
	3	**0.684318**	4.15E-10	5.57E-10	4.15E-10
	4	**0.185767**	3.65E-06	3.02E-11	3.65E-06
	5	**0.378963**	1.17E-09	3.02E-11	1.17E-09
	6	0.074827	9.21E-05	3.02E-11	9.21E-05
Zebra	2	0.027012	6.76E-05	3.02E-11	6.76E-05
	3	N/A	7.59E-06	3.82E-10	7.59E-06
	4	9.51E-06	2.34E-11	3.02E-11	2.34E-11
	5	0.027078	0.003183	3.02E-11	0.003183
	6	**0.245801**	4.15E-10	5.54E-10	4.15E-10
Butterfly	2	6.76E-05	2.05E-11	5.57E-10	2.05E-11
	3	**0.728262**	4.31E-08	3.02E-11	0.000239
	4	3.83E-06	2.83E-08	3.02E-11	0.012731
	5	0.001947	1.17E-09	3.02E-11	0.000446
	6	0.027086	**0.379036**	3.02E-11	**0.318304**

5.4 Segmented image quality measurements

To evaluate the segmented image quality, we used four measures: the PSNR, SSIM, computational time, and fitness function value using Eq. (1). The PSNR often served as a quality measurement between the segmented images and reference images. It provided the similarity of an image against a reference image based on the MSE of each pixel [1]. The PSNR is defined as follows:

Fig. 3 Thresholded images of man obtained by all the algorithms

$$PSNR \quad (in \ \ db) = 20\log_{10}\left(\frac{255}{RMSE}\right)$$

where *RMSE* is the root mean-squared error defined as

$$RMSE = \sqrt{\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left[I(i,j) - Seg\left(i,j\right)\right]^2}$$

where both M and N denote the sizes of the image, I denotes the original image, and *Seg* denotes the segmented images. The higher the PSNR value, the better the results of the segmented images.

The SSIM is often used to compute the similarity of the original image and segmented image. The mathematical model of the SSIM is defined as follows:

Fig. 4 Thresholded images of airplane obtained by all the algorithms

$$SSIM \quad (I, Seg) = \frac{\left(2\mu_I\mu_{Seg} + c_1\right)\left(2\sigma_{I,Seg} + c_2\right)}{\left(\mu_I^2 + \mu_{Seg}^2 + c_1\right)\left(\sigma_I^2 + \sigma_{Seg}^2 + c_2\right)}$$

where μ_I represents the mean intensity of image I and μ_{Seg} represents the mean intensity of image *Seg*. σ_I and σ_{Seg} are the standard deviation of image I and image *Seg*, respectively. $\sigma_{I,\,Seg}$ is a coefficient that denotes the covariance between image I and image *Seg*. c_1 and c_2 are constants. The higher the value of the SSIM, the better the result of the segmented image.

Fig. 5 Thresholded images of pepper obtained by all the algorithms

5.5 Experimental results

In this subsection, we present the segmented results obtained using the proposed algorithm and four state-of-the-art algorithms based on Kapur's entropy in Tables 1, 2, 3, 4, 5, and 6. Table 1 lists the number of thresholds ($k = 2, 3, 4, 5, 6$) and the best fitness value obtained by all the algorithms. For the convenience of reading, we also provide the ranking of the fitness values. Table 1 shows clearly that the MSA outperformed all the algorithms for all the test images when $K = 2$ and $K = 3$. When $K = 4$, the MSA failed to obtain the best fitness value for all images except the images Man and Zebra; however, the MSA still outperformed the other algorithms. For $K = 5$, for the image Man, the result obtained using WOA was slightly better

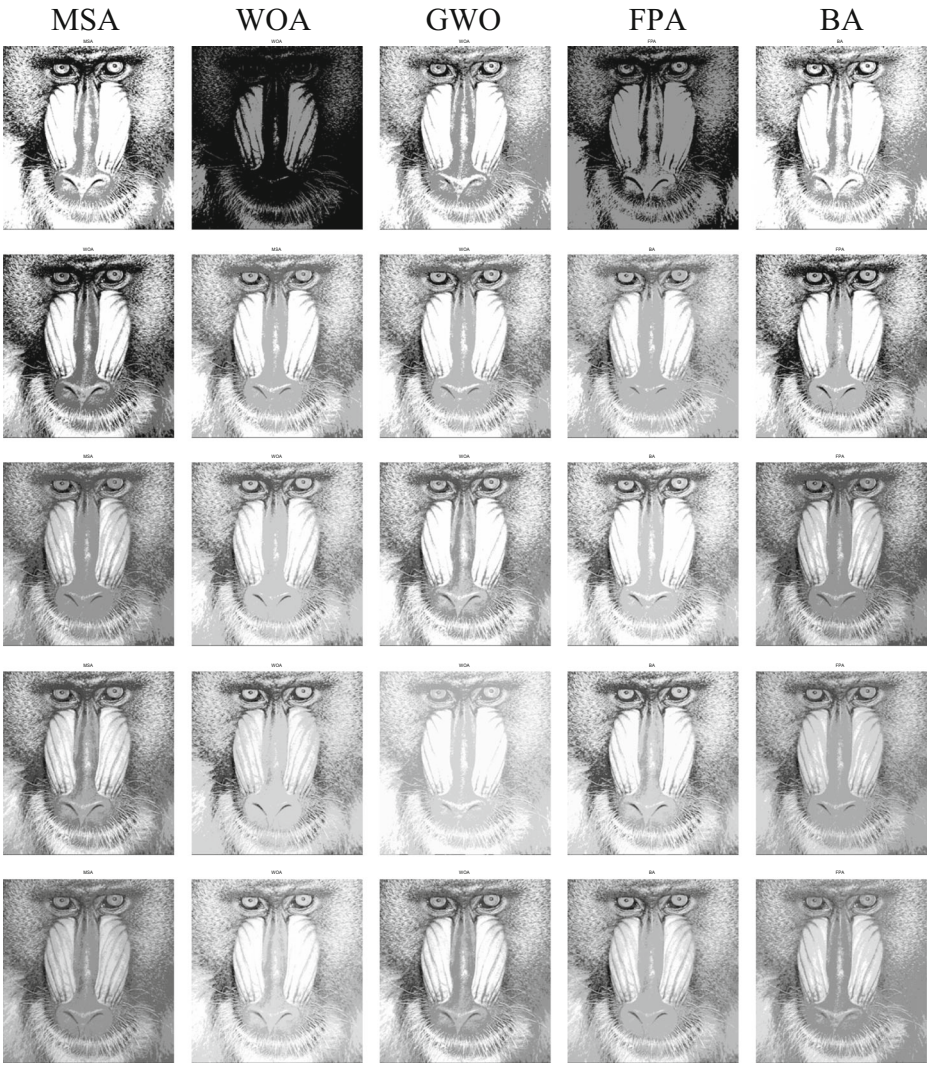


Fig. 6 Thresholded images of baboon obtained by all the algorithms

than using the MSA, whereas the MSA obtained a higher value than the other three algorithms. For the image Scene, the MSA ranked third after the WOA and BA. However, for the remaining images, the MSA outperformed all the other algorithms. When $K = 6$, the WOA and GWO obtained better results than the MSA for the image Starfish; however, for the other seven images, the MSA always determined the highest value compared with the other four algorithms. From Table 1, we conclude that the MSA outperformed the other four algorithms for almost all the test images with various threshold values.

Tables 2 and 3 report the best threshold values obtained from the algorithms for all the test images and the corresponding execution times, respectively. The results in Table 3 demonstrate

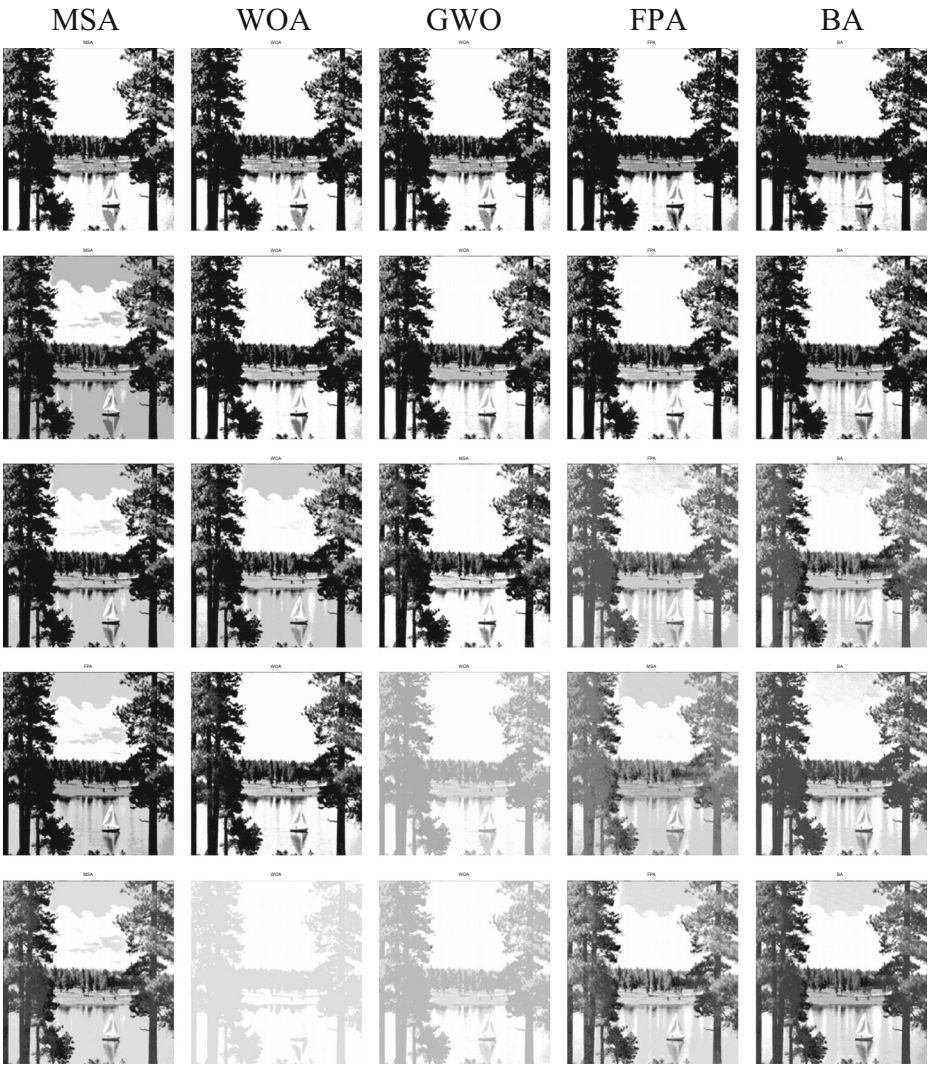


Fig. 7 Thresholded images of scene obtained by all the algorithms

that the average execution time of the MSA was less than that of the other algorithms for most of the test images under different thresholds.

Table 4 reports the PSNR values under different thresholds obtained by the five algorithms. As shown in Table 4, the MSA obtained the highest results in the majority of the test cases. There were 40 PSNR values in total for each algorithm, and the MSA obtained the best results for half of all the PSNR values when compared with the other four algorithms. Additionally, the MSA ranked second in 11 cases for the PSNR values. This demonstrates the superior ability of the MSA.

Table 5 illustrates the SSIM standard under various thresholds for the WOA, GWO, FPA, BA, and MSA. If the value of the SSIM was high, then the similarity between the segmented image and original image was high; that is, the SSIM is a method that measures the quality of

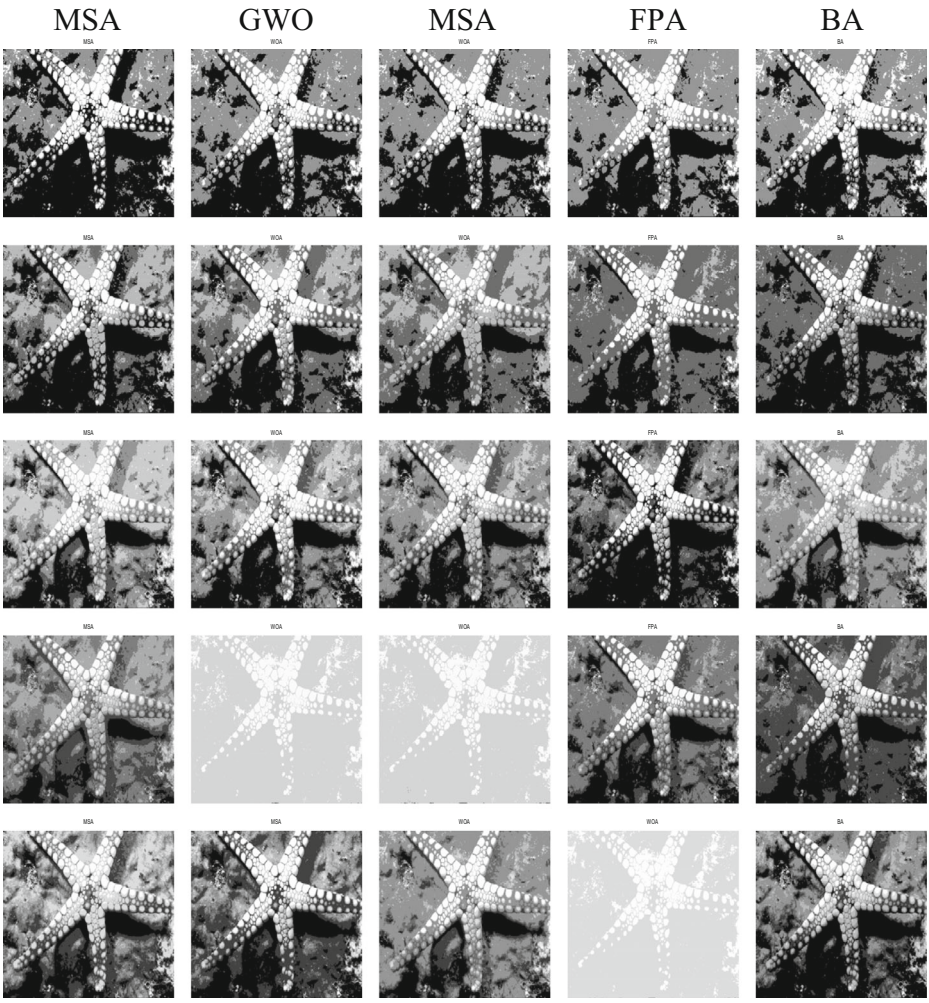


Fig. 8 Thresholded images of starfish obtained by all the algorithms

the segmented image and original image. The results reported in Table 5 indicate that the MSA always obtained the highest SSIM value or the second-best result for the 40 test cases. Hence, we conclude that the MSA-based method provided better quality segmentation compared with the WOA, GWO, FPA, and BA-based methods.

To further verify the performance of the swarm algorithms, we also conducted a statistical test: the Wilcoxon's rank-sum test was performed at a 5% significance level in our experiment [10, 30]. Generally, if $p-values < 0.5$, then this can be considered as sufficient evidence against the null hypothesis. The objective function values of the MSA method were compared with the other four algorithms. Table 6 provides the results of the $p-values$ for the Wilcoxon's rank-sum test compared the MSA and other algorithms over all the simulation cases. In Table 6, N/A means "Not Applicable," which indicates that the statistical test could not be executed because all the runs of the algorithms

Fig. 9 Thresholded images of zebra obtained by all the algorithms

determined the optimum successfully. It can be clearly observed from Table 6 that the MSA-based method provided better results than the BA, GWO, FPA, and WOA-based methods for almost all the test cases. The results demonstrate that there were significant differences between the MSA and the other four algorithms, and they prove that the MSA performed much better than the other algorithms.

The segmentation results of the MSA and the other algorithms are presented in Figs. 3, 4, 5, 6, 7, 8, 9, and 10. According to these figures, we can observe that the MSA demonstrated good segmentation results for different images under various thresholds. Additionally, these figures demonstrate that, for an increasing threshold value, the segmented images were better.

The comparison between the proposed algorithm and the other algorithms is shown in Tables 1, 2, 3, 4, 5, and 6 and Figs. 3, 4, 5, 6, 7, 8, 9, and 10. Tables 1 and 2 report the

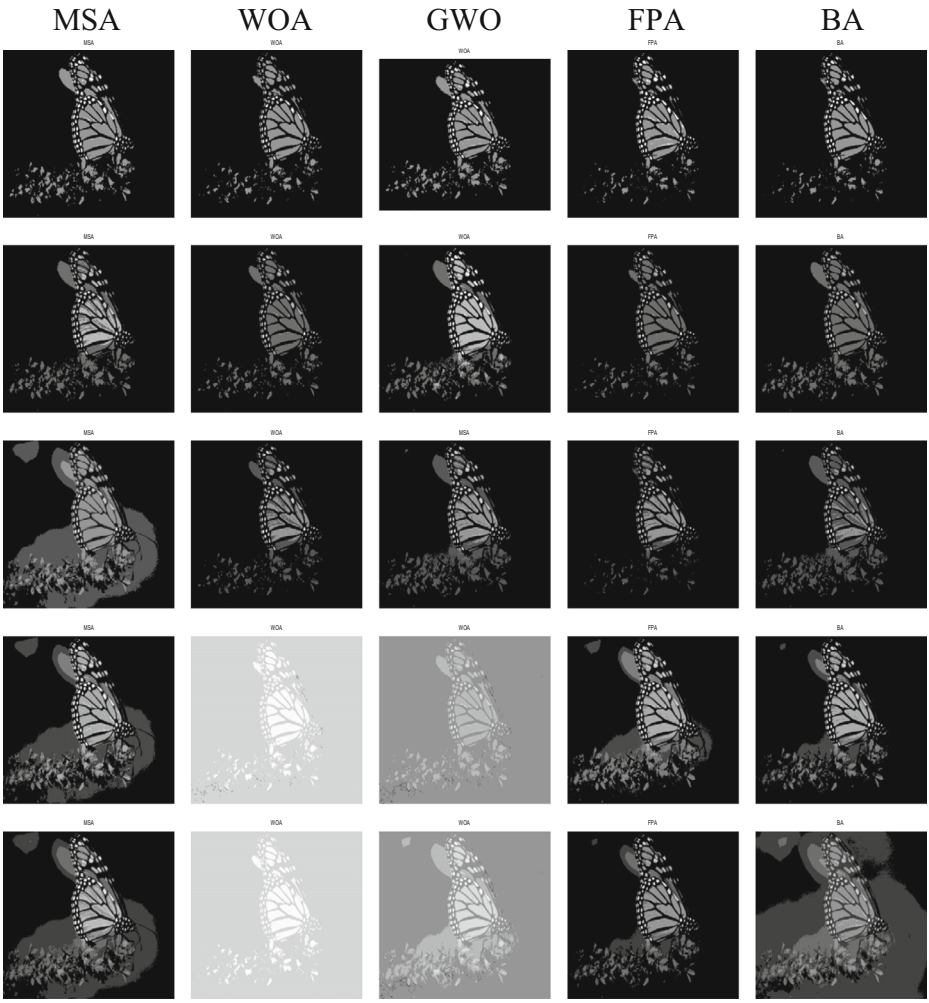


Fig. 10 Thresholded images of butterfly obtained by all the algorithms

fitness values and best threshold values, respectively, of the algorithms over 30 runs. The results in Table 1 indicate that all the algorithms performed nearly equally for $K = 2, 3, 4, 5, 6$; however, the MSA ranked in the top three for the fitness values. When $K = 3$, the WOA and GWO had nearly the same fitness function. Table 3 reports the experiments execution time. Depending on the results, the fastest algorithm was the MSA. The proposed algorithm obtained the best fitness value in most of the experiments, despite not being the fastest. The results demonstrate that the proposed algorithm had a better ability to switch between the exploration and exploitation phases than the other algorithms, and it had low complexity and high performance. The MSA has fewer control parameters than the other algorithms, which makes it more suitable for other optimization problems because the control parameters of the algorithm are likely to be more complex than the problem itself.

The values of the SSIM and PSNR are reported in Tables 4 and 5. These values indicate that the proposed method had better results in most cases. However, when $k = 2$, 4, 6 for image Zebra, the SSIM value and PSNR value were lower than those of the other algorithms because each image was considered as a different optimization problem and the randomness of the swarm approaches caused the results to vary in some cases. Figures 2, 3, 4, 5, 6, 7, 8, 9, and 10 show the segmentation results of the proposed algorithm and the other algorithms with different threshold levels. From these figures, we conclude that higher-level images contain more details than other images.

In image processing, it is often possible to manually mark the work, but it is difficult to write a complete rule for automatic processing. Sometimes there is an entire set of algorithms, but there are too many parameters, and it is too tedious to manually adjust and determine the correct parameters. Hence, we can use the machine learning method to extract a certain number of features and manually mark a batch of results, and then use machine learning to determine a set of automatic judgment criteria. Machine learning is more effective for developing such software.

In this section, we used different approaches to measure the segmented quality of the MSA-based method and compare it with other algorithms. In terms of the objective fitness value, PSNR, SSIM, and execution times, the MSA-based method provided the best results for most test cases. Additionally, Wilcoxon's rank-sum test demonstrated the good performance of the MSA-based method. This confirms the superior ability of the MSA for image segmentation.

6 Conclusions and future work

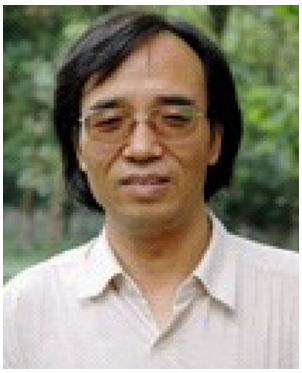
The objective of image segmentation thresholding is to obtain a good quality of segmented results without consuming a great deal of time. In our paper, we used the MSA-based Kapur's entropy method to solve image segmentation problems under different thresholds. The experimental results of the MSA-based method not only clearly demonstrated the efficiency and feasibility of this method in solving multilevel thresholding but also proved our proposed method's obvious superiority over four state-of-the-art algorithms, WOA, BA, GWO, and FPA, in terms of the PSNR, SSIM, and execution time. For future work, the MSA can be applied to solve image segmentation under higher threshold values and tested on many more images. Additionally, we aim to propose a simpler but more efficient MSA to solve image segmentation and apply it in egineering applications.

Acknowledgments This work was supported by the National Science Foundation of China under Grant No. 61463007 and the Project of the Guangxi Natural Science Foundation under Grant No. 2016GXNSFAA380264. We thank Maxine Garcia, PhD, from Liwen Bianji, Edanz Group China (www.liwenbianji.cn/ac) for editing the English text of a draft of this manuscript.

References

1. Bhandari AK, Singh VK, Kumar A, Singh GK (2014) Cuckoo search algorithm and wind driven optimization based study of satellite image segmentation for multilevel thresholding using Kapur's entropy. Expert Syst Appl 41(7):3538–3560
2. Bhandari AK, Singh VK, Singh GK, Singh GK (2014) Cuckoo search algorithm and wind driven optimization based study of satellite image segmentation for multilevel thresholding using kapur's entropy. Expert Syst Appl 41(7):3538–3560
3. Develi I (2012) A new approximation based on the differential evolution algorithm for the gaussian q-function. Int J Innov Comput Inf Control Ijicic 8(10):7095–7102
4. Duraisamy SP, Kayalvizhi R (2010) A new multilevel thresholding method using swarm intelligence algorithm for image segmentation. J Intell Learn Syst Appl 2(3):126–128
5. El Aziz MA, Ewees AA, Hassanien AE (2017) Whale optimization algorithm and moth-flame optimization for multilevel thresholding image segmentation. Expert Syst Appl 83:242–256
6. Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning, Machine Learning, 3:95-99
7. Hollander M, Wolfe DA, Chicken E (2014) Nonparametric statistical methods. Third Edition, John Wiley & Sons, Inc. Hoboken
8. Jiang Y, Tsai P, Hao Z, Cao L (2015) Automatic multilevel thresholding for image segmentation using stratified sampling and Tabu Search. Soft Comput 19(9):2605–2617
9. Kapur JN, Sahoo PK, Wong AK (1985) A new method for gray-level picture thresholding using the entropy of the histogram. Comput Vis Graph Image Process 29(3):273–285
10. Karaboga D (2005) An idea based on honey bee swarm for numerical optimization (Vol. 200). Technical report-tr06, Erciyes university, engineering faculty, computer engineering department
11. Kennedy J (2010) Particle Swarm Optimization, Encyclopedia of Machine Learning. Springer, US, 760-766
12. Khairuzzaman AKM, Chaudhury S (2017) Multilevel thresholding using grey wolf optimizer for image segmentation. Expert Syst Appl 86:64–76
13. Lai CC, Tseng DC (2004) A hybrid approach using gaussian smoothing and genetic algorithm for multilevel thresholding. Int J Hybrid Intell Syst 1(3–4):143–152
14. Liu Y, Zhang X, Cui J, Wu C, Aghajan H, Zha H (2010) Visual analysis of child-adult interactive behaviors in video sequences. In Virtual Systems and Multimedia (VSMM) International Conference on, IEEE pp 26–33
15. Liu Y, Cui J, Zhao H, Zha H (2012) Fusion of low-and high-dimensional approaches by trackers sampling for generic human motion tracking. In Pattern Recognition (ICPR) International Conference on, IEEE pp 898–901
16. Liu Y, Nie L, Han L, Zhang L, Rosenblum DS (2015) Action2Activity: recognizing complex activities from sensor data. In IJCAI, pp 1617–1623
17. Liu Y, Nie L, Liu L, Rosenblum DS (2016) From action to activity: Sensor-based activity recognition. Neurocomputing 181:108–115
18. Liu L, Cheng L, Liu Y, Jia Y, Rosenblum DS (2016) Recognizing Complex Activities by a Probabilistic Interval-Based Model. In AAAI, pp 1266–1272
19. Lu Y, Wei Y, Liu L, Zhong J, Sun L, Liu Y (2017) Towards unsupervised physical activity recognition using smartphone accelerometers. Multimed Tool Appl 76(8):10701–10719
20. Maitra M, Chatterjee A (2008) A hybrid cooperative–comprehensive learning based pso algorithm for image segmentation using multilevel thresholding. Expert Syst Appl 34(2):1341–1350
21. Mohamed AAA, Mohamed YS, El-Gaafary AA, Hemeida AM (2017) Optimal power flow using moth swarm algorithm. Electr Power Syst Res 142:190–206
22. Otsu N (2007) A threshold selection method from gray-level histograms. IEEE Trans Syst Man Cybern 9(1):62–66
23. Pal NR, Pal SK (1993) A review on image segmentation techniques. Pattern Recogn 38(9):1277–1294
24. Pun T (1980) A new method for grey-level picture thresholding using the entropy of the histogram. Signal Process 2(3):223–237
25. Sarkar S, Sen N, Kundu A, Das S, Chaudhuri SS (2013) A differential evolutionary multilevel segmentation of near infra-red images using Renyi's entropy. In Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) Springer, pp 699–706
26. Shore J, Johnson R (1980) Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy. ieee trans. inform. theory it-26, 26-37. IEEE Trans Inf Theory 26(1):26–37
27. Storn R, Price K (1997) Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11(4):341–359
28. Wang R, Zhou Y, Zhao C, Wu H (2015) A hybrid flower pollination algorithm based modified randomized location for multi-threshold medical image segmentation. Biomed Mater Eng 26(Suppl 1):S1345
29. Yang XS (2010) A new metaheuristic bat-inspired algorithm. Comput Knowl Technol 284:65–74

30. Yang XS (2013) Multiobjective firefly algorithm for continuous optimization. Eng Comput 29(2):175–184
31. Yen JC, Chang FJ, Chang S (1995) A new criterion for automatic multilevel thresholding. IEEE Trans Image Process 4(3):370–378
32. Yin PY (1999) A fast scheme for optimal thresholding using genetic algorithms. Signal Process 72(2):85–95



Yongquan Zhou , Ph.D & Prof, received the MS degree in computer science from Lanzhou University, Lanzhou, China, in 1993 and the PhD degree in computation intelligence from the Xiandian University, Xi'an, China, in 2006. He is currently a professor at Guangxi University for Nationalities. His research interests include computation intelligence, neural networks, and intelligence information processing.

Xiao Yang , M.S., His current research interests are computational intelligence, image processing.