ICIC Express Letters
Part B: Applications
Volume **7**, Number **10**, October **2016**

ICIC International ©2016 ISSN 2185-2766
pp. **2253–2259**

FLAME IMAGE SEGMENTATION USING THE DYNAMIC ADAPTATION CUCKOO SEARCH ALGORITHM BASED ON THE 2D MAXIMUM BETWEEN-CLUSTER VARIANCE

Xiaolin Zhang*, Chong Zhang and Tao Yang

School of Information Engineering
Inner Mongolia University of Science and Technology
No. 7, Inner Mongolia Arding Street, Baotou 014010, P. R. China
*Corresponding author: zhangxl@imust.cn

Received March 2016; accepted June 2016

Abstract. *Existing flame image segmentation methods are generally defective, have poor antinoise performance, and have a long computing time. This study proposes a flame image segmentation method using the dynamic adaptation cuckoo search algorithm based on the 2D maximum between-cluster variance to meet the real-time flame status monitoring requirements of power boilers. With the 2D maximum between-cluster variance as the thresholding rule, the antinoise performance of the algorithm is enhanced. The dynamic variance policy of differential evolution and the detection probability of dynamic changes were modified based on the update policy of cuckoo search (CS) algorithm, thereby enhancing the local search of the algorithm, balancing the mining and development of the algorithm, and reducing the segmentation time. Experimental data show that the proposed method outperforms other flame image segmentation methods that are based on particle swarm optimization and CS in terms of time and accuracy.*
Keywords: Flame image segmentation, 2D maximum between-cluster variance, Cuckoo search algorithm

1. **Introduction.** The image processing and recognition system is a key technology in monitoring the state of the furnace flame, of which flame image segmentation is a critical step that separates the flame from the background and directly affects the accuracy of flame status recognition. Various thresholding methods have been applied to image segmentation. Among them, the bionics algorithm is a hot direction.

In 2009, Professor Yang from the University of Cambridge and Deb proposed the cuckoo search (CS) algorithm [1], which could effectively solve the optimization problem by simulating the parasitic cuckoo brood. Given its simple structure, less control parameters, and strong search ability, the CS algorithm is considered more potent and efficient in many optimization problems than the GA and PSO algorithms [2,3]. However, the CS algorithm has a weak local search ability and slow convergence at the later stage. Srivastava et al. [4] introduced the tabu search (TS) algorithm while using Lévy flights for local search, by which the current optimum solution was stored in the queue for TS to avoid local optimization. Li and Yin [5] introduced orthogonal learning mechanisms that followed the preferred random walk to improve the local search algorithm; Li and Yin [6] introduced the modified adaptive CS to balance the mining and development of the CS algorithm. The aforementioned improvements have improved the performance of the CS algorithm, but none of them have been proposed specifically for flame images. This study proposes a 2D Otsu-based dynamic variation cuckoo segmentation (DACS) for flame images according to the characteristics of furnace flame images. This algorithm demonstrates two improvements in the following aspects: (1) using the 2D maximum between-cluster variance as the fitness function, the antinoise performance of the algorithm is enhanced; and (2) according to differential evolution (DE), the update policy of the CS algorithm

is replaced by the dynamic variance policy based on the current optimal and suboptimal values and on the detection probability of dynamic changes p_a. The nests are updated or eliminated by random walk to enhance the local search and convergence rate of the algorithm and to balance its mining and development. The organizational structure of the article is as follows: Chapter 2 introduces the 2D maximum between-cluster variance, Chapter 3 introduces the dynamic adaptation cuckoo search (DACS) algorithm, Chapter 4 provides the experimental results and analysis of the algorithm proposed, and finally, Chapter 5 provides the conclusions.

2. **2D Maximum Between-Cluster Variance.** The 2D maximum between-cluster variance does not only use the grayscale image pixel but also takes advantage of between-pixel space-related information, so it has strong antinoise performance. Figure 1 shows the 2D histogram of the grayscale–neighborhood average.

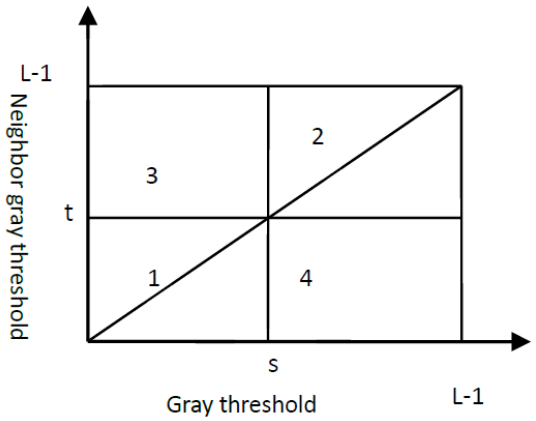


FIGURE 1. 2D histogram of the grayscale–neighborhood average

In Figure 1, the horizontal coordinate denotes the image grayscale, whereas the longitudinal coordinate denotes the neighborhood average. We assumed that P_{ij} is the joint probability density in the 2D histogram with the grayscale and neighborhood average defined as i and j, respectively. P_{ij} is defined as $P_{ij} = \frac{R_{ij}}{M \times N}$, where $(i, j = 0, 1, \ldots, L-1)$, $0 \leq P_{ij} \leq 1 \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} p_{ij} = 1$, and R_{ij} represents the occurrence of the binary system (i, j). Using the threshold (s, t), the image is divided into four rectangular regions, namely, Areas 1, 2, 3, and 4, of which Areas 1 and 2 represent the background and target areas, respectively, whereas Areas 3 and 4 represent the edge and noise of the image, respectively. Areas 3 and 4 are generally ignored in calculating the 2D maximum between-cluster variance. The probability for any pixel to be assigned to Areas 1 or 2 is computed as follows:

$$P_1 = \sum_{i=0}^{s} \sum_{j=0}^{t} p_{ij} \quad \text{and} \quad P_2 = \sum_{i=s+1}^{L-1} \sum_{j=t+1}^{L-1} p_{ij} \tag{1}$$

Definition 2.1. *The 2D maximum between-cluster variance is defined as follows:*

$$\varphi(s+t) = P_1 \left[(\mu_{0x} - \mu_{zx})^2 - (\mu_{0y} - \mu_{zy})^2 \right] + P_2 \left[(\mu_{1x} - \mu_{zx})^2 - (\mu_{1y} - \mu_{zy})^2 \right] \tag{2}$$

where μ_{0x}, μ_{0y}, μ_{1x}, μ_{1y}, μ_{zx}, and μ_{zy} denote the average grayscale of Area 1, the neighborhood average of Area 2, the average grayscale of Area 2, the neighborhood average of the entire image, the average grayscale of the entire image, and the neighborhood average, respectively.

3. Dynamic Adaptation Cuckoo Search (DACS) Algorithm.

3.1. CS algorithm.

The CS algorithm was discovered in studies on the behaviors of cuckoo species that randomly lay their eggs on the nests of other birds. The search process of this algorithm mimics the process of cuckoos finding a nest to lay their eggs.

Definition 3.1. *The basic flow of the CS algorithm is shown in Algorithm 1.*

Algorithm 1.

Begin
Initial population: n host nests X_i $(i = 1, 2, \ldots, n)$;
Calculating the fitness: F_i $(i = 1, 2, \ldots, n)$;
While (the stop condition is unsatisfied)
Using Levy flight, generate a new solution X_i;
Calculate the fitness F_i *of the new solution* X_i;
Select a candidate solution X_j;
If $(F_i > F_j)$
Replace the candidate with a new solution;
End
Abandon the unsatisfactory solution according to the detection probability p_a;
Replace the abandoned solution with a new one generated by the preferred random walk;
Retain the optimal solution.
End
End

3.2. Dynamic variance policy.

DE mainly involves three operations, namely, variation, crossing, and selection [7]. Variation [8] is the main way for DE to produce offspring individuals.

Considering the first policy update, the CS algorithm adopts Levy's random walk. Given that this method entirely depends on the random walk strategy, its convergence rate and accuracy cannot be easily guaranteed. When a flame image is processed by the rules for Otsu-threshold-based segmentation, large single peaks of the 2D maximum between-cluster variance appear in most of the flame images and approach the largest single peak from one generation to another generation, thereby accelerating the convergence rate and increasing the accuracy of the algorithm. Based on the update approach with the optimal and suboptimal values that are proposed in variation, the solution ensures that the high probability changes will appear in the vicinity of the nearest solution from the highest peak and that the late convergence rate of the algorithm will not decrease any further. The scale factor F is set to a greater value and decreases in each iteration, thereby ensuring the better global search of the algorithm at the early stage and its strong long local search at the late stage and balancing the two functions in the entire process of the algorithm, as follows:

$$X_i^t = X_i^t + F(X_{best1}^t - X_i^t) \quad \text{and} \quad X_i^t = X_i^t + F(X_{best2}^t - X_i^t) \tag{3}$$

In the CS algorithm, part of the solutions is eliminated by a fixed detection probability p_a and is updated based on the elimination. However, given that one cannot easily determine if the elimination is excellent or poor, the elimination-based update cannot further strengthen the local search or expand the global search of the algorithm. As for the CS algorithm, after the population size is fixed, the balance between the global random search and the local search will be dominated by p_a. Therefore, setting a fixed value to p_a does not present an excellent method in the CS algorithm.

The solution is as follows. (1) Complement the first location update. Although the first location update by the random walk based on the optimal and suboptimal values can ensure the sound evolution of the solutions as the walk range is expanded by the

scale factor, this update still has a poor search ability. Using the variation operator for random walk, the search capability of the algorithm is further enhanced, thereby allowing the algorithm to find the optimum solution within a short period, as follows:

$$X_i^t = X_{r1}^t + F\left(X_{r2}^t - X_{r3}^t\right) \tag{4}$$

whereX_{r1}^t, X_{r2}^t, and X_{r3}^t represent three random solutions of the tth generation.

(2) Ensure that the algorithm has a strong global search capability at the early stage and a better local search capability at the later stage. The detection probability of dynamic changes p_a quantitatively obtains the solutions that are eliminated at the early stage on the high side and those at the late stage on the low side, as follows:

$$p_a = 1.1 - \exp\left(-(t/t_{\max})^{0.6}\right) \tag{5}$$

Definition 3.2. *The DACS algorithm is shown in Algorithm 2.*

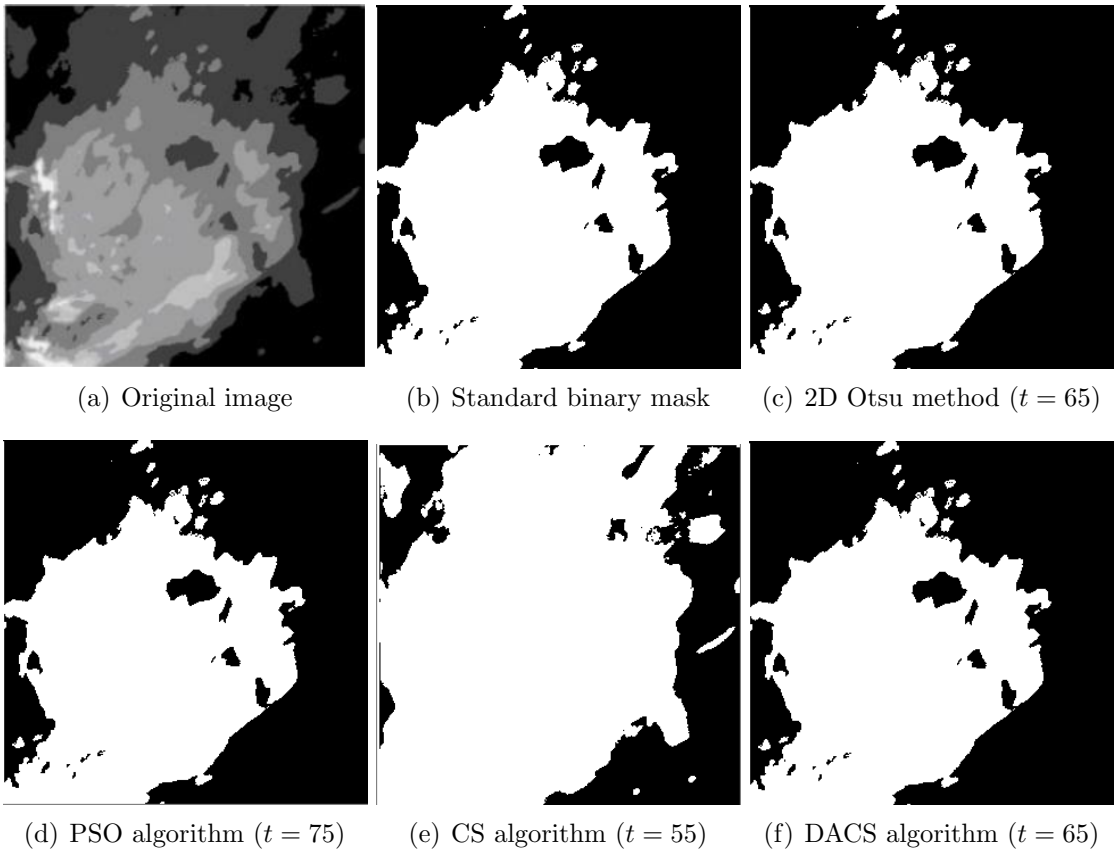
Algorithm 2.

Begin
Initial population: n host nests X_i $(i = 1, 2, \ldots, n)$*;*
Calculating the fitness: F_i $(i = 1, 2, \ldots, n)$*;*
While (the stop condition is unsatisfied)
Using the Formula (3), generate two new solutions X_{i1}*,* X_{i2}*;*
Calculate the fitness F_{i1} *and* F_{i2} *of the new solutions;*
If $(F_{i1} <= F_{i2})$
$F_{i1} = F_{i2}$*;*
$X_{i1} = X_{i2}$*;*
End
Select a candidate solution X_j*;*
If $(F_{i1} > F_j)$
Replace the candidate with X_{i1}*;*
End
Abandon the unsatisfactory solution according to the detection probability p_a*;*
Replace the abandoned solution with a new one generated by the Formula (4);
Retain the optimal solution.
End
End

4. **Experimental Results and Analysis.** To verify the flame image segmentation performance of the DACS algorithm, the test environment for the simulation is set as follows: The algorithm is implemented using Windows 7 Intel (R) Core i3-2348M CPU, 2.30 GHz, 4 G memory, and Microsoft VS2010 VC++ and opencv2.9.10.

4.1. **Results of flame image segmentation.** In this study, the flame images obtained are used for the segmentation test by 2D Otsu, CS, PSO, and DACS to verify the performance of the algorithms. Figure 2 shows the image segmentation results. In these figures, (a) represents the original image, (b) represents the standard binary mask, and (c), (d), (e), and (f) represent the results of image segmentation by 2D Otsu, PSO, CS, and DACS, respectively.

Figure 2 shows that, unlike DACS, neither PSO nor CS can accurately segment the flame image. Table 1 presents the time that each of the aforementioned algorithms spends on segmenting the images (Figure 2). The 2D Otsu method can accurately segment the image, but consumes much time. The experimental data show that the proposed method can effectively segment the flame images.



(a) Original image (b) Standard binary mask (c) 2D Otsu method ($t = 65$)

(d) PSO algorithm ($t = 75$) (e) CS algorithm ($t = 55$) (f) DACS algorithm ($t = 65$)

FIGURE 2. Image segmentation results

TABLE 1. Segmentation time of the image in Figure 2

	2D Otsu method	PSO	CS	DACS
CPU run time (ms)	30,144	1,907	1,548	1,073

TABLE 2. ME of the image in Figure 2

	2D Otsu	PSO	CS	DACS
ME	0.002034	0.036704	0.029852	0.002034

4.2. **Image segmentation analysis.** As the segmentation standard, the misclassification error (ME) can directly reveal the proportion of background pixels that are misclassified into the foreground region, calculated as follows:

$$ME = 1 - \frac{\sum(Bo \cap Bt) + \sum(Fo \cap Ft)}{\sum Bo + \sum Fo} \tag{6}$$

where Bo and Fo represent the background and standard binary masks (ground truth) in the target area of the image, respectively, Bt and Ft represent the background region and target area of the image after segmentation, respectively, $\cap$ denotes the crossing operation, and Σ denotes the operation of counting the number of pixels in each part. We observe artificial standard binary mask images in the actual segmentation and empirically show that a better image segmentation leads to a larger intersection (target or background) of the image before and after segmentation. Therefore, the ME value must be small. Table 2 shows the ME values of the images (Figure 2) by each of the aforementioned algorithms. The 2D Otsu and DACS have the smallest ME values.

4.3. **Convergence analysis.** Algorithm convergence is a basic indicator for measuring algorithm performance. The number of iterations is set to 100, whereas the iteration in each experiment is set to 30 times to analyze and compare the algorithm performance further. Figures 3 and 4 show the convergence performances of CS and DACS, respectively. CS is trapped into the local minima in the 15th to 20th iterations, but rapidly jumps out later. However, the convergence rate of CS obviously slows down at the later stage, which seriously affects its performance. Although DACS is trapped slightly longer than CS, its convergence rate at the later stage is accelerated to some extent, thereby significantly reducing the number of iterations.

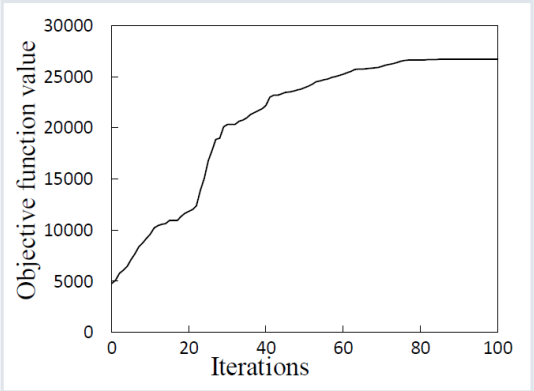


FIGURE 3. Convergence of the CS

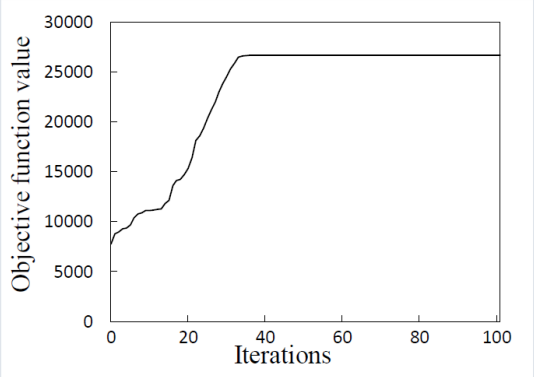


FIGURE 4. Convergence of the DACS

5. **Conclusions.** This study proposes a 2D Otsu DACS-based flame image segmentation method according to the characteristics of furnace flame images. The antinoise performance of the algorithm is improved using the 2D maximum between-cluster variance as the thresholding segmentation rule. The update policy of CS is improved by the variation operation of DE, whereas its poor local search ability and local convergence rate are improved based on the detection probability of dynamic changes. The experimental data show that the proposed method achieves better image segmentation within a shorter period at a higher convergence rate. In the future, we can try combining different bionics algorithms, and may be able to get better results.

Acknowledgment. This work has been supported by the National Science Foundation of China (61164018). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] X. S. Yang and S. Deb, Engineering optimisation by cuckoo search, *International Journal of Mathematical Modelling and Numerical Optimisation*, vol.1, no.4, pp.330-343, 2010.

[2] L. Yu and M. Liang, A new metaheuristic cuckoo search algorithm, *Systems Engineering*, vol.30, no.8, pp.64-69, 2012.

[3] Y. Zhang, L. Wang and Q. Wu, Modified adaptive cuckoo search (MACS) algorithm and formal description for global optimization, *International Journal of Computer Applications in Technology*, vol.44, no.2, pp.73-79, 2012.

[4] P. R. Srivastava, R. Khandelwal, S. Khandelwal, S. Kumar and S. S. Ranganatha, Automated test data generation using cuckoo search and tabu search (CSTS) algorithm, *Journal of Intelligent Systems*, vol.21, no.2, pp.195-224, 2012.

[5] X. T. Li and M. H. Yin, Parameter estimation for chaotic systems using the cuckoo search algorithm with an orthogonal learning method, *Chinese Physics B*, vol.21, no.5, pp.113-118, 2012.

[6] X. T. Li and M. H. Yin, Modified cuckoo search algorithm with self adaptive parameter method, *Information Sciences*, vol.298, pp.80-97, 2015.

[7] S. Das, A. Abraham, U. K. Chakraborty et al., Differential evolution using a neighborhood-based mutation operator, *IEEE Trans. Evolutionary Computation*, vol.13, no.3, pp.526-553, 2009.

[8] G. Liu and Y.-X. Li, Novel oppositional differential evolution algorithm based on theory of molecular motion, *Journal of Chinese Computer Systems*, vol.33, no.1, pp.115-120, 2012.