Multilevel Image Thresholding Selection Using the Modified Seeker Optimization Algorithm

Ivona BRAJEVIC
Faculty of Mathematics
University of Belgrade
Studentski trg 16
SERBIA
ivona.brajevic@googlemail.com

Milan TUBA
Faculty of Computer Science
Megatrend University Belgrade
Bulevar umetnosti 29
SERBIA
tuba@ieee.org

Abstract: - Multilevel thresholding is one of the most popular image segmentation techniques. This paper presents a new multilevel maximum entropy thresholding method based on modified seeker optimization (MSO) algorithm. In the proposed method the thresholding problem is treated as an optimization problem and solved by using the MSO metaheuristics. Particle swarm optimization (PSO) algorithm is also implemented for comparison with the results of the proposed method. Both algorithms were tested on four sample images. Experimental results show that the MSO algorithm performs better than PSO algorithm with respect to the quality of the segmentation results, while in term of execution time the PSO is more efficient than MSO.

Key-Words: - Maximum entropy thresholding, Image thresholding, Seeker optimization algorithm, Particle swarm optimization, Swarm intelligence

1 Introduction

Image segmentation is one of the most important operations in image analysis and computer vision [1], [2], [3]. Thresholding is one of the simplest techniques for performing image segmentation and it is very useful in separating objects from background image, or discriminating objects from objects that have distinct gray-levels. Thresholding involves bi-level thresholding and multilevel thresholding. For bi-level thresholding the main objective is to determine one threshold which separates pixels into two groups, one including those pixels with gray levels above certain threshold, the other including the rest. For multilevel thresholding the main objective is to determine multiple thresholds which divide pixels into several groups. The pixels which belong to the same class have gray levels within a specific range defined by two neighboor thresholds. The global thresholding methods [4], belonging to parametric and nonparametric approaches, select thresholds by optimizing (maximizing or minimizing) some criterion functions defined from images [5], [6].

The maximum entropy thresholding (MET) had been widely used in determining the optimal thresholding in image segmentation. The maximum entropy criterion for image thresholding was first proposed by Pun, and later it was corrected and improved by Kapur [4]. Basically, the entropy thresholding considers an image histogram as a probability distribution, and then selects as an optimal threshold value that yields the maximum entropy.

Many meta-heuristic algorithms were adopted to search for the multilevel thresholds. Yin [7] presented a new method that adopts the particle swarm optimization to select the thresholds based on the minimum cross-entropy. Horng [8] applied the honey bee mating optimization (HBMO) and the artificial bee colony (ABC) algorithm to search for the thresholds using the maximum entropy criterion. In [9] the adaptation and comparison of six meta-heuristic algorithms: genetic algorithm, particle swarm optimization, differential evolution, ant colony, simulated annealing and tabu search were presented. The experimental results have shown that the genetic algorithm, the particle swarm optimization and the differential evolution outperformed the other algorithms.

Seeker optimization algorithm (SOA) is a novel swarm intelligence algorithm based on simulating the act of human searching, which has been shown to be a promising search algorithm for unconstrained function optimization [10]. The SOA results for multimodal test functions were not very satisfactory and in order to enhance its performance,

This research is supported by Ministry of Education and Science, Republic of Serbia, Project No. III-44006

ISBN: 978-1-61804-092-3

the modified seeker optimization algorithm named MSO was proposed [11]. This paper applies the MSO algorithm to search for the multilevel thresholds using the maximum entropy criterion. The PSO algorithm is implemented for purpos of comparison. Also, the exhaustive search method is conducted to derive the optimal solutions for comparison with the results generated by PSO and MSO algorithms.

The rest of the paper is organized as follows. In the Section 2 the problem of the multilevel thresholding is formulated and the objective function is presented. The Section 3 presents our proposed MSO algorithm based on maximum entropy criterion. Section 4 gives comparative results of the implemented MSO and PSO algorithms.

2 Multilevel Thresholding Problem Formulation

Let there be L gray levels in a given image I having M pixels and these gray levels are in the range $\{0,1,...L-1\}$. The multilevel thresholding problem can be configured as a k-dimensional optimization problem, for determination of k optimal thresholds $[t_1,t_2,...,t_k]$ which optimizes an objective function.

Several objective functions devoted to the thresholding have been proposed in the literature [4]. Generally, these functions are determined from the histogram of the image, denoted by $h(i)$, $i=0,1...L-1$, where $h(i)$ represents the number of pixels having the gray level i. The normalized probability at level i is defined by the ratio $P_i = h(i)/M$. One of the most popular objective function is defined by Kapur. The aim is to maximize the objective function:

$$f([t_1,t_2,...,t_k]) = H_0 + H_1 + ... + H_k \tag{1}$$

where

$$H_0 = -\sum_{i=0}^{t_1-1} \frac{P_i}{w_0} \ln \frac{P_i}{w_0}, \; w_0 = \sum_{i=0}^{t_1-1} P_i ,$$

$$H_1 = -\sum_{i=t_1}^{t_2-1} \frac{P_i}{w_1} \ln \frac{P_i}{w_1}, \; w_1 = \sum_{i=t_1}^{t_2-1} P_i ,....$$

$$H_k = -\sum_{i=t_k}^{L-1} \frac{P_i}{w_k} \ln \frac{P_i}{w_k}, \; w_K = \sum_{i=t_k}^{L-1} P_i$$

3 Modified Seeker Optimization Algorithm Based on Maximum Entropy Criterion

Seeker optimization algorithm (SOA) mimics the behaviour of human search population based on their memory, experience, uncertainty reasoning and communication with each other. It was concluded that SOA adhered to five basic principles of swarm intelligence [10]. SOA was analyzed with a challenging set of benchmark problems for function optimization. The simulation results showed that the proposed algorithm is a promising candidate of swarm algorithms for numerical function optimization. For multimodal test functions the results were not very satisfactory because it was noticed that for this type of problems SOA may be stuck at a local optimum. In order to enhance the performance of SOA, the modified seeker optimization (MSO) algorithm was developed [11]. MSO algorithm uses two search equations for producing new population: search equation of artificial bee colony (ABC) algorithm [12] and the search equation of seeker optimization algorithm. Also, MSO algorithm implements the modified inter-subpopulation learning using the binomial crossover operator.

The proposed MSO algorithm based on maximum entropy criterion tries to obtain optimum K-dimensional vector $[t_1,t_2,...,t_k]$ which can maximize Eq.(1). The details of the developed approach are introduced as follows.

Step 1. Initialize population

MSO algorithm generates a randomly distributed initial population of SN solutions or seekers t_i *(i = 1, 2, ..., SN)* with K dimensions denoted by matrix T.

$$T = [t_1,t_2,..t_{SN}] \text{ and } t_i = (t_{i,1},t_{i,2},..t_{i,K}) \tag{2}$$

where t_{ij} is the j^{th} component value that is restricted into [0,…,L-1] and the $t_{ij} < t_{ij+1}$ for all j. Each seeker t_i *(i = 1, 2, ..., SN)* is generated by:

$$t_{i,j} = t_{\min} + rand(0,1)\cdot(t_{\max} - t_{\min}) \tag{3}$$

where $t_{\min}$ and $t_{\max}$ are the minimum and the maximum gray values in the image, the rand(0, 1) is a random number between 0 and 1. In MSO algorithm, as in SOA, the total population is categorized into N subpopulations according to the indexes of the seekers. Each seeker t_i, beside of its current position and its objective function value, has the following attributes: the personal best position

ISBN: 978-1-61804-092-3

p_{ibest} so far and the neighborhood best position g_{best} so far.

Step 2. Evaluate population

For each seeker t_i *(i = 1, 2, ..., SN)* evaluate the objective function values by Eq.(1).

Step 3. Record the best solution

In this step, the best solution vector, i.e. the solution vector with the highest objective function value is recorded.

Step 4. is repeated a fixed number of iterations. It consists of three parts. The details of each part are described as follows.

Part 1. Calculate new population

Perform an update process for each solution in the search population using a randomly selected search equation. The MSO included a new control parameter which is called behaviour rate (*BR*) in order to select the search equation in the following way: If a uniformly distributed real random number between [0,1) is less then *BR*, the SOA search equation is used, otherwise the search equation of ABC algorithm is performed.

The variant of ABC search equation for producing a new solution v_i, $i \in \{1, 2, ..., SN\}$ which is used in MSO algorithm is:

$$v_{i,j} = \begin{cases} t_{i,j} + \varphi_i(t_{i,j} - t_{k,j}), if \ \ R_j \le 0.5 \\ t_{i,j} \qquad\qquad\qquad\qquad , \ \ otherwise \end{cases} \tag{4}$$

where k is a randomly chosen index of a solution from the subpopulation to which the i^{th} seeker belongs, *k* has to be different from *i*, $j = 1,2,...,K$, φ_i is a uniformly distributed real random number between [-1, 1) and R_j is a uniformly random real number within [0, 1).

The SOA search solution equation uses search direction $d_{i,j}$ and step length $\alpha_{i,j}$ for producing a new solution v_i, $i \in \{1, 2, ..., SN\}$. It can be described by:

$$v_{i,j} = t_{i,j} + \alpha_{i,j} \cdot d_{i,j} \ , \ \ j = 1,2,...K \tag{5}$$

A search direction $d_{i,j}$ and a step length $\alpha_{i,j}$ are separately computed for each individual *i* on each dimension *j* at each iteration. The calculation of the search direction is based on a compromise among egotistic behavior, altruistic behavior and pro-activeness behavior. The egotistic behavior of each seeker t_i may be modeled by vector called egotistic direction d_{iego} by:

$$d_{iego,j} = p_{ibest,j} - t_{i,j} \ , \ j = 1,2,...K \tag{6}$$

The altruistic behavior of each seeker t_i may be modeled by vector called altruistic direction d_{ialt} by:

$$d_{ialt,j} = g_{best,j} - t_{i,j} \ , \ j = 1,2,...K \tag{7}$$

where g_{best} represents the neighbourhood best position so far.

The pro-active behavior of each seeker t_i may be modeled by vector called pro-activeness direction d_{ipro} by:

$$d_{ipro,j} = t_{i,j}(iter_1) - t_{i,j}(iter_2) \ , \ j = 1,2,...K \tag{8}$$

where $iter_1, iter_2 \in \{iter, iter-1, iter-2\}$, $t_i(iter_1)$ and $t_i(iter_2)$ are the best and the worst positions in the set $\{t_i(iter-2), t_i(iter-1), t_i(iter)\}$ respectively. Here, *iter* denotes the current iteration, while $iter-1$ and $iter-2$ denote the previous two iterations.

The expression of search direction for the i^{th} seeker is set to the stochastic combination of egotistic direction, altruistic direction and pro-activeness direction by:

$$d_{ij} = sign(w \cdot d_{ipro,j} + \varphi_1 \cdot d_{iego,j} + \varphi_2 \cdot d_{ialt,j}) \tag{9}$$

where $j = 1,2,...K$, the function sign $(\cdot)$ is a signum function on each dimension of the input vector, w is the inertia weight and φ_1 and φ_2 are real numbers chosen uniformly and randomly in the range [0,1]. Inertia weight is used to gradually reduce the local search effect of pro-activeness direction d_{ipro} and provide a balance between global and local exploration and exploitation. Inertia weight is linearly decreased from 0.9 to 0.1 during a run.

Fuzzy reasoning is used to generate the step length because the uncertain reasoning of human searching. From the view point of human searching behavior, it may be assumed that better points are likely to be found in the neighborhood of families of good points. For calculating the step length of i^{th} seeker we need to calculate vector μ_i by:

$$\mu_i = \mu_{\max} - \frac{S - I_i}{S-1} \cdot (\mu_{\max} - \mu_{\min}) \tag{10}$$

where *S* denotes the size of the subpopulation to which the seekers belong, I_i is the sequence number of t_i after sorting the objective function values in ascending order, $\mu_{\max}$ is the maximum

ISBN: 978-1-61804-092-3

membership degree value which is equal to or a little less than 1.0, $\mu_{\min}$ is set to 0.0111. Beside of vector μ_i, we need to calculate vector δ_i by :

$$\delta_i = w \cdot abs(t_{\max} - t_{\min}) \tag{11}$$

In Eq.(11), the absolute value of the input vector as the corresponding output vector is represented by the symbol abs $(\cdot)$, $t_{\max}$ and $t_{\min}$ are the positions of the best and the worst seeker in the subpopulation to which the i^{th} seeker belongs, respectively. In order to introduce the randomness in each variable and to improve the local search capability, the following equation is introduced to convert μ_i into a vector with elements as given by:

$$\mu_{ij} = rand(\mu_i, 1), \; j = 1,2,...K \tag{12}$$

The equation used for generating the step length $\alpha_{i,j}$ for i^{th} seeker is :

$$\alpha_{i,j} = \delta_{i,j} \cdot \sqrt{-\ln(\mu_{i,j})}, \; j = 1,2,...K \tag{13}$$

For each seeker t_i *(i = 1, 2, ..., SN)* evaluate the objective function values by Eq.(1).

Part 2. Evaluating all the seekers and saving the historical best position.

Part 3. Apply the modified inter-subpopulation learning operation

The modified inter-subpopulation learning is implemented as follows: The positions of seekers with the lowest objective function values of each subpopulation *l* are combined with the positions of seekers with the highest objective function values of *(l+z) mod N* subpopulations respectively, where *z=1,2,.. NSC*. *NSC* denotes the number of the worst seekers of each population which are combined with the best seekers. The appropriate seekers are combined using the following binomial crossover operator as expressed in:

$$t_{l_n j,worst} = \begin{cases} t_{i_j,best} & ,if \quad R_j \le 0.5 \\ t_{l_n j,worst} & , \quad otherwise \end{cases} \tag{14}$$

where R_j is a uniformly random real number within [0, 1), $t_{l_n j,worst}$ is denoted as the j^{th} variable of the n^{th} worst position in the l^{th} subpopulation, $t_{i_j best}$ is the j^{th} variable of the best position in the *i*th subpopulation. Additionally, we included a new parameter which we named inter-subpopulation learning increase period (*ILIP*). After *ILIP* iterations the number of the worst seekers of each subpopulation which are combined with the best seekers is increased to $2 \cdot NSC$.

Step 5. Output best recorded solution

After a predefined number of iterations the positions of the best recorded solution are the optimal threshold values.

4 Experimental Results and Discussion

The MSO and PSO algorithms have been implemented in Java programming language. Four well-known images, namely Lena, peppers, cameraman and boats with 256 gray levels are taken as the test images. All the images are of size (512 x 512), except the peppers image which is of size (256 x 256). These original images are shown in Fig 1. Tests were done on a PC with Intel® Core™ 2 Duo T8500 processor @4GHz with 4GB of RAM and Windows 7 x64 Ultimate operating system.

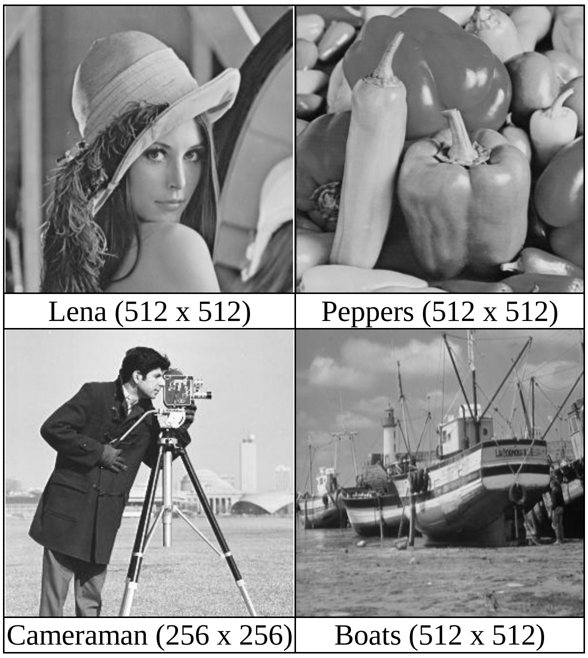

Fig. 1 Four experimental images

In all experiments for both algorithms the same size of population (*SP*) of 80 is used and the same size of maximum number of iterations (*MCN*) of 100 is taken. In proposed MSO algorithm the number of subpopulations (*N*) is 5, the behavior rate (*BR)* is 0.4, the number of seekers of each subpopulation for combination (*NSC*) is 1 and the inter-subpopulation learning increase period (*ILIP*) is $0.4 \cdot MCN$. Parameters of PSO algorithm are: inertia weight (*w*) is 0.5, minimum velocity ($v_{\min}$) is -5, maximum velocity ($v_{\max}$) is 5 , $\phi_{\min}$ is 0 and $\phi_{\max}$ is 2. Since MSO and PSO algorithms are of stochastic type and therefore the results of experiments are not

ISBN: 978-1-61804-092-3

absolutely the same in each run of algorithm, each experiment was repeated 50 times. Table 1 shows the optimal thresholds, the optimal objective function values and the processing time provided by the exhaustive search method. Table 2 presents the mean values, standard deviations and average processing time over 50 runs provided by both algorithms for each image with a threshold numbers from 1 to 4. The mean values and standard deviations obtained by MSO and PSO algorithms can be compared to the optimal objective function values derived by the exhaustive search method.

Image	k	Kapur		
		Threshold values	Objective function	Time (s)
Lena	1	123	8.9419	0.016
	2	97, 164	12.3470	1.297
	3	82, 126, 175	15.3181	46.828
	4	64, 97, 138, 179	18.0124	3079.703
Pepper	1	97	9.1190	0.000
	2	74, 149	12.5574	1.359
	3	69, 119, 167	15.6220	48.922
	4	55, 94, 134, 177	18.4005	3196.578
Cameraman	1	177	8.5289	0.015
	2	85, 176	12.2342	1.360
	3	76, 128, 180	15.4377	49.578
	4	76, 128, 180, 232	18.3286	3164.922
Boats	1	115	8.9642	0.000
	2	107, 176	12.5748	1.421
	3	64, 119, 176	15.8209	52.109
	4	48, 88, 128, 181	18.6557	3461.078

Table 1: Thresholds, objective function values and time processing provided by the exhaustive search

Image	k	PSO			MSO		
		Mean value	St. Dev.	Time(s)	Mean value	St. Dev.	Time(s)
Lena	1	8.9419	1.24E-14	0.1250	8.9419	1.24E-14	0.5250
	2	12.3470	5.33E-15	0.1256	12.3470	5.33E-15	0.5327
	3	15.3181	1.24E-14	0.1265	15.3181	1.24E-14	0.5421
	4	18.0027	1.05E-02	0.1269	18.0111	3.88E-03	0.5756
Pepper	1	9.1190	7.11E-15	0.1269	9.1190	7.11E-15	0.5556
	2	12.5574	1.07E-14	0.1275	12.5574	1.07E-14	0.5597
	3	15.6220	1.42E-14	0.1290	15.6220	1.42E-14	0.5666
	4	18.3992	9.07E-03	0.1316	18.4005	2.49E-14	0.5881
Cameraman	1	8.5289	0	0.1272	8.5289	0	0.5653
	2	12.2342	8.88E-15	0.1288	12.2342	8.88E-15	0.5628
	3	15.4377	1.07E-14	0.1288	15.4377	1.07E-14	0.5847
	4	18.2867	5.84E-02	0.1297	18.3286	2.13E-14	0.6031
Boats	1	8.9642	5.33E-15	0.1421	8.9642	5.33E-15	0.5978
	2	12.5748	1.42E-14	0.1422	12.5748	1.42E-14	0.6072
	3	15.8209	8.88E-15	0.1447	15.8209	8.88E-15	0.6172
	4	18.6345	3.40E-02	0.1448	18.6557	9.87E-03	0.6222

Table 2: Mean values, standard deviations and average processing time over 50 runs

From Table 2 it can be seen that both algorithms give good results both in terms of accuracy (mean fitness) and robustness (similar results of repeated runs or small standard deviation), for the threshold numbers from 1 to 3. For each image, for the threshold numbers from 1 to 3, MSO and PSO algorithms converged consistently to the same solution which is equal to the optimal solution. In this case, the standard deviations provided by both algorithms are very low. In the case when the number of thresholds is equal to 4, the MSO algorithm performs better than PSO algorithm for each image. We can see that for the threshold number 4, the mean values of MSO are closer to the optimal ones than the same of PSO. Also, in that case, the standard deviations obtained by MSO are lower than the standard deviations obtained by PSO, which is specially noticeable for the images peppers and cameraman. It can be concluded that MSO algorithm is superior to PSO in terms of precision and robustness of the results.

From the Table 1 we found that the computation times of exhaustive search method is exponential and for $k = 4$ it is unacceptable. From Table 2 it can be concluded that the execution times of segmentation using the MSO algorithm are about four to five times longer compared to the same of PSO algorithm. However, times of computation for both algorithms are less than one second, and therefore it can be considered that the differences in the computation time are negligible.

5 Conclusion

In this paper, the modified seeker optimization (MSO) algorithm based on simulating the act of human searching is proposed for multilevel thresholds selection using the maximum entropy criterion. Particle swarm optimization (PSO) algorithm is also implemented for comparison. The experimental results on four standard test images show that the MSO algorithm performs better than PSO algorithm with respect to the precision and robustness. From this work, it can be concluded that the proposed algorithm is a promising approach for image segmentation due to quality of its segmentation results and computational efficiency.

References:

[1] Batlle J., Casals A., Freixenet J., Marti J., A review on strategies for recognizing natural objects in colour images of outdoor scenes, Image and Vision Computing, Volume 18, Issues 6–7, 2000, pp. 515–530

[2] Chen Y.L., Nighttime Vehicle Light Detection on a Moving Vehicle using Image Segmentation and Analysis Techniques, WSEAS transactions on computers, Volume 8, Issue 3, 2009, pp. 506-515

[3] X. D. Zhuang, N. E. Mastorakis, Image Analysis Based on the Discrete Magnetic Field Generated by the Virtual Edge Current in Digital Images, WSEAS transactions on computers, Volume 10, Issue 8, August 2011, pp. 259-273

[4] M. Sezgin, B. Sankur, Survey over image thresholding techniques and quantitative performance evaluation, Journal of Electronic Imaging, Vol. 13, No. 1, 2004, pp. 146–165

[5] I. A. Yusoff, N. A. M. Isa, Two-Dimensional Clustering Algorithms for Image Segmentation, WSEAS transactions on computers, Volume 10, Issue 10, 2011, pp. 332-342

[6] S. P. Duraisamy, R. Kayalvizhi, A new multilevel thresholding method using swarm intelligence algorithm for image segmentation, Journal of Intelligent Learning Systems and Applications, Volume 2, Number 3, 2010, pp.126-138

[7] P. Y. Yin, Multilevel minimum cross entropy threshold selection based on particle swarm optimization, Applied Mathematics and Computation, Volume 184, Issue 2, 2007, pp. 503-513

[8] M. H. Horng, Multilevel thresholding selection based on the artificial bee colony algorithm for image segmentation, Expert Systems with Applications, Volume 38, Issue 11, 2011, pp. 13785–13791

[9] K. Hammouche, M. Diaf, P. Siarry, A comparative study of various meta-heuristic techniques applied to the multilevel thresholding problem, Engineering Applications of Artificial Intelligence, Volume 23, Issue 5, 2010, pp. 676–688

[10] Dai C., Chen W., Song Y., Zhu Y., Seeker optimization algorithm: a novel stochastic search algorithm for global numerical optimization, Journal of Systems Engineering and Electronics, Vol. 21, No. 2, 2010, pp. 300-311

[11] Brajevic I., Tuba M., Modified Seeker Optimization Algorithm for Unconstrained Optimization Problems, *Applied Mathematics in Electrical and Computer Engineering*, Proceedings of the American Conference on Applied Mathematics (AMERICAN-MATH '12), pp. 395-400, Harvard, Cambridge, USA, January 2012

[12] D. Karaboga, B. Akay, A Modified Artificial Bee Colony (ABC) Algorithm for Constrained Optimization Problems, Applied Soft Computing, Volume 11, Issue 3, 2011, pp. 3021-3031