Bat Algorithm (BA) for Image Thresholding

Adis ALIHODZIC
Faculty of Mathematics
University of Sarajevo
Zmaja od Bosne 33
Bosnia and Herzegovina
adis_mtkn@yahoo.com

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

Abstract: - Thresholding is an important approach for image segmentation and it is the first step in the image processing for many applications. Segmentation is a low level operation that can segment an image in non-overlapping regions. The optimal thresholds are found by maximizing Kapur's entropy-based thresholding function in a grey level image. However, the required CPU time increases exponentially with the number of desired optimal thresholds. In this paper a global multilevel thresholding algorithm for image segmentation is proposed based on the Bat inspired algorithm (BA). Cuckoo search (CS) algorithm was also implemented and compared with Kapur's and BA's algorithms. All algorithms have been tested on four sample images and experimental results show that both metaheuristics find excellent solutions, while computational time is negligible compared to exhaustive search.

Key-Words: - Bat algorithm, Maximum entropy thresholding, Image thresholding, Optimization metaheuristics, Nature inspired metaheuristics, Swarm intelligence

1 Introduction

Thresholding is a straightforward and an effective method for image segmentation. Image segmentation is the process of dividing an image into related non-overlapping regions which consists of image pixels and their relationship. In a grayscale image, the multilevel thresholding determines multiple thresholds which divide an image into multiple clusters. Thresholding is widely used in image processing and it has many applications in: automatic target recognition [1], clustering [2], medical image applications [3], optical character recognition [4] etc. For a classification of thresholding methods are used parametric and nonparametric approaches. When a parametric approach is used, the gray- level distribution of each cluster is assumed to obey a Gaussian distribution and then this approach attempts to find an estimate of the parameters of Gaussian distribution that best fits the histogram. However, nonparametric approach finds thresholds that seperate the gray-level clusters of an image by optimizing discriminating criteria defined from test images.

In this paper the entropy-based approach is used as an image multilevel thresholding technique, that has been studied by many researchers [5], [6], [7], [8]. Yin [5] suggested a new method that uses the particle swarm optimization to choose the thresholds based on the minimum cross-entropy. In order to improve the efficiency of Kapur's method, different methods have been developed for solving the multilevel thresholding problem [11]. In order to find optimal multilevel thresholds of images and increasing Kapur's segmentation result accuracy, a new metaheuristic BA-inspired algorithm (BA) [12] is combined with Kapur's method. Its performance was compared with the performance of the Kapur's method and cuckoo search algorithm (CS). An object-oriented software implementation of BA and CS algorithms was also provided for unconstrained optimization problems [13], [14]. The combined version of BA algorithm is used to search for the multilevel thresholds using the maximum entropy-based criterion. This algorithm can reduce the total runtimes, and is shown to give very good results when is compared with the rest algorithms [15]. The CS algorithm is also object-oriented implemented for purposes of comparison. The exhaustive search method is also object-oriented implemented for obtaining the optimal solutions in order to perform comparisons with the experimental results made by CS and BA algorithms.

The rest of the paper is structured as follows. Section 2 introduces the BA algorithm. Section 3 gives a detailed description of the multilevel

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

thresholding by BA algorithm. The experimental and comparative results of the implemented algorithms are discussed in Section 4. Section 5 presents the concluding of this work.

2 BA algorithm

The BA algorithm is new population based meta-heuristic approach proposed by *Xin-She Yang*. The algorithm exploits the so-called echolocation of the bats. The bat use sonar echoes to detect and avoid obstacles. It's generally known that sound pulses are transformed into a frequency which reflects from obstacles. The bats navigate by using the time delay from emission to reflection. The pulse rate is usually defined as 10 to 20 times per second, and it only lasts up about 8 to 10 ms. After hitting and reflecting, the bats transform their own pulse into useful information to explore how far away the prey is. The bats are using wavelength λ that vary in the range from 0.7 to 17 mm or inbound frequencies *f* of 20-500 kHz. Hence, we can also vary *f* while fixing λ, because λ and *f* are related due to the fact λf is constant. The pulse rate can be simply determined in the range from 0 to 1, where 0 means that there is no emission and 1 means that the bat's emitting is their maximum [16, 17]. The bat behaviour can be used to formulate a new BAT. Yang used three generalized rules when implementing the BA algorithms:

1. All bats use echolocation to sense distance, and they also 'know' the difference between food/prey and background barriers in some magical way;
2. Bats fly randomly with velocity v_i at position x_i with a fixed frequency f_{min}, varying wavelength λ and loudness A_0 to search for prey. They can automatically adjust the wavelength of their emitted pulses and adjust the rate of pulse emission $r \in [0,1]$, depending on the proximity of their target;
3. Although the loudness can vary in many ways, we assume that the loudness varies from a large (positive) A_0 to a minimum constant value A_{min}.

In BA algorithm, initialization of the bat population is performed randomly. In our simulations, we use virtual bats naturally. Namely, generating new solutions is performed by moving virtual bats according to the following equations:

$$f_i = f_{min} + (f_{max} - f_{min})\beta \quad (1)$$

$$v_i^t = v_i^{t-1} + (x_i^t - x_*)f_i \quad (2)$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (3)$$

where $\beta \in [0,1]$ is a random vector drawn from a uniform distribution. Here x_* is the current global best location (solution) which is located after comparing all the solutions among all the bats. In our implementation, we will use $f_{min}=0$ and $f_{max}=1$, depending the domain size of the problem of interest. Initially, each bat is randomly assigned a frequency which is drawn uniformly from $[f_{min}, f_{max}]$. A random walk with direct exploitation is used for the local search that modifies the current best solution according the equation:

$$x_{new} = x_{old} + \partial A^t \quad (4)$$

where $\partial \in [-1,1]$ is a random number, while A^t is the average loudness of all the best at this time step. The local search is launched with the proximity depending on the rate r_i of pulse emission. As the loudness usually decreases once a bat has found its pray, while the rate of pulse emission increases, the loudness can be chosen as any value of convenience. Hence, both characteristics imitate the natural bats. Mathematically, these characteristics are captured with the following equations:

$$A_i^{t+1} = \alpha A_i^t,\ r_i^{t+1} = r_i^0[1-\exp(-\gamma t)] \quad (5)$$

where α and γ are constants. Actually, α parameter plays a similar role as the cooling factor of a cooling schedule in the simulated annealing.

3 Proposed approach

The proposed algorithm consists from two parts. The first part introduces the objective function based on image entropy for later developing the BA algorithm. The second part gives a detailed description of the BA algorithm for multilevel image thresholding.

3.1 Entropy criterion based measure

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. The maximum entropy criterion for image thresholding, first proposed by Pun, and later corrected and improved by Kapur have been widely used in determining the optimal thresholding [11]. Kapur has developed the algorithm for bi-level thresholding, which can also extend to solve multilevel thresholding problems and can be described as follows.

Let there be *L* gray levels in a given image *I* having *M* pixels and these grey levels are in the range $\{0,1,...L-1\}$. The objective function is

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 $P_i = h(i)/M$. The aim is to maximize the objective function:

$$f([t_1, t_2, ..., t_k]) = H_0 + H_1 + H_2 ... + H_k \quad (6)$$

where

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

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

$$H_2 = -\sum_{i=t_2}^{t_3-1} \frac{P_i}{w_2} \ln \frac{P_i}{w_2}, \quad w_2 = \sum_{i=t_2}^{t_3-1} P_i , ...$$

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

3.2 Image thresholding based on BA

The proposed BA algorithm based on maximum entropy criterion tries to obtain this optimum *K*-dimensional vector $[t_1, t_2, ..., t_k]$ which can maximize Eq.(3). The objective function is also used as the fitness function for the proposed algorithm. The details of the developed approach are introduced as follows.

Step 1. (*Generate the initial population of solutions*)
BA algorithm generates a randomly distributed initial population of *N* solutions (bats) x_i *(i = 1, 2, ..., N)* with *K* dimensions denoted by matrix *X*,

$$X = [x_1, x_2, ...x_N] \text{ and } x_i = (x_{i,1}, x_{i,2}, ...x_{i,K}) \quad (4)$$

where x_{ij} is the j^{th} component value that is restricted into [0,…,L-1] and the $x_{ij} < x_{ij+1}$ for all *j*. The fitness values of all solutions x_i are evaluated and set cycle = 1. Before starting to iterative search process, the BA algorithm detects the most successful solution as x_{best} solution.

Step 2. (*Generation of new solutions*)
Generating new solution is performed by moving virtual bats according to equations (1), (2), (3). Namely, we generate new solutions by adjusting the vector of frequencies *f* and by updating the matrix of velocities *V* using the Eq. (1) and (2) respectively. After, we perform an update process for each solution in the search population *X* using the Eq. (3).

Step 3. (*Local searching*)
A random walk with direct exploitation is used for the local search that modifies the current best solution vector according the Eq. (4). Namely, the local search is launched with the proximity depending on pulse rate *r* (Eq. 5).

Step 4. (*Generation of a new solution by flying randomly*)
In this step, for the current best solution vector obtained in step 3, we evaluate the objective function values by Eq.(6). If the fitness value of solution vector is higher than the old fitness value and loudness *A* obtained by Eq. (5) is no loud, then accept new solution vector and update the old fitness value. Otherwise, keep the old best solution.

Step 5. (*Record the best solution*)
Memorize the best solution vector with the currently highest objective function value and the best optimum as (x_{best}). Add the cycle by one.

Step 6. (*Check the termination criterion*)
If the cycle is equal to the maximum number of iterations then finish the algorithm, else go to Step 2.

4 Experimental results and discussion

The BA and CS algorithms have been object/oriented implemented in C# programming language. Four well-known images, namely Lena, Barbara, Peppers and Living room with 256 grey levels are taken as the test images. All the images are of size (512 x 512). These images with their histograms are shown in Fig 1. Tests were done on a PC with AMD Core2Duo TK-55 (1.8 GHz, 2 x 256 KB L2 cache) with 2GB of RAM on Windows 7 Operating System in Visual Studio 2010. Control parameters of the BA algorithm are: population size (*N*), the maximum number of iterations, loudness (*A*), pulse rate (r), minimum frequency (f_{min}), maximum frequency (f_{max}). Control parameters of the CS algorithm are: the number of nests (*N*), the maximum number of iterations and discovering probability (p_α).

In all experiments both CS and BA algorithms use the same size of population (N=50). Parameters of BA algorithm are the following: number of iterations is 2500, loudness (*A*) is 0.95, pulse rate (*r*) is 0.45, minimum frequency (f_{min}) is 0, maximum frequency (f_{max}) is 1. Parameters of CS algorithm are: the number of iterations is 2000, discovering

ISBN: 978-960-474-330-8

probability (p_α) is 0.7. Each experiment was repeated 30 times within different thread. In this paper we use multithreaded approach. The size N and the maximum number of iterations have a great impact on the convergence and on the computing time. As these two parameters are related, for both algorithms the same size N (number of nests or population size) is used. The maximum number of iterations was taken as a variable, in order to further facilitate the comparison between them for the time convergence. In order to compare the quality of the results achieved by BA and CS algorithms for the multilevel thresholding, the value of the best fitness $F(T^*)$ corresponding to the best threshold solution T^* is used as **comparative criterion**. The run of each algorithm within different thread was stopped when: $\left|F(T^*) - F_{opt}\right| \le \varepsilon = 10^{-9}$, where F_{opt} is the optimal value of the objective function, and ε is a threshold value which fixes the accuracy of the measurement. We have computed and recorded the iteration number and the time taken by each algorithm to achieve the desired accuracy. In that way the stopping condition for all algorithms is based on the value of the fitness and not on the number of iterations.

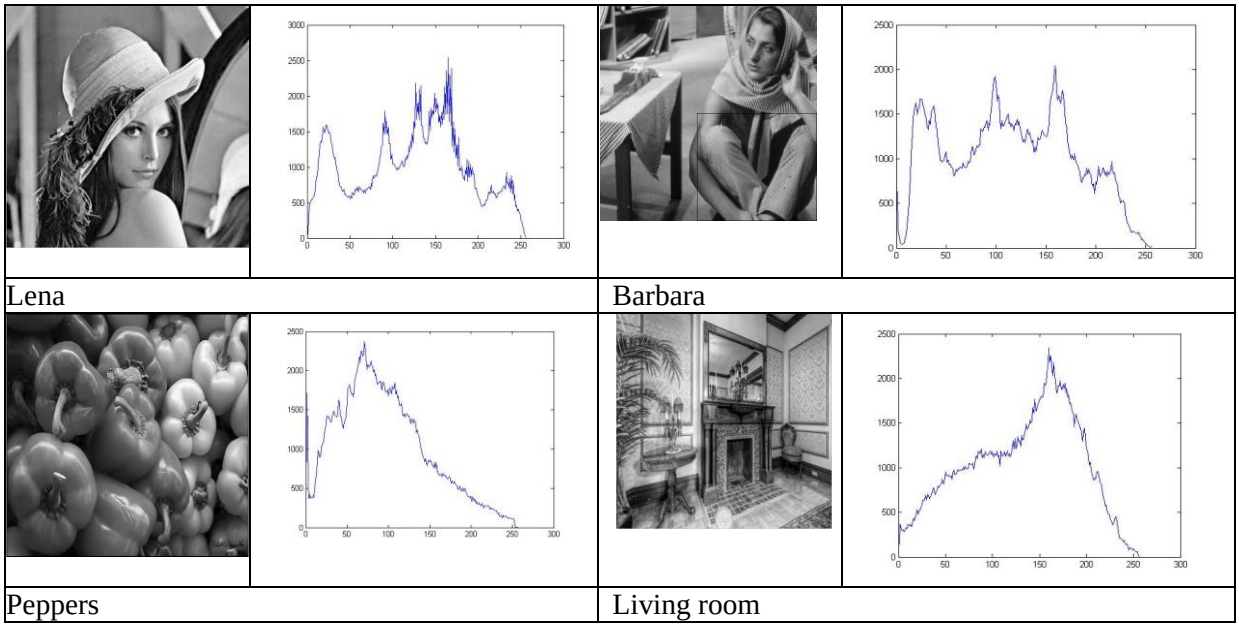

Fig 1: Test images and their histograms

Image	K	Kapur's entropy-based algorithm		
		Threshold values	Objective function	Time (sec)
Lena	1	120	9.48564855803021	0.009
	2	88, 177	13.1492917680291	0.689
	3	65, 124, 189	16.3895110974036	29.781
	4	43, 86, 140, 195	19.3648944426231	1919.311
Barbara	1	123	9.42434018184525	0.008
	2	89, 171	13.0022314713505	0.707
	3	68, 125, 183	16.196072952122	30.809
	4	51, 95, 143, 193	19.1186242078948	1971.163
Peppers	1	139	9.45598950787067	0.008
	2	91, 174	13.0383663093328	0.694
	3	66, 135, 195	16.3679772729138	29.623
	4	50, 101, 152, 203	19.4094392684649	1369.68
Living room	1	119	9.43277474985872	0.009
	2	78, 154	12.9532418577676	0.69
	3	67, 133, 199	16.2419474407854	30.46
	4	49, 99, 149, 202	19.2846272108906	1272.907

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

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 and standard deviations over 30 runs within 30 different threads provided by all algorithms for each image with a threshold numbers from 1 to 4, while Table 3 reports the mean number of iterations and the average of the CPU time taken by each algorithm within 30 different threads to satisfy the stopping condition.

Image	K	CS		BA	
		Mean value	St. Dev.	Mean value	St. Dev.
Lena	1	**9.48564855803022**	**5.33E-15**	**9.48564855803022**	**5.33E-15**
	2	13.1492917680291	3.55E-15	13.1459677521564	6.07E-03
	3	16.3895110974036	7.11E-15	16.3726415549379	1.36E-02
	4	19.3648944426231	1.42E-14	19.3247367486576	2.81E-02
Barbara	1	**9.42434018184525**	**0.00E+00**	**9.42434018184525**	**0.00E+00**
	2	13.0022314713505	8.88E-15	12.9985036993387	7.62E-03
	3	16.196072952122	0.00E+00	16.1706800302371	2.76E-02
	4	19.1186242078948	3.55E-15	19.0807732071767	3.81E-02
Peppers	1	**9.45598950787067**	**1.78E-15**	**9.45598950787067**	**1.78E-15**
	2	13.0383663093328	8.88E-15	13.0361434865082	4.43E-03
	3	16.3679772729138	3.55E-15	16.3406231299671	2.51E-02
	4	19.4094392684649	1.07E-14	19.3609622300783	4.59E-02
Living room	1	**9.43277474985871**	**3.55E-15**	**9.43277474985871**	**3.55E-15**
	2	12.9532418577676	3.55E-15	12.9524139842829	1.70E-03
	3	16.2419474407854	7.11E-15	16.2260812700426	1.32E-02
	4	19.2846272108906	1.07E-14	19.2438310795026	4.39E-02

Table 2: Mean values and standard deviations over 30 multi-threaded runs

Image	K	CS		BA	
		Time (ms)	Iteration number	Time (ms)	Iteration number
Lena	1	15.96	5.86	2.26	0.710
	2	106.50	44.86	49.86	39.49
	3	398.30	145.76	61.76	48.34
	4	881.00	362.26	70.53	50.00
Barbara	1	16.46	5.83	2.20	0.776
	2	114.26	48.76	38.13	29.55
	3	406.36	144.10	61.66	48.43
	4	1000.93	415.20	69.8	50.00
Peppers	1	18.76	6.03	2.30	0.82
	2	126.90	50.33	42.66	40.30
	3	390.66	137.93	62.36	50.00
	4	909.56	373.63	69.1	50.00
Living room	1	19.03	5.8	5.1	3.11
	2	143.8	57.6	45.13	41.02
	3	423.66	152.30	60.70	50.00
	4	857.36	354.00	72.2	50.00

Table 3: Average computational time of the CS and BA algorithms over 30 multi-threaded runs

From Table 2 it can be seen that for threshold number 1 (each image), BA algorithm performs equally as CS algorithm, both in terms of accuracy (mean fitness) and robustness (small standard deviation). The mean values obtained by BA algorithm are almost equal to the optimal objective

ISBN: 978-960-474-330-8

function values derived from the exhaustive search method for each image and each threshold number.

From the Table 1 it can be seen that the computation time of exhaustive search method is exponential and for $K = 4$ it is unacceptable. The reported results from the Table 3 show that as for the exhaustive search, for both algorithms, the number of iterations and the run time increase with the number of threshold, but not in the same manner. We can see from Table 3 that for the threshold numbers from 1 to 4, for all of the test images, BA algorithm is more efficient in terms of computation time than CS algorithm. Also, the convergence times of both algorithms are faster than those of the exhaustive search.

5 Conclusion

We propose the bat inspired algorith (BA) algorithm based on echo locations behaviour for multilevel thresholds selection using the maximum entropy criterion. The experimental results demonstrated that the proposed BA algorithm can search for multiple thresholds which are close to the optimal ones determined by the exhaustive search method. Compared to the CS, the computational times show that the BA algorithm outperformed CS algorithm. The contribution of this paper is to demonstrate the feasibility of BA method for multilevel thresholding. Also, it offers a new option to the conventional methods due to its simplicity and efficiency.

References:

[1] G.C. Anagnostopoulos, SVM-based target recognition from synthetic aperture radar images using target region outline descriptors. Nonlinear Anal.-Theor. Meth. App. **2009**, 71(12), e2934–e2939.

[2] H.F. Ng, Automatic thresholding for defect detection, Pattern Recognition Letters, Volume 27, Issue 14, 2006, pp. 1644-1649.

[3] Doelken, M.T.; Stefan, H.; Pauli, E.; Stadlbauer, A.; Struffert, T.; Engelhorn, T.;Richter, G.;Ganslandt, O.; Doerfler, A.;Hammen, T. 1H-MRS profile in MRI positive- *versus* MRI negative patients with temporal lobe epilepsy. *Seizure* **2008**, *17*(6), 490–497.

[4] Lázaro, J.; Martín, J.L.; Arias, J.; Astarloa, A.; Cuadrado, C. Neuro semantic thresholding using OCR software for high precision OCR applications. Image Vision Comput. 2010, 28(4), 571–578.

[5] 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.

[6] M.H. Horng, Multilevel Minimum Cross Entropy Threshold Selection based on Honey Bee Mating Optimization, Proc. of the 3rd WSEAS Int. Conf. on Circuits, Systems, Signal and Telecommunications (CISST'09),2009, pp. 25-30.

[7] 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.

[8] M.H. Horng, T.W. Jiang, Multilevel Image Thresholding Selection Based on the Firefly Algorithm, 2010 Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing, 2010, pp.58-63.

[9] M. Tuba, I. Brajevic, and R. Jovanovic, "Hybrid seeker optimization algorithm for global optimization", Applied Mathematics and Information Sciences, Vol. 7, No. 3, May 2013, pp. 867–875

[10] M. Tuba, "Asymptotic Behavior of the Maximum Entropy Routing in Computer Networks", Entropy, Vol. 15, Issue 1, 2013, pp. 361-371

[11] 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.

[12] X. S. Yang. A new metaheuristic BA-inspired algorithm. Nature Inspired Cooperative Strategies for Optimisation (NICO 2010), pages 65-74, 2011.

[13] A. Alihodzić, M. Tuba, Framework for BA Algorithm Optimization Metaheuristic, 4th International Conference on Bioscience and Bioinformatics (ICBB '13), Chania, Crete Island, Greece, August 27-29, 2013.

[14] N. Bacanin, Implementation and performance of an object-oriented software system for cuckoo search algorithm, International Journal of Mathematics and Computers in Simulation, Volume 6, Issue 1, 2012, pp. 185 - 193.

[15] M. Tuba, R. Jovanovic, "Improved ant colony optimization algorithm with pheromone correction strategy for the traveling salesman problem", International Journal of Computers, Communications & Control, Vol. 8, Issue 3, Jun 2013, pp. 477–485