

Improved Glowworm Swarm Optimization Algorithm applied to Multi-level Thresholding

Simone A. Ludwig
North Dakota State University
Fargo, ND, USA
simone.ludwig@ndsu.edu

Abstract—Image segmentation is considered an important basic task in the analysis and understanding of images. The process of image segmentation involves the partitioning of an image into multiple regions, i.e., each pixel in the image is assigned a label that corresponds to certain visual characteristics. Therefore, image segmentation is widely used for image processing tasks such as classification and object recognition. The thresholding process, however, is an NP-hard problem since n thresholds need to be found and each threshold can take numbers from 0 to 255 according to the intensity values. Thus, exact methods are only feasible for a small number of thresholds. Nature-inspired algorithms have recently been applied to this hard optimization problem and shown very good results. In particular, many swarm intelligence methods have been applied to multi-level thresholding. This paper uses a glowworm swarm optimization (GSO) approach and further enhances it in order to improve the accuracy of the thresholding as well as to improve the robustness of the results. The proposed approach makes adjustments to the local decision radius, the selection method, and the step size. The proposed GSO algorithm shows improved results in particular for larger numbers of thresholds as compared to the basic GSO algorithm.

I. INTRODUCTION

Image processing is a method that converts an image into digital form and performs some operation on it such as to get an enhanced image or to extract some useful information. Image processing is a sort of signal dispensation whereby the input is an image such as a video frame or a photograph and the output might be an image or characteristics that are associated with a particular image. Generally, an image processing system considers images as two dimensional signals that are modified by applying signal processing methods.

There are three major steps involved in image processing:

- Import of an image via optical scanner or digital photography;
- Analysis and manipulation of the image involving data compression, image enhancement, and identifying patterns that are not visible by the human eye such as satellite photographs;
- Output of image processing can be either the altered image or a report containing statistics based on the image analysis.

The field of image processing is usually categorized into the following five sections:

- Visualization: observation of objects that are not visible;
- Image sharpening and restoration: improved image;

- Image retrieval: Searching for the image of interest;
- Measurement of patterns: Measuring of various objects in an image;
- Image recognition: Objects in an image are distinguished from other objects or the background.

Image processing encompasses different techniques that belong to signal processing, morphological processing and segmentation for feature detection, as well as high level artificial intelligence algorithms for object recognition, information extraction, representation, and understanding. During the different stages of image processing hard optimization problems are encountered. One example is image thresholding, which is a step in segmentation where n thresholds need to be found all in the range [0-255] according to the intensity values. This has been proven to be a NP-hard problem [1].

The image segmentation process partitions an image into multiple regions, i.e., each pixel in the image is assigned a label that corresponds to certain visual characteristics. For example, for binary image segmentation, one threshold will be found which identifies the pixels in the image that will be represented as white (above the threshold), and the other pixels as black (below the threshold). Image segmentation is considered an important basic task in the analysis and understanding of images. Therefore, image segmentation is widely used for image processing tasks such as classification and object recognition [2]. Image segmentation can be classified into different techniques based on the threshold method [3], clustering algorithm [4], compression-based method [5], histogram-based method [6], edge detection method [7], and region based split and merging methods [8].

Thresholding methods are divided into two categories: optimal thresholding methods [9], [10], [11], [12], [13] and property-based thresholding methods [14], [15], [16]. Optimal thresholding methods search for the optimal thresholds based on desired characteristics of the thresholded classes of the histogram. This is usually achieved by the optimization of an objective function. The property-based thresholding methods obtain the thresholds by measuring some chosen property of the histogram, and are fast methods, which make them suitable for multilevel thresholding. However, the number of thresholds has to be known in advance, which is difficult to determine.

Image thresholding is one of the most widely used segmentation methods applied, which uses information contained in the image histogram. The selection of the multiple thresholds

is the important portion in image segmentation since the segmentation depends on the optimally identified thresholds. Depending on the number of peaks of the histogram of an image, the number of thresholds should be determined.

This paper proposes an enhanced glowworm swarm optimization (GSO) algorithm applied to the multi-level thresholding segmentation task. The paper is structured as follows: Section 2 presents related work in the area of image segmentation including a description of different thresholding techniques. In Section 3, first the Otsu method is described followed by a description of the basic GSO algorithm and the enhanced GSO algorithm. Section 4 consists of the benchmark images used, parameters set, and evaluation measures used for the evaluation. In Section 5, the results of the experiments are given and discussed. Section 6 concludes this paper with a summary of the findings.

II. RELATED WORK

As has been proven in [4], finding the optimal thresholds using an exhaustive search method such as Otsu [11] (see details in the following section) is computationally very expensive. As stated in the paper, the exhaustive search for $n-1$ optimal thresholds results in the evaluation of $n(L-n+1)^{n-1}$ combinations of thresholds (L is the intensity level). Thus, this method is not feasible for larger numbers of thresholds. Therefore, approximate techniques have been applied to this thresholding optimization problem.

In particular, nature-inspired algorithms have been applied to the thresholding optimization problem since they have been successfully applied in situations where conventional optimization techniques cannot find satisfactory solutions, in particular, when the function to be optimized is discontinuous, non-differentiable, and/or involves too many nonlinear parameters [17].

Evolutionary computation and swarm intelligence algorithms are very popular global optimization methods and have been applied to the image segmentation thresholding problem. A range of different algorithms has been applied to the multi-level thresholding problem including genetic algorithm [18], [19], particle swarm optimization [20], [21], [22], artificial bee colony [23], [24], [25], differential evolution [26], [27], [28], firefly algorithm [29], [30], and cuckoo search [31], [32]. All these approaches have used different evolutionary techniques and swarm intelligence concepts in combination with different objective functions.

Glowworm swarm optimization (GSO) has also been investigated. In [33], a basic GSO algorithm has been applied to the multi-level thresholding problem using the criterion of minimum cross entropy. The GSO approach has been compared to the exhaustive search algorithm, the honey bee mating optimization, the firefly algorithm, the artificial bee colony algorithm, and the particle swarm optimization algorithm. The experiments were carried out using five benchmark images and the experimental results showed that the proposed GSO approach efficiently identifies up to five thresholds that are very close to the optimal thresholds identified by the

exhaustive search method. Furthermore, compared to the other thresholding techniques, the computational time of GSO is competitive taking the second or third place behind the firefly algorithm and the artificial bee colony algorithm.

Another GSO algorithm [34] optimizes the threshold image segmentation problem using the Otsu method. Three test images were used and the comparison is performed using the Otsu method and the combined Otsu+GSO algorithm for one, two, and three thresholds. The experimental results demonstrate that the Otsu+GSO algorithm performs the image segmentation more effectively, and also reduces the execution time significantly as is the case with applying a meta-heuristic in general.

This paper applies the standard GSO algorithm to six images using the settings of the previous algorithm [34] (setting of [33] can not be compared with since a different optimization objective - criterion of minimum cross entropy - was used). Furthermore, a few modifications are made in order to improve the GSO algorithm applied to the image threshold segmentation problem, which are explained in detail in the following section.

III. MULTI-LEVEL THRESHOLD SEGMENTATION ALGORITHM USING GSO

There are several threshold segmentation algorithms available that use different objective functions to find the optimal thresholds in an image. For example, the Tsallis entropy [35] method uses a generalization of the standard Boltzmann-Gibbs entropy (entropy is a measure of uncertainty of the information content of a system). Another measure is the minimum cross entropy method [36] whereby two probability distributions are assumed to belong to the same set, and thus, the cross entropy between these two distributions is minimized. The most famous is the Otsu method, which is described in more detail in the next subsection.

A. Between-class Variance Method - Otsu

The Otsu method is a non-parametric segmentation method, which divides an image into classes such that the between-class variance is maximized. Let us assume an image has N pixels and L gray levels. The number of pixels at level i is represented by f_i , then $N = f_1 + f_2 + \dots + f_L$. The occurrence probability of level i is defined by:

$$p_i = \frac{f_i}{N}, p_i \geq 0, \sum_{i=1}^L p_i = 1 \quad (1)$$

For bi-level thresholding, the optimum threshold t divides the image into two classes, and the cumulative probabilities of each class are described as:

$$\omega_0 = \sum_{i=1}^t p_i, \omega_1 = \sum_{i=t+1}^L p_i \quad (2)$$

The mean level of the two classes are then obtained by:

$$\mu_0 = \sum_{i=1}^t \frac{ip_i}{\omega_0}, \mu_1 = \sum_{i=t+1}^L \frac{ip_i}{\omega_1} \quad (3)$$

The between-class variance of the two classes is then defined by:

$$f(t) = \sigma_0 + \sigma_1 \quad (4)$$

$$\sigma_0 = \omega_0(\mu_0 - \mu_T)^2 \quad (5)$$

$$\sigma_1 = \omega_1(\mu_1 - \mu_T)^2 \quad (6)$$

where μ_T is the mean level of the entire image, $\mu_T = \sum_{i=1}^L ip_i$. The optimum threshold t^* is identified with an exhaustive search by maximizing the between-class variance. Thus, the optimal threshold is then:

$$t^* = \arg \max_{1 \leq t \leq L} (f(t)) \quad (7)$$

The Otsu method can be extended to multi-level thresholding by the following. Let us assume an image is divided into M classes, then the extended between-class variance of M classes is calculated by:

$$f(t) = \sum_{i=0}^{M-1} \sigma_i \quad (8)$$

The sigma terms are obtained by:

$$\begin{aligned} \sigma_0 &= \omega_0(\mu_0 - \mu_T)^2, \\ \sigma_1 &= \omega_1(\mu_1 - \mu_T)^2, \dots, \\ \sigma_{M-1} &= \omega_{M-1}(\mu_{M-1} - \mu_T)^2 \end{aligned} \quad (9)$$

And, the mean levels are calculated by:

$$\begin{aligned} \mu_0 &= \sum_{i=1}^{t_1} \frac{ip_i}{\omega_0}, \\ \mu_1 &= \sum_{i=t_1+1}^{t_2} \frac{ip_i}{\omega_1}, \dots, \\ \mu_{M-1} &= \sum_{i=t_{M-1}+1}^L \frac{ip_i}{\omega_{M-1}} \end{aligned} \quad (10)$$

Thus, the optimum thresholds are obtained by maximizing the between-class variance as such:

$$t^* = \arg \max_{1 \leq t_1, t_2, \dots, t_{M-1} \leq L} \left(\sum_{i=0}^{M-1} \sigma_i \right) \quad (11)$$

B. GSO based Multi-level Thresholding Segmentation Algorithm

This subsection first describes the GSO algorithm, followed by the GSO based multi-level thresholding algorithm, and the enhanced version of the GSO based multi-level thresholding algorithm.

1) *GSO Algorithm*: The GSO algorithm operates as follows. First, a swarm of glowworms are randomly initialized within the search space. Each glowworm carries an individual amount of luciferin l_i and this amount affects nearby glowworms within a variable neighborhood. Each glowworm has its own vision scope referred to as the local decision range $r_d^i (0 < r_d^i < r_s)$ (r_s is the sensor range). Glowworm i uses a probabilistic mechanism to select its adjacent glowworm j with a larger luciferin value within the decision range. The brighter the glowworm glows, the better is the position (objective value). The regional decision radius is adaptively adjusted depending on the number of neighbors a glowworm has.

The luciferin update process is based on the glowworm's current position. In each iteration, the luciferin value is based on the value of the last iteration, and is proportional to the objective value of the current position ($J(x_i(t))$) including some decay with time. The update process is described by:

$$l_i(t) = (1 - \rho)l_i(t-1) + \gamma J(x_i(t)) \quad (12)$$

where $\rho \in (0, 1)$ is a parameter to control the luciferin value, γ is the luciferin enhancement constant.

The set of neighbors of each glowworm include only those glowworms that have a higher luciferin value and are located within the dynamic decision domain based on the sensor range $0 < r_d^i(t) < r_s$. The neighborhood is defined by $N_i(t) = j : d_{ij}(t) < r_d^i(t); l_i(t) < l_j(t)$, where r_d^i is the decision radius of the i^{th} glowworm, $x_j(t)$ is the j^{th} glowworm's position of the t^{th} iteration, $l_i(t)$ is the i^{th} glowworm's luciferin value, and d_{ij} is the euclidean distance between glowworm i and j . Glowworm i uses a probabilistic mechanism to select its adjacent glowworm j and will move towards the direction of glowworm j according to:

$$P_{ij} = \frac{l_j(t) - l_i(t)}{\sum_{k \in N_i(t)} l_k(t) - l_i(t)} \quad (13)$$

The movement update is performed by:

$$x_{i+1}(t) = x_i(t) + s \left(\frac{x_j(t) - x_i(t)}{\|x_j(t) - x_i(t)\|} \right) \quad (14)$$

where s is the step size of the movement of glowworm i . The local decision range is updated by:

$$r_d^i(t+1) = \min\{r_s, \max\{0, r_d^i(t) + \beta(n_t - |N_i(t)|)\}\} \quad (15)$$

2) *GSO based Multi-level Thresholding Segmentation Algorithm (GSO)*: Algorithm 1 outlines the steps of the GSO based segmentation algorithm. The algorithm starts with the random initialization of the glowworm population. Furthermore, the necessary parameters of the algorithm as well as the maximum number of iterations, and the number of thresholds are defined. Then, the while loops proceeds until the stopping criterion (max. number of iterations) is reached, whereby all glowworms are iterated over performing the following steps: (1) calculate the luciferin value, (2) choose the neighbor within the decision making radius, (3) determine the direction of movement, (4) glowworm movement and update of location,

(5) calculation of fitness value (objective value) according to Eq. (11), and (6) dynamic decision radius update. At the end of the run, the best glowworm will be returned, i.e., the threshold values are returned.

Algorithm 1 GSO based Multi-level Thresholding Algorithm

```

Random initialization of glowworm population
Initialization  $l_0, r_0, r_s, s, \rho, \gamma, x_i(0)$ 
Initialization of maximum number of iterations
Set number of thresholds
while stopping criterion is not met do
  for all glowworms do
    Set luciferin value to objective function value
    Choose neighbor within decision making radius
    Determine the direction of movement
    Move and update the location
    Update the fitness value according to Eq. (11)
    Update the dynamic decision radius
  end for
end while
return best glowworm (solution)

```

3) *Enhanced GSO based Multi-level Thresholding Segmentation Algorithm (EGSO)*: The GSO based multi-level thresholding algorithm is enhanced as follows:

- The local decision radius is an important parameter that affects the search ability. Thus, the sensor range of each glowworm is extended to the whole image.
- As used in [33], instead of the probabilistic mechanism outlined in Section II, a tournament selection mechanism with $k = 2$ is used for the update of the location of each glowworm.
- The step size s is a very important parameter that affects the convergence of the GSO algorithm, and thus, the step size should be smaller than the distance to the optimal solution. A lower step size might lead to a slower convergence speed whereas a large step size might lead to no convergence at all. Thus, as is the case with all evolutionary computation and swarm intelligence algorithms, the exploration of the search space is favored during the early iterations, and the exploitation is favored during the later iterations of the search. Thus, a variable step size is proposed by:

$$s(t) = \frac{s_{max} - (s_{max} - s_{min})}{\left(\frac{t}{t_{max}}\right)^d} \quad (16)$$

where s_{max} and s_{min} are the maximum (set to 3) and minimum step size (set to 0.001), respectively, t is the current iteration, t_{max} is the maximum number of iterations, and d is the dimensionality of the search space, which is based on number of thresholds defined.

- Once 10% of the population have reached the same solution, the random movement of 5% is performed.
- Different parameters than previously used are listed in Table I.

IV. EXPERIMENTAL SETUP

This section describes the benchmark images used for the experiments, the parameters used to run the algorithms, and the evaluation measures used to identify the best performing algorithm.

A. Benchmark Images

Six different benchmark images from the USC Viterbi Image repository [37], and from the Waterloo Image repository [38] were selected for the experiments. The intensity histograms are given in Figure 1.

B. Parameters of GSO and EGSO algorithms

Table I lists the parameter setup of the basic GSO and EGSO algorithms. The parameters for the GSO algorithm are used as given in [34]. Compared to other evolutionary algorithms and swarm intelligence techniques, many more parameters are involved and need to be predefined.

TABLE I
PARAMETERS OF GSO AND EGSO

Parameter	GSO	EGSO
Number of glowworms	50	50
Maximum number of iterations	100	100
Initial luciferin l_0	5	5
Neighborhood threshold n_t	3	3
Luciferin enhancement constant γ	0.6	0.6
Luciferin decay constant ρ	0.4	0.4
Step size s	0.3	see Eq. (16)
Decision domain update β	0.08	0.08
Initial sensing radius r_0	5	255

C. Evaluation measures

Besides the objective measure given in Eq. (11), the segmentation algorithms usually use the peak-to-signal-ratio (PSNR), and the structural similarity index (SSIM) [39]. The PSNR is given by:

$$\text{PSNR}(x, y) = 20 \log_{10} \left(\frac{255}{\text{RMSE}(x, y)} \right) \quad (17)$$

where

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^X \sum_{j=1}^Y (I(i, j) - I'(i, j))^2}{X \times Y}} \quad (18)$$

where $X \times Y$ is the size of the image, I is the original image, and I' is the segmented image.

The SSIM index measures the similarity between the original image and the segmented image and is expressed by:

$$\text{SSIM}(I, I') = \frac{(2\mu_I \mu_{I'} + c_1)(2\sigma_{II'} + c_2)}{(\mu_I^2 + \mu_{I'}^2 + c_1)(\sigma_I^2 + \sigma_{I'}^2 + c_2)} \quad (19)$$

where μ_I is the mean of image I , $\mu_{I'}$ is the mean of image I' , σ_I^2 is the variance of image I , and $\sigma_{I'}^2$ is the variance of image I' . The two variables $c_1 = (k_1 D)^2$ and $c_2 = (k_2 D)^2$ are used to stabilize the division operation, D is the dynamic range of the pixel-values, $k_1 = 0.01$, and $k_2 = 0.03$.

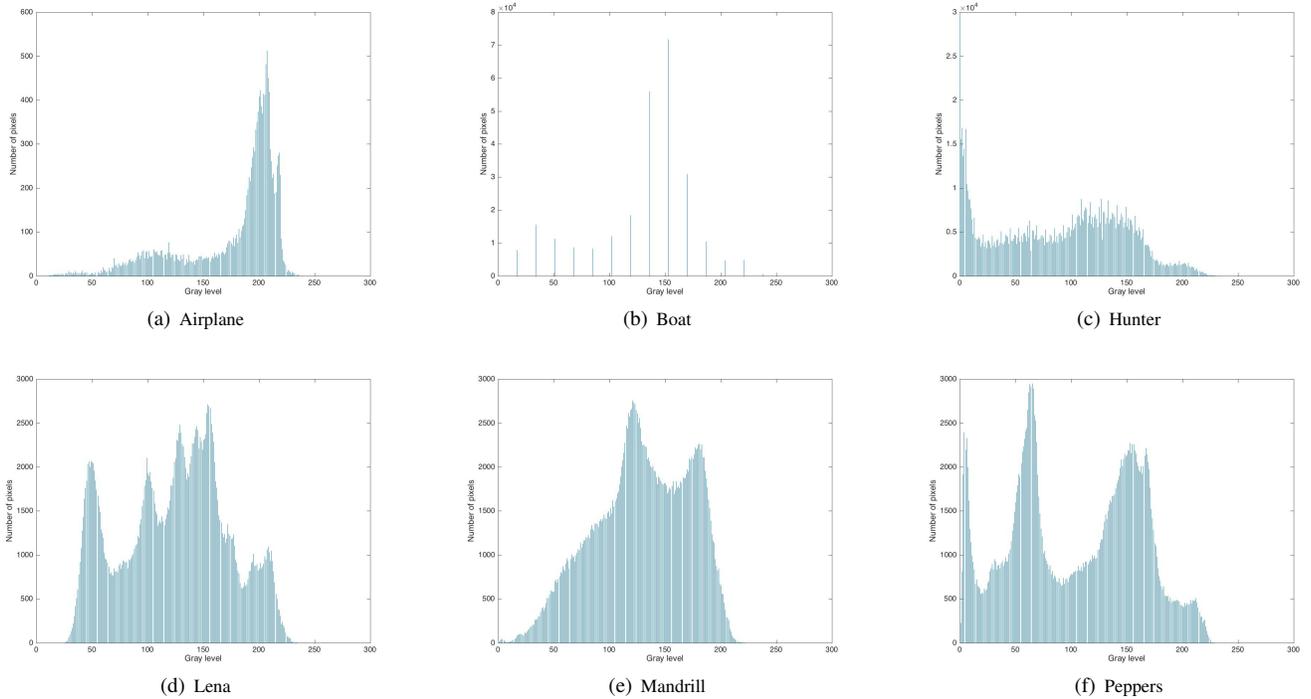


Fig. 1. Intensity histograms

V. RESULTS

All results presented in the following tables are average values of 50 independent runs. The runs of the GSO and EGSO algorithms returned the threshold intensity values for threshold values set from 1 to 5, the objective value, execution time, PSNR value, and SSIM value.

Table II shows the threshold intensity values, objective value, and the execution time for the predefined threshold values of 1 to 5. The objective values of GSO and EGSO are compared and the best values are given in bold font. As can be seen from the table, the EGSO algorithm achieves the better objective values for threshold values above 2. For threshold values of 1 and 2, the GSO algorithm scores either equally well or better. However, a Wilcoxon signed rank test revealed that the objective values of EGSO compared to GSO are not statistically significantly better. In terms of execution time, the EGSO algorithm needs some extra time for the additions made to the algorithm. However, the average increase is around 0.118s exhibiting larger values for larger number of thresholds.

Table III shows the average PSNR values and standard deviation obtained of 50 independent runs. As can be seen in the table, the PSNR values of the EGSO algorithm are better than the GSO algorithm for threshold values including and above 2 with the exception of the Boat image (at threshold=2, the PSO algorithm has a higher PSNR value than the GSO algorithm). The GSO algorithm achieves the same PSNR value or above for a given threshold of 1. The EGSO algorithm compared to GSO turns out to be more robust as can be seen by the smaller standard deviation values. A Wilcoxon signed

rank test returned a p-value of 0.000028 and confirmed that EGSO is significantly better than GSO in terms of PSNR.

Table IV shows the average SSIM values with standard deviation obtained of 50 independent runs. The GSO algorithm scores higher SSIM values for threshold=1, and also on the Boat and Hunter images for threshold=2. However, for all the other threshold values EGSO achieves the higher values. Again, the EGSO algorithm is more robust compared to GSO based on the smaller standard deviation values obtained. Some of the standard deviation values are 0 indicating that the 50 runs produced the same result i.e., found the same thresholds. A Wilcoxon signed rank test returned a p-value of 0.000262 and confirmed that EGSO is significantly better than GSO in terms of SSIM.

VI. CONCLUSION

Image segmentation involves the partitioning of an image into multiple regions, i.e., each pixel in the image is assigned a label that corresponds to certain visual characteristics. Since the thresholding process is an NP-hard problem, exact methods are only feasible for a small number of thresholds. Thus, nature-inspired algorithms have recently been applied to this hard optimization problem and shown very good results. This paper investigated a glowworm swarm optimization approach with further enhancements made to improve the accuracy of the thresholding as well as the robustness. The proposed approach improves the basic GSO algorithm with adjustments to the local decision radius, the selection method, and the step size. The non-parametric segmentation method measuring

TABLE II
THRESHOLD INTENSITY, OBJECTIVE VALUE AND EXECUTION TIME

Image	GSO			EGSO		
	Threshold intensity	Objective value	Exec. time (s)	Intensity	Objective value	Exec. time (s)
Airplane	152	1.587E+03	1.148E+00	152	1.587E+03	1.216E+00
	121; 178	1.757E+03	1.144E+00	121; 177	1.757E+03	1.209E+00
	93; 146; 187	1.837E+03	1.229E+00	91; 141; 186	1.838E+03	1.265E+00
	86; 140; 179; 200	1.877E+03	1.339E+00	84; 128; 169; 198	1.881E+03	1.553E+00
	65; 111; 135; 180; 202	1.900E+03	1.440E+00	67; 106; 142; 177; 201	1.908E+03	1.839E+00
Boat	112	1.622E+03	1.513E+00	105	1.622E+03	1.182E+00
	93; 156	1.892E+03	1.609E+00	101; 168	1.892E+03	1.497E+00
	72; 125; 165	2.027E+03	1.685E+00	83; 128; 153	2.027E+03	1.786E+00
	55; 93; 129; 168	2.071E+03	1.801E+00	67; 109; 148; 185	2.099E+03	2.060E+00
	52; 85; 114; 139; 174	2.135E+03	1.919E+00	61; 118; 140; 161; 203	2.128E+03	2.340E+00
Hunter	83	2.523E+03	3.078E+00	84	2.523E+03	2.808E+00
	56; 122	2.947E+03	3.217E+00	56; 123	2.948E+03	3.146E+00
	39; 91; 142	3.127E+03	3.323E+00	38; 91; 141	3.127E+03	3.482E+00
	28; 79; 118; 151	3.199E+03	3.410E+00	34; 81; 123; 163	3.209E+03	3.750E+00
	23; 56; 92; 130; 156	3.243E+03	3.672E+00	27; 64; 99; 133; 171	3.255E+03	4.131E+00
Lena	117	1.601E+03	1.533E+00	117	1.601E+03	1.238E+00
	91; 150	1.961E+03	1.623E+00	92; 150	1.962E+03	1.549E+00
	82; 123; 170	2.126E+03	1.635E+00	79; 125; 170	2.128E+03	1.839E+00
	75; 107; 144; 179	2.188E+03	1.728E+00	74; 113; 144; 179	2.191E+03	2.104E+00
	76; 88; 111; 145; 183	2.200E+03	1.829E+00	72; 108; 135; 159; 187	2.217E+03	2.409E+00
Mandrill	127	1.219E+03	1.543E+00	127	1.219E+03	1.247E+00
	96; 148	1.548E+03	1.595E+00	96; 148	1.548E+03	1.525E+00
	85; 124; 161	1.638E+03	1.701E+00	84; 124; 160	1.638E+03	1.878E+00
	72; 107; 136; 165	1.692E+03	1.787E+00	71; 105; 136; 167	1.692E+03	2.114E+00
	63; 104; 131; 153; 171	1.711E+03	1.908E+00	66; 98; 124; 148; 173	1.718E+03	2.382E+00
Peppers	102	2.594E+03	1.484E+00	103	2.594E+03	1.228E+00
	48; 115	2.866E+03	1.602E+00	48; 115	2.866E+03	1.506E+00
	41; 100; 151	3.065E+03	1.670E+00	42; 98; 152	3.066E+03	1.773E+00
	37; 91; 140; 176	3.149E+03	1.869E+00	40; 88; 134; 174	3.152E+03	2.081E+00
	38; 91; 120; 149; 186	3.187E+03	1.959E+00	38; 79; 117; 149; 181	3.196E+03	2.399E+00

TABLE III
PSNR - MEAN AND STANDARD DEVIATION

Image	Level	GSO		EGSO	
		Average PSNR	Std. dev. PSNR	Average PSNR	Std. dev. PSNR
Airplane	1	1.151E+01	5.216E-03	1.151E+01	7.022E-03
	2	1.483E+01	5.656E-01	1.494E+01	3.346E-03
	3	1.760E+01	1.090E+00	1.900E+01	0.000E+00
	4	1.955E+01	1.286E+00	2.099E+01	1.628E-02
	5	2.077E+01	1.124E+00	2.322E+01	1.121E-02
Boat	1	1.387E+01	4.718E-01	1.372E+01	4.327E-01
	2	1.557E+01	8.822E-01	1.495E+01	5.915E-01
	3	1.832E+01	1.083E+00	1.891E+01	5.843E-01
	4	1.914E+01	1.704E+00	2.045E+01	5.940E-01
	5	1.948E+01	2.854E+00	2.112E+01	7.015E-01
Hunter	1	1.374E+01	1.770E-02	1.376E+01	2.550E-02
	2	1.687E+01	2.896E-01	1.703E+01	2.607E-03
	3	1.881E+01	5.807E-01	1.943E+01	0.000E+00
	4	2.009E+01	7.352E-01	2.097E+01	1.379E-02
	5	2.129E+01	6.322E-01	2.258E+01	1.501E-02
Lena	1	1.207E+01	9.537E-03	1.206E+01	1.569E-02
	2	1.520E+01	3.990E-01	1.534E+01	0.000E+00
	3	1.692E+01	4.952E-01	1.738E+01	0.000E+00
	4	1.798E+01	8.766E-01	1.866E+01	9.433E-03
	5	1.894E+01	1.165E+00	1.969E+01	3.549E-01
Mandrill	1	1.098E+01	4.055E-02	1.095E+01	5.032E-02
	2	1.524E+01	5.311E-01	1.549E+01	0.000E+00
	3	1.718E+01	1.079E+00	1.782E+01	3.150E-03
	4	1.927E+01	1.066E+00	2.031E+01	3.987E-02
	5	2.016E+01	1.395E+00	2.177E+01	7.831E-02
Peppers	1	1.294E+01	9.602E-03	1.295E+01	1.374E-02
	2	1.589E+01	3.887E-01	1.614E+01	8.408E-02
	3	1.821E+01	8.511E-01	1.886E+01	0.000E+00
	4	1.928E+01	8.808E-01	2.041E+01	2.351E-02
	5	2.049E+01	8.560E-01	2.183E+01	2.165E-02

TABLE IV
SSIM - MEAN AND STANDARD DEVIATION

Image	Level	GSO		EGSO	
		Average SSIM	Std. dev. SSIM	Average SSIM	Std. dev. SSIM
Airplane	1	6.060E-01	3.860E-04	6.057E-01	5.200E-04
	2	7.324E-01	1.466E-02	7.379E-01	4.100E-05
	3	7.964E-01	2.944E-02	8.172E-01	0.000E+00
	4	8.152E-01	3.195E-02	8.288E-01	2.560E-04
	5	8.293E-01	3.154E-02	8.585E-01	4.940E-04
Boat	1	4.183E-01	4.019E-03	4.171E-01	3.745E-03
	2	5.062E-01	3.182E-02	4.836E-01	1.443E-02
	3	6.102E-01	3.583E-02	6.382E-01	1.538E-02
	4	6.520E-01	6.391E-02	7.107E-01	2.141E-02
	5	6.747E-01	9.542E-02	7.484E-01	2.726E-02
Hunter	1	3.516E-01	1.420E-04	3.515E-01	2.040E-04
	2	4.604E-01	1.003E-02	4.599E-01	2.270E-04
	3	5.461E-01	1.849E-02	5.599E-01	0.000E+00
	4	5.921E-01	1.755E-02	6.020E-01	2.870E-04
	5	6.297E-01	2.638E-02	6.581E-01	4.410E-04
Lena	1	4.199E-01	1.192E-03	4.187E-01	1.961E-03
	2	5.303E-01	2.786E-02	5.304E-01	0.000E+00
	3	5.986E-01	2.415E-02	6.050E-01	0.000E+00
	4	6.278E-01	4.069E-02	6.435E-01	3.350E-04
	5	6.608E-01	4.305E-02	6.805E-01	1.422E-02
Mandrill	1	3.750E-01	1.605E-03	3.740E-01	1.992E-03
	2	6.104E-01	2.422E-02	6.227E-01	0.000E+00
	3	6.862E-01	3.612E-02	7.069E-01	6.000E-06
	4	7.566E-01	3.254E-02	7.949E-01	9.650E-04
	5	7.820E-01	3.872E-02	8.321E-01	9.620E-04
Peppers	1	3.534E-01	3.000E-04	3.532E-01	4.290E-04
	2	5.169E-01	7.593E-02	5.645E-01	2.340E-02
	3	5.881E-01	1.648E-02	5.975E-01	0.000E+00
	4	6.161E-01	2.341E-02	6.395E-01	6.120E-04
	5	6.507E-01	2.552E-02	6.732E-01	5.560E-04

the between-class variance (Otsu) was used as the objective measure for the evaluation.

The enhanced GSO algorithm (EGSO) is compared against the basic GSO algorithm. The experiments are performed using six standard benchmark images measuring the PSNR (peak-to-signal-ratio) and SSIM (structural similarity index). Besides these measures, the algorithms obtains the threshold intensity values, objective value, and execution time. The results reveal that the EGSO algorithm is more robust than GSO as seen by the standard deviation values. Furthermore, EGSO obtains significantly better PSNR and SSIM values for threshold values of 2 and above.

Further work will investigate the proposed algorithm using different objective measures such as Tsallis entropy and minimum cross entropy in order to see whether the same improvements of EGSO can be achieved.

REFERENCES

- [1] A. Nakib, H. Oulhadj, P. Siarry, Image thresholding based on Pareto multiobjective optimization. *Engineering Applications of Artificial Intelligence*, 23, 313-320, 2010.
- [2] M. Sezgin, B. Sankur, Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1), 146-168, 2004.
- [3] S. Q. Han and L. Wang. A Survey of Thresholding Methods for Image Segmentation System Engineering and Electronics, 4 (6) pp. 91-102, 2002.
- [4] R. V. Kulkarni, G. K. Venayagamoorthy, Bio-inspired algorithms for autonomous deployment and localization of sensor. *IEEE Transactions on Systems*, 40(6), 663-675, 2010.
- [5] H. Mobahi, S. R. Rao, A. Y. Yang, S. S. Sastry and Y. Ma. Segmentation of Natural Images by Texture and Boundary Compression. *International Journal of Computer Vision (IJCV)*, 95 (1), 86-98, 2011.
- [6] L. G. Shapiro, G. C. Stockman, *Computer Vision*, New Jersey, Prentice-Hall, 279-325, 2001.
- [7] S. L. Horowitz, T. Pavlidis. Picture Segmentation by a Directed Split and Merge Procedure. *Proc. ICPR*, 424-433, 1974.
- [8] A. D. Brink, Minimum spatial entropy threshold selection. *IEEE Proceedings on Vision Image and Signal Processing*, 142, 128-132, 1995.
- [9] J. N. Kapur, P. K. Sahoo, A. K. C. Wong, A new method for gray-level picture thresholding using the entropy of the histogram. *Computer Vision Graphics Image Processing*, 2, 273-285, 1985.
- [10] J. Kittler, J. Illingworth, Minimum error thresholding. *Pattern Recognition*, 19, 41-47, 1986.
- [11] N. Otsu, A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, Cybernetics, SMC-9*, 62-66, 1979.
- [12] T. Pun, A new method for grey-level picture thresholding using the entropy of the histogram. *Signal Processing*, 2, 223-237, 1980.
- [13] T. Pun, Entropy thresholding: A new approach. *Computer Vision Graphics Image Processing*, 16, 210-239, 1981.
- [14] Y. K. Lim, S. U. Lee, On the color image segmentation algorithm based on the thresholding and the fuzzy c-means techniques. *Pattern Recognition*, 23, 935-952, 1990.
- [15] D. M. Tsai, A fast thresholding selection procedure for multimodal and unimodal histograms. *Pattern Recognition Letters*, 16, 653-666, 1995.

- [16] P. Y. Yin, L. H. Chen, New method for multilevel thresholding using the symmetry and duality of the histogram. *Journal of Electronics and Imaging*, 2, 337-344, 1993.
- [17] D. Floreano, C. Mattiussi, *Bio-inspired artificial intelligence: Theories, methods, and technologies*. Cambridge, MA: MIT Press, 2008.
- [18] W. B. Tao, J. W. Tian, J. Liu, Image segmentation by three-level thresholding based on maximum fuzzy entropy and genetic algorithm, *Pattern Recognition Letters*, 24, 3069-3078, 2003.
- [19] K. Hammouche, M. Diaf, P. Siarry, A multilevel automatic thresholding method based on a genetic algorithm for a fast image segmentation, *Computer Vision and Image Understanding*, 109, 163-175, 2008.
- [20] P. Y. Yin, Multilevel minimum cross entropy threshold selection based on particle swarm optimization, *Applied Mathematics and Computation*, 184, 503-513, 2007.
- [21] M. Maitra, A. Chatterjee, A hybrid cooperative-comprehensive learning based PSO algorithm for image segmentation using multilevel thresholding, *Expert Systems with Applications*, 34, 1341-1350, 2008.
- [22] B. Akay, A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding, *Applied Soft Computing*, 13, 3066-3091, 2013.
- [23] M. H. Horng, Multilevel thresholding selection based on the artificial bee colony algorithm for image segmentation, *Expert Systems with Applications*, 38, 13785-13791, 2011.
- [24] Y. Zhang, L. Wu, Optimal multi-level thresholding based on maximum Tsallis entropy via an artificial bee colony approach, *Entropy*, 13, 841-859, 2011.
- [25] A. K. Bhandari, A. Kumar, G. K. Singh, Modified artificial bee colony based computationally efficient multilevel thresholding for satellite image segmentation using Kapur's, Otsu and Tsallis functions, *Expert Systems with Applications*, 42, 1573-1601, 2015.
- [26] E. Cuevas, D. Zaldivar, M. Perez-Cisneros, A novel multi-threshold segmentation approach based on differential evolution optimization, *Expert Systems with Applications*, 37, 5265-5271, 2010.
- [27] S. Sarkar, S. Das, Multilevel image thresholding based on 2D histogram and maximum Tsallis entropy - a differential evolution approach. *Image Processing, IEEE Transactions on*, 22, 4788-4797, 2013.
- [28] H. V. H. Ayala, F. M. dos Santos, V. C. Mariani, L. dos Santos Coelho, Image thresholding segmentation based on a novel beta differential evolution approach, *Expert Systems with Applications*, 42, 2136-2142, 2015.
- [29] M. H. Horng, T. W. Jiang, Multilevel image thresholding selection based on the firefly algorithm. In *Ubiquitous Intelligence and Computing and 7th International Conference on Autonomic Trusted Computing (UIC/ATC)*, 58-63, 2010.
- [30] M. H. Horng, R. J. Liou, Multilevel minimum cross entropy threshold selection based on the firefly algorithm, *Expert Systems with Applications*, 38, 14805-14811, 2011.
- [31] S. Agrawal, R. Panda, S. Bhuyan, B. K. Panigrahi, Tsallis entropy based optimal multilevel thresholding using cuckoo search algorithm, *Swarm and Evolutionary Computation*, 11, 16-30, 2013.
- [32] A. K. Bhandari, V. K. Singh, A. Kumar, G. K. Singh, Cuckoo search algorithm and wind driven optimization based study of satellite image segmentation for multilevel thresholding using Kapur's entropy, *Expert Systems with Applications*, 41, 3538-3560, 2014.
- [33] M. H. Horng, Multilevel Image Thresholding with Glowworm Swarm Optimization Algorithm based on the Minimum Cross Entropy, *Journal of Advances in Information Sciences and Service Sciences (AISS)*, Volume 5 Issue 10, 1290-1298, 2013.
- [34] Q. Luo, Z. OuYang, X. Chen, Y. Zhou, A Multilevel Threshold Image Segmentation Algorithm Based on Glowworm Swarm Optimization, *Journal of Computational Information Systems* 10: 4, 1621-1628, 2014.
- [35] C. Tsallis, In: S. Abe, Y. Okamoto (Eds), *Non-extensive statistical Mechanics and its applications*, Series Lecture Notes in Physics, Springer, Berlin, 2001.
- [36] N. R. Pal, On Minimum Cross-Entropy Thresholding, *Pattern Recognition Journal, Elsevier*, 29(4), 575-580, 1996.
- [37] Image Benchmark Repository, USC Viterbi, <http://sipi.usc.edu/database/database.php?volume=misc>, last accessed January 2016.
- [38] Waterloo Image Repository, University of Waterloo, <http://links.uwaterloo.ca/Repository.html>, last accessed January 2016.
- [39] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: from error visibility to structural similarity, *IEEE Transactions on Image Processing*, 13, 600-612, 2004.