

# Optimal Threshold Computing in Automatic Image Thresholding using Adaptive Particle Swarm Optimization

P. Moallem<sup>\*1</sup>, N. Razmjoooy<sup>2</sup>

<sup>1</sup> Department of Electrical Engineering  
University of Isfahan  
Hezarjerib Street, Isfahan, Iran.  
\*p\_moallem@eng.ui.ac.ir

<sup>2</sup> Young Researchers club, Majlesi branch  
Islamic Azad University  
Isfahan, Iran.

## ABSTRACT

Selecting an optimal threshold value is the most important step in image thresholding algorithms. For a bimodal histogram which can be modeled as a mixture of two Gaussian density functions, estimating these densities in practice is not simply feasible. The objective of this paper is to use adaptive particle swarm optimization (APSO) for the suboptimal estimation of the means and variances of these two Gaussian density functions; then, the computation of the optimal threshold value is straightforward. The comparisons of experimental results in a wide range of complex bimodal images show that this proposed thresholding algorithm presents higher correct detection rate of object and background in comparison to the other methods including Otsu's method and estimating the parameters of Gaussian density functions using genetic algorithm (GA). Meanwhile, the proposed thresholding method needs lower execution time than the PSO-based method, while it shows a little higher correct detection rate of object and background, with lower false acceptance rate and false rejection rate.

Keywords: histogram-based thresholding, adaptive particle swarm optimization, genetic algorithm, fitness function, object and background detection

## 1. Introduction

Segmentation, which refers to the process of partitioning a digital image into multiple segments, plays an important role in many applications of computer vision [1]. The simplest method for digital image segmentation is the thresholding method.

For a bimodal gray-level image which includes bright objects on dark backgrounds, the thresholding process labels individual pixels in the image as "object" pixels, if their gray level is greater than some threshold value, and as "background" pixels otherwise. Finally, regarding the label of pixels, a binary image is created by assigning each pixel '1' or '0'. The thresholding is widely used in applications including digital books, legal documents, architectural and electronic maps, and drawings [2].

Computing the threshold value as the key parameter in the thresholding method is not always straightforward. Automatic global thresholding

methods, which try to compute the overall threshold value automatically based on some image information, are categorized into the following five groups based on the manipulated information [3]:

1- *Histogram shape-based methods*: in which some information such as peaks and valleys from the histogram of the smoothed image are extracted and analyzed. Ideally, the histogram of a bimodal image consists of two peak points; the bottommost point (valley) which is between these two peaks denotes the threshold point, as shown in Figure 1 [4].

2- *Clustering-based methods*, where the image histogram is modeled as a mixture of two Gaussian density function, or alternately the gray-level samples are clustered in two partitions as background and object.

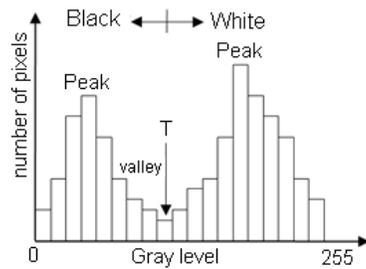


Figure 1. Histogram of a sample gray-level bimodal image,  $T$  is the threshold value.

3- *Entropy-based methods* use the entropy of the object and background regions, or the cross-entropy between the original and binarized image.

4- *Object attribute-based methods* analyze a measure of similarity between the gray-level and the binarized images.

5- *Spatial methods* use some spatial information of pixels as well as the other information.

The three last groups use some extra information such as spatial information or binarized images, while the first two methods are based on the distribution of image gray levels, which are the basic information for image thresholding. Therefore, a histogram-based image thresholding algorithm can be simply used in the other image thresholding algorithms.

Otsu's method as one of the most famous automatic image thresholding algorithm is also used to perform histogram shape-based image thresholding. This algorithm assumes two peaks and hence two classes (object and background) in the image histogram and calculates the optimal threshold separating those two classes which minimizes the intra-class variance [5].

For an image containing only two principal gray-level regions (e.g., "object" and "background"), the image histogram can be modeled as a mixture of two Gaussian density functions [6]. After the estimation of the parameters of these two Gaussian density functions, the mean and the variance, the optimal threshold can be computed simply [7].

Accuracy of parameters estimation of "object" and "background" density functions play an important role in the image thresholding results (e.g., the correct detection rate of "object" and "background"). Typically, experiments show that evolutionary methods estimate accurately the parameter of models in image segmentation [8].

The genetic algorithm (GA) is effectively used to estimate the means and the variances of two Gaussian density functions in the histogram-based image thresholding [9]. The accuracy of this method is outstanding, but the computational cost is higher than with Otsu's method. In order to improve the accuracy of GA-based thresholding, it is proposed to use particle swarm optimization (PSO) for image thresholding [10, 11]; however, the computational cost increases as well. In this paper, in order to simultaneously improve the accuracy and decrease the computational cost, Adaptive PSO (APSO) is used to accurately estimate a threshold in a bimodal gray-level image.

The rest of this paper is organized as follows. In Section 2, an overview on PSO algorithms is presented. Section 3 presents our proposed thresholding algorithm including its fitness function and APSO optimization for computing the optimal threshold value. The experimental results and discussions are given in Section 4 and finally, the paper is summarized and concluded in Section 5.

## 2. PSO Overview

Since our proposed thresholding algorithm is based on APSO algorithm, an overview on PSO algorithm as one of the evolutionary optimization methods is presented in this section. Typically, experiments show that evolutionary methods are successfully used for segmentation purposes [8,9,12]. Evolutionary algorithms often perform well approximating solutions to all types of problems because they ideally do not make any assumption about the underlying problem.

Kennedy, a social psychologist, and Eberhart, an electrical engineer, presented the PSO algorithm [11]. Their first aim was to exploit social models and a social relation in soft computing. Their first imagery done in 1995 led them to simulate the behavior of birds at seed finding. PSO algorithm is

known as a random initial value applied to population to start analyzing.

In the standard PSO algorithm, each particle in a swarm population traces its status in the search space by tracking two maximum values that consist of the best position (*Pbest*) which is found in a distant place and the best fitted particle's position in the total number of populations.

Unlike other evolutionary algorithms, in PSO each candidate depends on a value that was renowned as velocity. The resolvent candidate is the particle that searches the included searching space.

The velocity value is obtained on the basis of experiences and use of the other particle's experience adjusted as a constant value. Velocity updating frames  $v_{ij}$  of each particle and its position  $x_{ij}$  in standard PSO is presented as

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_1 (Pbest_i(t) - x_{ij}(t)) + c_2 r_2 (Gbest_i(t) - x_{ij}(t)) \tag{1}$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \tag{2}$$

where *Pbest* and *Gbest* are the best position and the position of the known best-fit particle in the entire population, respectively;  $c_1$ , and  $c_2$  are the learning factors and are nonnegative constants;  $r_1$  and  $r_2$  are random numbers uniformly distributed in the interval [0, 1];  $v_{ij}$  and  $x_{ij}$  are particle's velocity and position, respectively, and  $v_{max}$  is a designated maximum velocity which is a constant preset by users according to the objective optimization function. If the velocity on one dimension exceeds the maximum, it will be set to  $v_{max}$ .

### 2.1 Velocity update in PSO

To offset the target wide range exploitation and local nearby exploration abilities of swarm optimization, Eberhart and Shi introduced the concept of a varying inertia weight which is applied in a much better way than a fixed inertia weight. The weight update is written as

$$w = w_{max} - gen \cdot \frac{w_{max} - w_{min}}{gen_{max}} \tag{3}$$

In Equation 3,  $gen_{max}$  consists of the maximum generation selected by the user,  $w_{max}$  and  $w_{min}$  are the maximum and minimum of inertia weight; those can be set by the user respectively.

### 3. The Proposed Algorithm

In this section, the proposed algorithm for computing the optimal threshold value is presented. Firstly, it is necessary to estimate all the parameters of a parametric function from the image histogram. This problem can be solved by optimizing a fitness function that is minimized by the APSO optimization algorithm. Then, the optimal threshold is computed by the computed parameters.

#### 3.1 Fitness Function and Threshold Value

The gray-level distribution of an image or the image histogram is the main input information for any histogram-based image thresholding. For an image with  $L-1$  gray levels, the image histogram can be declared as a discrete function,  $p(r_k)$ , as follows:

$$p(r_k) = \frac{n_k}{n} \tag{4}$$

Where

$$n = \sum_{r_k=0}^{L-1} n_k \tag{5}$$

$$\sum_{r_k=0}^{L-1} p(r_k) = 1,$$

where  $r_k$  is the  $k^{th}$  gray level ( $r_k \in [0, L-1]$ ),  $n_k$  is the number of pixels whose gray level is  $r_k$ , and  $n$  is the total number of pixels in image; in other words, the histogram of an image is an approximation of a gray level  $r_k$  odds in an image. Histogram of an image for all values of  $k$  gives a general description from the image status without any information about its inside.

The histogram of a bimodal gray-level image can be approximated by mixing two probability density functions,  $p_1(x)$  and  $p_2(x)$ ; then, the image histogram,  $p(x)$ , can be computed by Equation 6:

$$p(x) = P_1 p_1(x) + P_2 p_2(x) \quad (6)$$

$$P_1 + P_2 = 1$$

where  $P_i$  denotes the prior probability of class  $i$  and  $x$  is a gray level between 0 to  $L-1$ .

Optimal threshold value  $T$  can be computed by minimizing total error probability and considering the Bayesian rule as Equation 6 [7] shows.

$$P_1 p_1(T) = P_2 p_2(T) \quad (7)$$

It is ideal for histogram-based image thresholding algorithms to find the threshold value as close as possible to the optimal threshold value (See Equation 7). However, the image histogram preserves the image statistical information, but the histogram shape-based methods for selecting the threshold value, a valley between two peaks in the histogram, is not always the optimal threshold value.

Figure 2(a) shows a typical histogram of a bimodal image, and Figure 2(b) shows the  $p_1(x)$  and  $p_2(x)$ , the probability density function of the black and white zones, respectively.  $T'$  is the computed threshold by considering the valley between two peaks in the histogram shown in Figure 2(a). In equal prior probability on the black and white zones, it means  $P_1=P_2$  in Equation 7,  $T$  is the computed optimal threshold, by considering  $p_1(T)=p_2(T)$ , which is shown in Figure 2(b).

The main purpose of this paper is to segment the image into two regions (black and white zones) and find the suboptimal threshold for the thresholding. In complex situations where there is high inference between two probability density functions, for example when  $p_1(x)$  and  $p_2(x)$  are close to each other, the image histogram is not a two-hump one and, unfortunately, the other powerful histogram shape-based method, Otsu's method, sometimes fails to find the optimal threshold. For example, even in Figure 2, the computed threshold value by Otsu's method is between  $T$  and  $T'$ , and the output binarized image is not suitable for some applications.

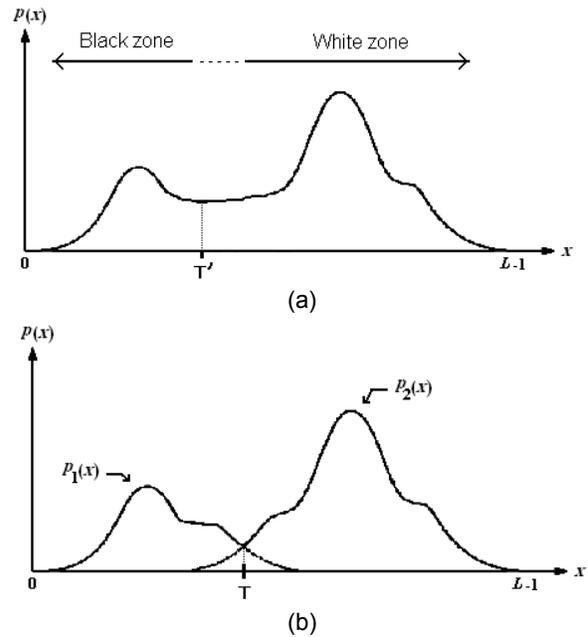


Figure 2. a) A typical histogram of a bimodal image in which  $T'$  is the valley between two peaks; b)  $p_1(x)$  and  $p_2(x)$  are the probability density functions of the black and white zones, respectively shown by bold lines; and  $T$  is the optimal threshold value considering equal prior probability of the black and white zones.

On the other hand, estimating  $p_1(x)$  and  $p_2(x)$  from the image histogram is not a straightforward problem. Therefore, using Equation 7 to find the optimal threshold value is challenging. Considering  $p_1(x)$  and  $p_2(x)$  as two Gaussian probability density functions is one of the practical solutions. It means that the histogram of a bimodal gray-level image can be approximated by mixing two Gaussian probability density functions by Equation 8:

$$p_{\text{estimated}}(x) = \sum_{i=1}^2 P_i p_i(x) \quad (8)$$

$$= \sum_{i=1}^2 \frac{P_i}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right)$$

where  $P_i$  denotes the prior probability of class  $i$ ,  $p_i(x)$  is Gaussian distribution function  $x$  of class  $i$ ,  $\mu_i$  and  $\sigma_i$  are the mean and standard deviations of the  $i^{\text{th}}$  distribution probability function.

For the given gray-level image, it is apparent that if the image is composed of distinct objects with different gray levels, the image histogram usually shows several peaks. Each peak corresponds to one object and adjacent peaks which are separate by a valley. The height of a peak implies the number of pixels falling in the bin corresponding to the location of the peak [7], [13]. Hence, the image can be considered as a multi-modal gray-level image and Equation 4 can be extended to the multimodal case.

Typically for a bimodal case, a root mean square (RMS) method is proposed to estimate the six parameters including  $P_1, \mu_1, \sigma_1$  for the first class and  $P_2, \mu_2, \sigma_2$  for the second class. The root mean square between the histogram function and the estimated Gaussian probability function is simplified as

$$E(P_1, \mu_1, \sigma_1, P_2, \mu_2, \sigma_2) = \sum_{r_k=0}^{L-1} [p_{\text{estimated}}(r_k) - p(r_k)]^2 \quad (9)$$

Fitness function  $E$  is defined on the  $L$ -point image histogram. Typically, solving such problems imply solving a system of complex nonlinear equations.

After computing the optimal values of the six parameters ( $P_1, \mu_1, \sigma_1$  and  $P_2, \mu_2, \sigma_2$ ), the optimal threshold value  $T$  can be computed by solving Equation 4 which leads the quadratic Equation 10 [7].

$$AT^2 + BT + C = 0 \quad (10)$$

Although the above quadratic equation has two possible solutions, only one of them is a feasible one. In Equation 10,  $T$  is the threshold point and parameters  $A, B$  and  $C$  are computed by Equations 11 through 13, respectively [7].

$$A = \sigma_1^2 - \sigma_2^2 \quad (11)$$

$$B = 2(\mu_1\sigma_2^2 - \mu_2\sigma_1^2) \quad (12)$$

$$C = (\sigma_1\mu_2)^2 - (\sigma_2\mu_1)^2 + 2(\sigma_1\sigma_2)^2 \ln\left(\frac{\sigma_2 P_1}{\sigma_1 P_2}\right) \quad (13)$$

where  $\mu_1$  and  $\sigma_1$  are the mean value and the variance of one side of the bimodal image histogram, and  $\mu_2$  and  $\sigma_2$  are the mean value and the variance of the other side of the bimodal image histogram. Both parameters,  $\mu$  and  $\sigma$ , describe the global properties of the histogram-based measurements.

In order to optimize fitness function  $E$  and find the six parameters in Equation 10, it is essential to use a proper iterative optimization method. Since the corresponding search space may be large, the use of evolutionary methods like GA [9] and PSO [10] is proposed. Because of being nonlinear and having dynamic inertia in every iterative, APSO algorithm has a great ability for dynamic balance between local and global search; therefore, our proposed thresholding algorithm uses APSO for computing the optimal threshold value. Finally the proposed thresholding algorithm is compared with GA and PSO-based thresholding algorithms.

### 3.2 Adaptive PSO

Within the evolutionary algorithms, APSO is a suitable algorithm for optimizing targets because of its population-based property, minor variables, and speedy rate computation [14]. The structure of this algorithm is different from those of other algorithms such as GA, and it is a social method instead of a competition one. Hence, whereas the particles are weak by themselves, they become strong when combined together.

In the conventional PSO, particles that cannot become optimal slowly shift to the location of  $Gbest$ ; therefore, the global optimal particle must actively detect new areas and update  $Gbest$  to give reasons for searching other particles. In this work APSO is used to improve the performance. It is used for different particles by various tasks to find the best of them. In this way, the particles with better performance are weighed by more inertia weight and the particles with poor performance are weighed by smaller one to obtain an optimal point in general.

Therewith, a large inertia weight simplifies a global search while a small inertia weight simplifies a local search. The particles are sorted according to their individual optimal location from the best to the

poorest. The inertia weight and positive constant of the particle, which is in  $i^{\text{th}}$  place, are presented as

$$w_i = w_{\min} + \frac{(w_{\max} - w_{\min}) \cdot (m - i)}{m - 1} \quad (14)$$

$$c_{i1} = c_{i2} = \frac{(w_i + 1 + 2\sqrt{w_i})}{2} \quad (15)$$

In Equation 14,  $m$  is the population size; inertia weight function  $w_i$  is adjusted adaptively. The strategy can balance the global and local search in each iteration step.

APSO algorithm can be summarized as follows:

- 1- Generate an initial random position and velocity for particles
- 2- Compute the fitness function of all particles; calculate the best position solution and the best personal velocity one
- 3- Use Equations 1 and 2 to update the position and velocity for each particle. Inertia weights and learning factors ( $c_{i1}$  and  $c_{i2}$ ) are updated by Equations 14 and 15.
- 4- Repeat levels (2) and (3) until stopping criteria are met.

The selected values for the algorithm are shown below:

Number of populations: 50  
Iteration: 20  
 $w_{\min}$ : 0.4  
 $w_{\max}$ : 0.9  
Stopping Criteria: 1e-6

#### 4. Experimental Results

In order to evaluate the performance of the proposed thresholding algorithm that is based on APSO algorithm, three different thresholding algorithms were compared including Otsu's method [5] as well as GA and PSO-based thresholding algorithms. The fitness function in the last two methods compared is the same as for our proposed algorithm only that they use GA [9] and PSO [10] optimization algorithm.

All compared thresholding algorithms are applied on the USC-SIPI image database [15]. This benchmark image database contains 44 real complex images with different brightness, contrasts and sizes. The USC-SIPI image database is maintained primarily to support research in image processing, image analysis, and machine vision. The database is divided into volumes based on the basic character of the pictures. Images in each volume are of various sizes such as 256×256, 512×512, or 1024×1024 in pixels. All images are 8 bits/pixel for gray-level images, 24 bits/pixel for color images.

The main components in the segmentation algorithms are accuracy and complexity. Accuracy can be measured by some known metrics such as correct detection rate, false acceptance rate and false rejection rate. Again, complexity can be measured by running time in the same conditions. In order to compare the competency of the compared thresholding algorithms, including complexity and accuracy, we have implemented all algorithms in Matlab environment, run on a 1.6 GHz Core 2 Duo CPU, on Windows XP platform.

As mentioned before, to evaluate the performance of the proposed algorithm in comparison with other algorithms, three performance metrics are defined. The first metric is the correct detection rate (CDR) and is given in Equation 16. The false acceptance rate (FAR) is the percentage of identification moments in which false acceptance happens. The false rejection rate (FRR) is the percentage of recognition moments in which false rejection occurs. FAR and FRR are expressed in Equations 17 and 18, respectively:

$$\text{CDR} = \frac{\text{Number of samples correctly classified in dataset}}{\text{Total number of samples in dataset}} \quad (16)$$

$$\text{FAR} = \frac{\text{Number of false accepted samples in dataset}}{\text{Total number of samples in dataset}} \quad (17)$$

$$\text{FRR} = \frac{\text{Number of false rejected samples in dataset}}{\text{Total number of samples in dataset}} \quad (18)$$

Criterion	Otsu [5]	GA [9]	PSO [10]	APSO (our method)
CDR (%)	34.09	79.54	88.63	93.18
FAR (%)	43.18	13.63	6.82	4.54
FRR (%)	22.72	6.82	4.54	2.27
Time (Sec.)	10	71	107	68

Table 1. Average segmentation accuracy and average execution time for different applied methods on USC-SIPI images database, including 44 images.

Table (1) shows the average segmentation accuracy and average execution time on all 44 images in the benchmark dataset, USC-SIPI image database [15], for the compared thresholding algorithms. In order to compute the segmentation accuracy of each image in the dataset, the corresponding best threshold value is obtained manually and the true binary image is used to compute the percentage of thresholding accuracy.

The results show that whereas evolutionary algorithms have less speed than Otsu's method, they reach higher segmentation accuracy than Otsu's method. It is also clear that our proposed thresholding algorithm, which is based on APSO, shows the highest accuracy in comparison with all the algorithms compared while it shows the lowest complexity among all the evolutionary algorithms compared. Although running time in Matlab environment is not a good criterion to compare the complexity of algorithms, we believe APSO shows less complexity because

1- APSO has the least among the evolutionary algorithms employed.

2- The APSO simulation in this paper is implemented by traditional Matlab programming which is not optimized for reducing complexity, whereas the other evolutionary algorithms (GA and PSO) used here are based on Matlab toolbox functions which are optimized for speed and memory usage.

In order to qualitatively compare the results, three sample images are selected from the image dataset. Figures 3 through 5 illustrate the output binary images for all the compared thresholding algorithms.

Figure 3 shows the "peppers" image. Figure 3(a) is the original image and Figure 3(f) is the histogram of the input image. Figures 3(b), (c) and (d) show thresholding with Otsu's method, GA thresholding and PSO thresholding, respectively. It can be seen that the thresholding result by the proposed method, which is shown in Figure 3(f), in addition to being a faster and simplex approach, results in a higher quality than with the other compared methods, in comparison to human-based metrics such as shadows and details.

As for the "boat" image (Figure 4), the thresholded images as shown in Figures 4(b), (c), (d) and (e) are not so different; however, some small parts obtained by the proposed approach (Figure 4(e)) appear more clearly than the others, like shadows of boats and trees in the lake.

Figure 5 shows a specified image which shows the accuracy difference between the proposed method and the others; as it can be seen, just PSO algorithm shows very close results to our algorithm and the thresholded images by using both methods have minor differences.

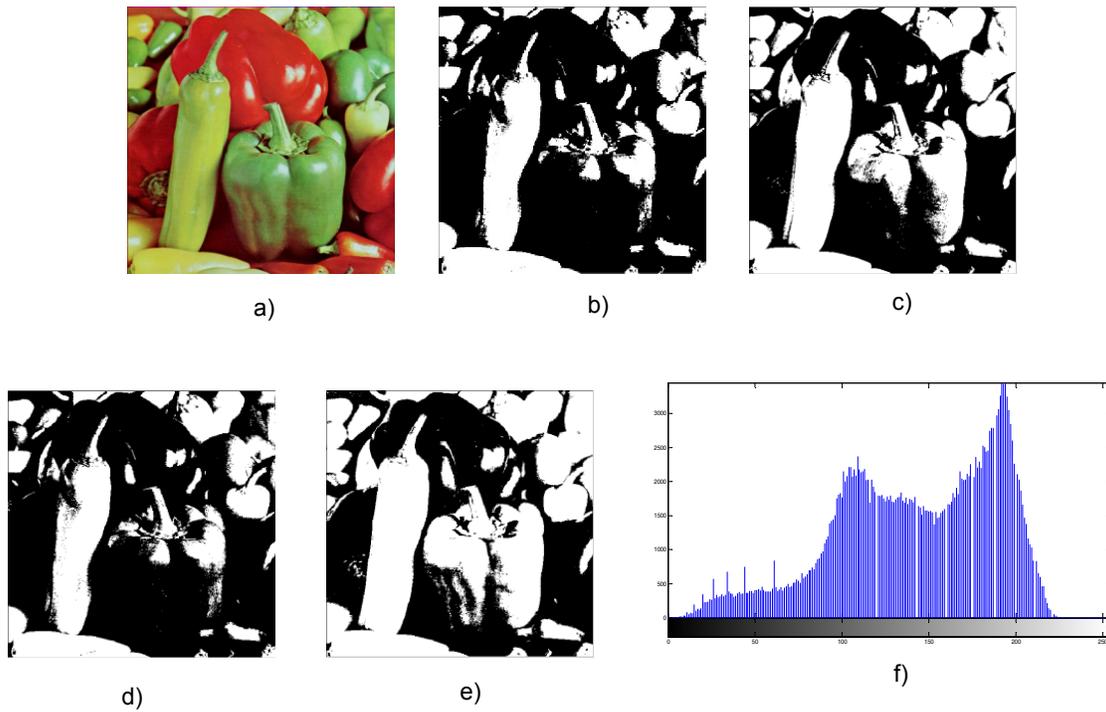


Figure 3. a) Input image (peppers), b) thresholding with Otsu's method, c) GA thresholding, d) PSO thresholding, e) APSO thresholding (our proposed algorithm), f) histogram of the input image.

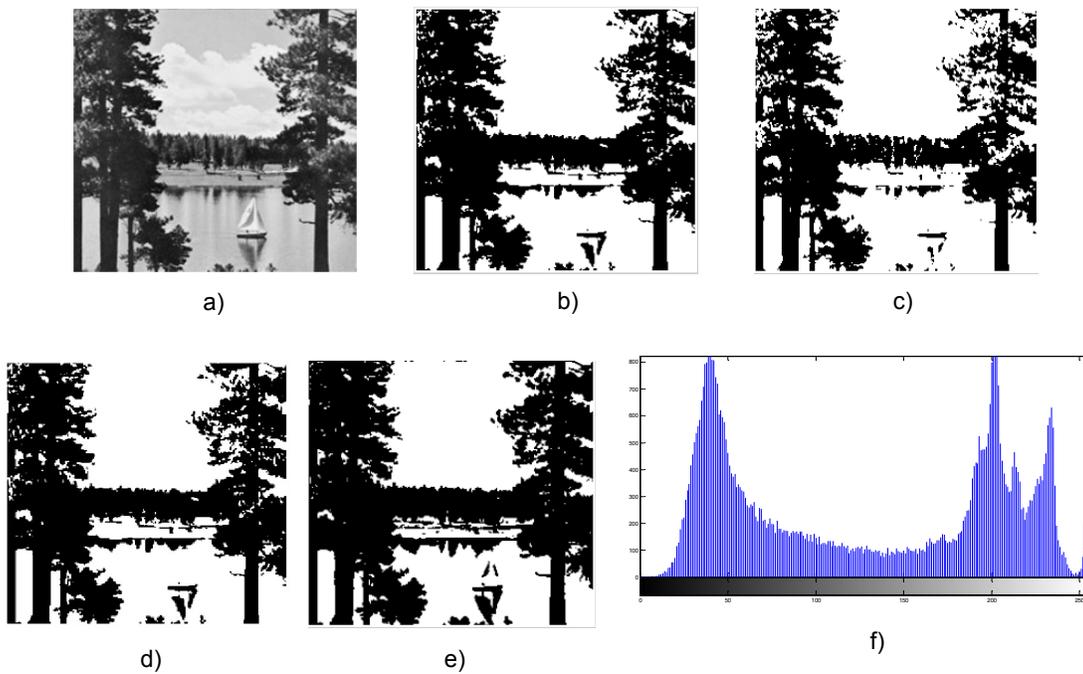


Figure 4. a) Input image (boat), b) thresholding with Otsu's method, c) GA thresholding, d) PSO thresholding, e) APSO thresholding (our proposed algorithm), f) histogram of the input image.

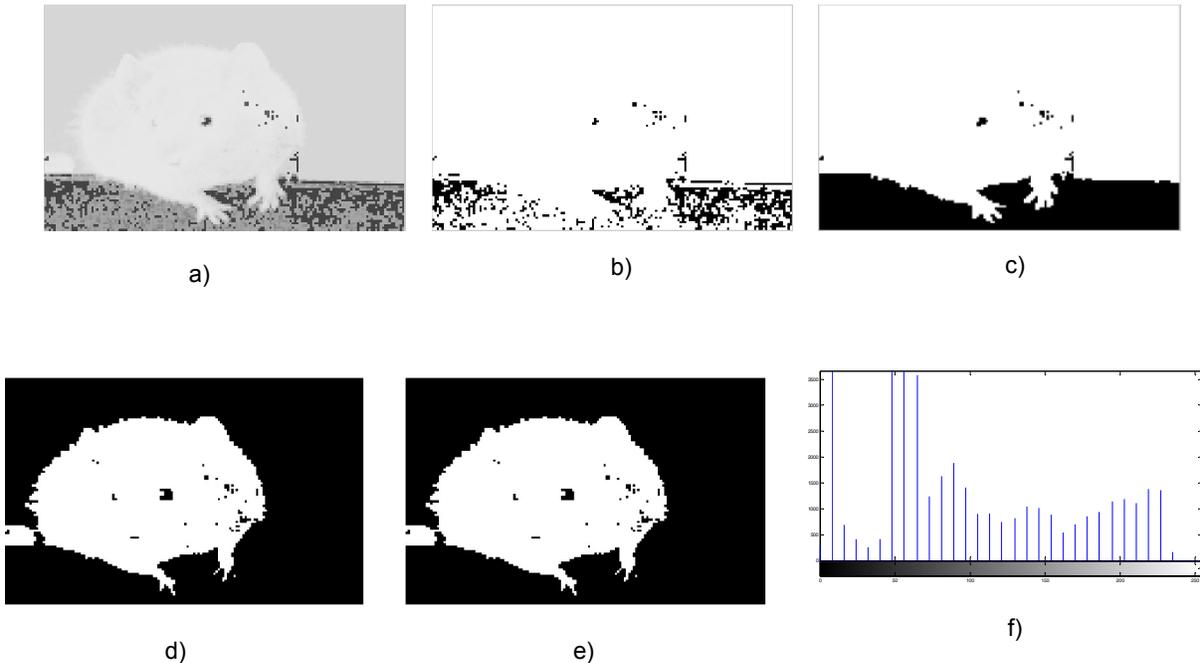


Figure 5. a) Input image (mouse), b) thresholding with Otsu's method, c) GA thresholding, d) PSO thresholding, e) APSO thresholding (our proposed algorithm), f) histogram of the input image.

## 5. Conclusions

Thresholding is one of the most important topics in machine vision and image processing segmentation. Practically, finding a proper threshold is difficult, especially when an overlap happens in the image histogram [9].

In this paper, we apply the adaptive particle swarm optimization algorithm (APSO) to find the threshold value based on the adaptive and local histogram optimization function. We assumed that the intensity distributions of objects and background in an image can be estimated by Gaussian probability density functions. Therefore, the histogram of the given image is fitted by a mixture of two Gaussian probability density functions.

The APSO algorithm is then used to estimate the required parameters so that the difference between the estimated histogram and the image histogram

reaches its minimum value. Then, the optimal threshold is computed by some straightforward relationships. Experimental results on the benchmark image dataset show that the computational cost of the proposed algorithm is lower than the threshold computation based on the PSO and GA, while the correct segmentation rates of the proposed algorithm are also the highest among all compared thresholding methods, while the false acceptance rate and the false rejection rate are lowest.

## References

- [1] E. D. López-Espinoza and L. Altamirano-Robles, "Reference Fields Analysis of a Markov Random Field Model to Improve Image Segmentation," *Journal of Applied Research Technology*, vol. 8, no.2, pp. 260-273, 2010.

- [2] G. J. Chhajed et al., "Review on Binary Image Steganography and Watermarking," *International Journal on Computer Science and Engineering*, vol. 3, no. 11, pp. 3645-3651, 2011.
- [3] M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," *Journal of Electronic Imaging*, vol. 13, no. 1, pp. 146-165, 2004.
- [4] Y. E. Fernandez et al., "development of a prototype for classification of potato mini-tubers based on artificial vision," 6<sup>th</sup> International Conference on Electrical Engineering, Computing Science and Automatic Control, Toluca, 2009.
- [5] P. P. Kumar et al., "A Novel Stroke Width Based Binarization Method to Handle Closely Spaced Thick Characters," *International Journal of Computer Applications*, vol. 1, no. 23, pp. 32-39, 2010.
- [6] Z. K. Huang and K. W. Chau, "A New Image Thresholding Method Based on Gaussian Mixture Model", *Applied Mathematics and Computation*, vol. 205, no. 2, pp. 899-907, 2008.
- [7] R. Gonzalez et al., "Digital Image Processing," 3<sup>rd</sup> Edition, Prentice Hall, 2008.
- [8] L. M. Palanivelu and P. Vijayakumar, "A Particle Swarm Optimization for Image Segmentation in Multi Application Smart Cards," *European Journal of Scientific Research*, vol. 70, no. 3, pp. 354-360, 2012.
- [9] P. Kanungo et al., "Image Segmentation Using Thresholding and Genetic Algorithm," *Conference on Soft Computing Technique for Engineering Applications*, Rourkela, India, 2006.
- [10] C. C. Lai, "A Novel Image Segmentation Approach Based on Particle Swarm Optimization," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E89-A, no. 1, pp. 324-327, 2006.
- [11] S. Cheng et al., "Experimental Study on Boundary Constraints Handling in Particle Swarm Optimization: From Population Diversity Perspective," *International Journal of Swarm Intelligence Research*, vol. 2, no. 3, pp. 43-69, 2011.
- [12] P. Moallem and N. Razmjooy, "A Multi Layer Perceptron Neural Network Trained by Invasive Weed Optimization for Potato Color Image Segmentation," *Trends in Applied Sciences Research*, vol. 7, no. 6, pp. 445-455, 2012.
- [13] P.D. Sathya and R. Kayalvizhi, "PSO-Based Tsallis Thresholding Selection Procedure for Image Segmentation," *International Journal of Computer Applications*, vol. 5, no.4, pp.39-46, 2010.
- [14] J. Zhang et al., "Bidding Strategy Based on Adaptive Particle Swarm Optimization for Electricity Market," 8<sup>th</sup> World Congress on Intelligent Control and Automation, Jinan, China, 2010.
- [15] The USC-SIPI Image Database, Available from: <http://sipi.usc.edu/database/>