Using The ACO Algorithm in Image Segmentation for Optimal Thresholding

陳香伶
財務金融系

Abstract

Despite the fact that the problem of thresholding has been quite extensively studied for many years, the automatic determination of an optimum threshold value continues to be of great challenge. For one, traditional optimal thresholding methods exhaustively search the optimal thresholds to optimize the object functions therefore become very computationally expensive when dealing with multi-level thresholding. Hence, our study proposes a hybrid optimization scheme using ant colony optimization algorithm to render the optimal thresholding technique more applicable and effective. We employed the properties of discriminate analysis using Otsu's method to analyze the separability among the gray levels in the image. The ACO-Otsu algorithm, considered a non-parametric and unsupervised method of automatic threshold selection for image segmentation, has several desirable advantages. The experimental results show that the ACO-Otsu efficiently speed up the Otsu's method to a great extent at multi-level thresholding, and that such method can provide better effectiveness at population size of 20 for all given image types at multi-level thresholding in this study.

1. Introduction

Many applications such as document image analysis, map processing, scene processing, computer vision, pattern recognition, and quality inspection of materials consider the image thresholding technique a crucial operation because further process steps have to rely on the segmentation results. The widely-used technique which extracts the objects from the background has both bi-level and multilevel types recognized. For an image with clear objects in the background, the bi-level thresholding which divides the object pixels at one gray level while the background pixels at another is widely used. For rather complex images, on the other hand, the multilevel thresholding segments the pixels into several distinct groups in which the pixels of the same group have gray levels within a specific range. In recent practices, the multilevel thresholding has been much accepted, yet the complexity of the thresholding problem and the computation time to solve such problem still impose significant challenges as the number of levels required increases. For this reason, many thresholding techniques have been proposed and classified. Some techniques are identified as either global or local thresholdings based on the role of the intensity value while other methods have been classified as either optimal or property-based.

Kapur *et al.* [1985] used the concept of the entropy of a histogram and developed a global thresholding method separating the histogram of gray level probabilities into two distributions of the image. Such method was also considered an optimal-based thresholding which maximized the combined entropy of the thresholded classes to determine the optimal threshold value. Yin [1993] proposed a property-based method which first developed a peak-finding method based on symmetry; then, the duality property was used to identify the valleys of the histogram. Perhaps, the most important and widely-accepted concept is on the characteristics of thresholding techniques. Sahoo *et al.* [1988] presented a comprehensive survey of a variety of thresholding techniques and Abutaleb [1989] classified them into parametric or non-parametric approaches. Parametric approaches assume each group having the probability density function of a Gaussian distribution and find an estimate of the parameters of such distribution which will best fit the given histogram data [Tsai 1995]. Unfortunately, when the desired number of classes is much lower than the number of peaks in the original histogram, the computation time to find the solutions of threshold values often becomes expensive. Different from parametric approaches, non-parametric methods find the threshold level according to some discriminating criteria such as the between-class variance Otsu [1979] and entropy Kapur *et al.* [1985] which both separate the gray-level regions of an image in an optimum manner. As the result, the non-parametric approaches are proven to be more computationally efficient and simpler to apply.

Despite the fact that the problem of thresholding has been quite extensively studied for many years, the automatic determination of an optimum threshold value continues to be of great challenge. For one, traditional optimal thresholding methods exhaustively search the optimal thresholds to optimize the object functions therefore become very computationally expensive when dealing with multi-level thresholding. Till recent years, several researchers attempt to using heuristics as alternative ways to solve multi-level thresholding. Yin [1999] proposed a fast scheme for optimal thresholding using genetic algorithms. His method, an optimal thresholding technique, has shown better performance than those of some property-based ones. Cheng *et al.* [2000] applied fuzzy entropy in image segmentation, used it to select the fuzzy region of membership function automatically so that an image can be transformed into fuzz domain with maximum fuzzy entropy, and implemented genetic algorithm to find the optimal combination of fuzzy parameters. Zahara *et al.* [2005] presented a hybrid optimization scheme which applied the Otsu's method with Nelder-Mead simplex search and partical swarm optimation (the NM-PSO-Otsu method) and proven to not only expedite the Otsu's method efficiently but also extent its effectiveness to a multi-level thresholding problem.

In this paper, a fast scheme using ant colony optimization algorithm is proposed to render the optimal thresholding technique more applicable and effective. We employed the properties of discriminate analysis using Otsu's method to analyze the separability among the gray levels in the image. This method, considered a non-parametric and unsupervised method of automatic threshold selection for image segmentation, has several desirable advantages. First, the Otsu's method is very simple and straightforward to the multi-thresholding problems. Also, an optimal threshold can be selected automatically based on the global property of the histogram (i.e. the between-class variance). Second of all, our study considers the ant colony optimization (ACO) algorithm to find the optimal threshold values because the ACO algorithm itself imposes several valuable features. For examples, a stochastic component allows the artificial ants to build a wide variety of different solutions and hence explore a much larger number of solutions. Also, the heuristic information is applied to guide and influence the ants moving and learning towards the most promising solutions. As well as, the use of a colony of ants acts as the collective interaction of a population of search agents which increase the algorithm's robustness and efficiently solve a problem. For these reasons, the ACO algorithm has been successfully employed to solve various combinatorial optimization problems. Nevertheless, it is also our curiosity to apply and evaluate the ACO algorithm in the field of image segmentation.

2. Otsu's Method for Image Thresholding

The concept of using discriminate analysis for classification problems was first introduced by Fisher [1936] and was applied on image thresholding by Otsu [1979]. In Otsu's paper, the elementary case of threshold selection where only the gray-level histogram suffices without other a priori knowledge is discussed, and their method is proposed from the viewpoint of discriminate analysis. The feasibility of evaluating the "goodness" of threshold is done through exhaustive search to maximize the between class variance between dark and bright regions of the image. Our study uses the extended properties of the discriminate criterion to determine the number of objects into which the image should be segmented, and describes the concept of an automatic multilevel thresholding method as follows.

For multi-level thresholding, a gray level image $f(x,y)$ is transformed to a multi-level image $g(x,y)$ by a threshold set $T=\{t_1,t_2,..,t_n,....,t_k\}$, which is composed of k thresholds. With a given gray level i, n_i denote the observed occurrence frequencies (histogram) of pixels and the total number of pixels $N=n_1+n_2+....+n_L$ where L is the number of gray values in the histogram. Then the gray-level histogram is normalized and regarded as a probability distribution having a given gray level i:

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

Suppose we segment these pixels into a suitable number of classes. With k denoting the number of selected thresholds (i.e. $0\leq k\leq L-1$), the image is then partitioned into k+1 classes which can be represented by $C_0=\{0,1,...,t_1\}$,.... $C_n=\{t_n+1,t_n+2,...t_{n+1}\}$,.... $C_k=\{t_k+1,t_k+2,...L-1\}$. Hence, the probabilities of class occurrences (w_n), the class-mean levels (μ_n), and the class variances (σ_n^2) are given as follows, respectively:

$$w_n = \sum_{i=t_{n+1}}^{t_{n+1}} p_i \ , \quad \mu_n = \frac{\sum_{i=t_{n+1}}^{t_{n+1}} ip_i}{w_n}, \quad \text{and} \quad \sigma_n^2 = \frac{\sum_{i=t_{n+1}}^{t_{n+1}} p_i (i-\mu_n)^2}{w_n}$$

The within-class variance, denoted by σ_{WC}^2, of all segmented classes of pixels is computed as,

$$\sigma_{WC}^2 = \sum_{n=0}^{k} w_n \sigma_n^2$$

The between-class variance, denoted by σ_{BC}^2, is used to measure the separability among all classes and is expressed as,

$$\sigma_{BC}^2 = \sum_{n=0}^{k} w_n (\mu_n - \mu_T)^2$$

The total variance σ_T^2 and the overall mean μ_T of pixel in a given gray level image $f(x,y)$ are computed as,

$$\sigma_T^2 = \sum_{i=0}^{L-1} (i-\mu_T)^2 p_i \ , \text{and} \quad \mu_T = \sum_{i=0}^{L-1} ip_i$$

In order to evaluate the "goodness" of the threshold at level *k*, the following discriminate criterion measures are used:

$$\lambda = \frac{\sigma_{BC}^2}{\sigma_{WC}^2}, \quad \kappa = \frac{\sigma_T^2}{\sigma_{WC}^2}, \quad \text{and} \quad \eta = \frac{\sigma_{BC}^2}{\sigma_T^2}$$

Among these measures, the parameter η is the simplest one with respect to *k*, and therefore the optimal threshold k^* that maximizes η, or equivalently maximizes σ_{BC}^2 is also followed as,

$$\sigma_{BC}^2(k_1^*, k_2^*, .., k_L^*) = \max_{1 \le k_1 < k_2 < L} \sigma_{BC}^2(k_1, k_2, ..k_L)$$

3. Ant Colony Optimization (ACO) Algorithm

Just like other meta-heuristics inspired by the natural process, the Ant Colony Optimization (ACO) algorithm is imitating the behavior of real ants. In ACO, a colony of simple agents, called artificial ants, search for good solutions at every generation. Every artificial ant of a generation builds up a solution step by step. These ants, once build a solution, will evaluate the partial solution and deposit some amount of pheromone to mark their paths. The following ants of the next generation are attracted by the pheromone so that they will likely search in these areas nearby. The ACO algorithm has its general framework like below.

Set all parameters and initialize the pheromone trails
Loop (no. of iterations)
Sub-Loop (population size, *popsize*)
Build solutions based on **the state transition probability**
Continue until all ants have been generated
Evaluate all solutions during the iteration and select the best one
Apply the **pheromone update** rule
Continue until **the stopping criterion** is reached

For activity selection, the ***state transition probability*** shown below is used in the solution construction process.

$$p_{ij} = \begin{cases} \dfrac{(\tau_{ij})^{\alpha}}{\sum_{l \in \{1,2,..,UB_i - LB_i + 1\}} (\tau_{il})^{\alpha}} & j \in \{1,2,..,UB_i - LB_i + 1\} \\ 0 & otherwise \end{cases}$$

where i denotes the index of the threshold at multi-level (e.g. i = 1 for bi-level, i = 2 for tri-level, and i = 3 for four-level), j refers to the index for the gray level ranging from the lower bound and the upper bound of the ith threshold, and α denotes the parameter controlling the relative weight of pheromone. The pheromone trails, denoted by τ_{ij} are constantly updated according to the pheromone updating rule.

The pheromone update rule consists of the *offline* updating which is formally expressed as $\tau_{ij}^{new} = \rho \cdot \tau_{ij}^{old} + (1-\rho) \cdot \Delta\tau^{e}$, where a parameter $\rho \in [0,1]$ controls the pheromone persistence and $1-\rho$ represents the proportion of the pheromone evaporated. Meanwhile $\Delta\tau^{e}$ represents the amount of pheromone trail added to τ_{ij} by the elitist ants for all combinations (i, j) belonging to the best solution found so far, and is determined by $\Delta\tau^{e} = Q \times \sigma_{BC}^{2}$ where a constant Q controls the magnitude of the pheromone contribution.

4. Computational Results and Analysis

In this section, the performance of the proposed ACO-Otsu algorithm is evaluated and compared to the Otsu's method with exhaustive search and the NM-PSO-Otsu algorithm [Zahara 2005]. Our study deals with a binary image which is produced by thresholding a grayscale image at multi-levels. These test images were taken under natural room lighting without the support of any special light source. As it is commonly understood, the images are composed of a collection of discrete cells, known as pixels, which has given values ranging from 0 to 255, or 1 to 256. So at the beginning of our experiments on the parameter setting for the ACO-Otsu method, we implemented three standard images with rectangular objects of uniform gray values (see Figure 1-a, Figure 2-a, and Figure 3-a). As well as, we employed another three test images – "Dragon", "Screws", and "Blocks" (shown in Figure 2-a to 2-c, respectively) to analyze and evaluate the performance of our algorithm.

The ACO-Otsu method is implemented on a Pentium IV 3.0GHz, 768 MB personal computer using C++ programming language while the Otsu method with NM-PSO-Otsu methods [Zahara *et al.* 2005] were implemented on a Athlon XP 2200+ (166×11) with 1 GB RAM using Matlab. A stopping criterion used in Zahara *et al.* [2005] and our studies are the maximum number of iterations reached when solving an N-dimensional problem. Our preliminary experiments have shown the following set of all parameters to account for both efficiency and effectiveness; therefore, we set up as follows: $\alpha = 1$, $\tau_0 = 0.01$, $\rho = 0.9$, and $Q = 0.01\tau_0$ for all experimental runs. Table 1 lists out different parameter settings in our preliminary experiments.

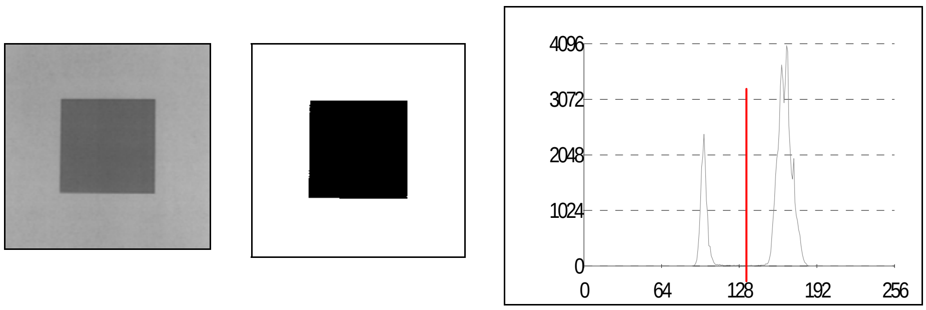
Table 1. Settings of Different Parameters Implemented in the ACO-Otsu algorithm

Parameters	Values		
α	0.5	**1**	2
ρ	0.1	0.5	**0.9**
Q	10^{-3}	$\mathbf{10^{-4}}$	10^{-5}
$\boldsymbol{\tau_0}$	10^{-1}	$\mathbf{10^{-2}}$	10^{-3}

The experiment starts with three standard test images (shown in Figures 1-a, 2-a, and 3-a, respectively) with rectangular objects of uniform gray values. Then the resulting images of the bi-level, tri-level, and four-level (shown in Figure 1-b, 2-b, and 3-b, respectively) verify that ACO-Otsu method can provide a quality performance in image segmentation. Comparison results (see Table 2) for these three standard test images have reveal identical optimal threshold values for both Otsu's and NM-PSO-Otsu methods in [Zahara *et al.* 2005], but slightly different values for our ACO-Otsu method. The appealing difference in optimal threshold values perhaps comes from using different gray-level scales: (0, 255) in [Zahara *et al.* 2005] while (1,256) in this study. In addition, for both Otsu's and NM-PSO-Otsu methods in [Zahara *et al.* 2005], the optimal objective values greatly differs from our optimal objective values in ACO-Otsu method of this study (shown in Table 2). The reason has been that Zahara *et al.* [2005] minimized the within-group variance in the objective function while our study maximized the between-class variance.

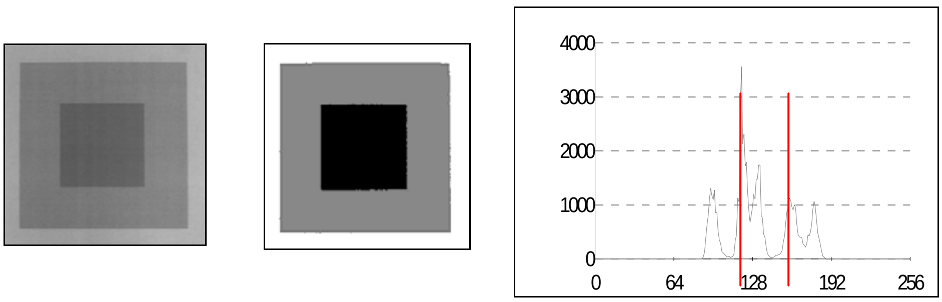
Table 2. Computational Results for the three standard test images at the multi-level thresholding

Standard Test Images	Optimal Thresholds		Optimal Objective Values (over 10 runs)	
	Otsu and NM-PSO-Otsu	ACO-Otsu	Otsu and NM-PSO-Otsu	ACO-Otsu
Bi-level thresholding	133	132	29.85	737.89
Tri-level thresholding	111, 146	110, 145	54.11	622.72
Four-level thresholding	71, 114, 150	70, 113, 149	36.93	967.70



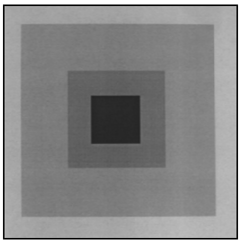
Original image (a) T=132 (b) Histogram (c)

Figure 1. Bi-level thresholding test image: (a) original image, (b) ACO-Otsu method, and (c) histogram of (b)

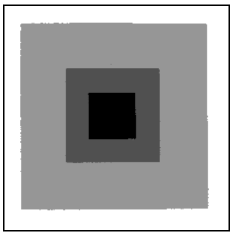


Original image (a) T=110, 145 (b) Histogram (c)

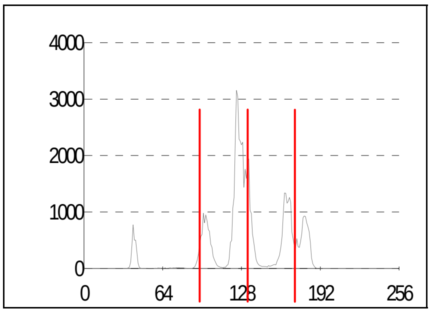
Figure 2. Tri-level thresholding test image: (a) original image, (b) ACO-Otsu method, and (c) histogram of (b)



Original image
(a)



T=70, 113, 149
(b)



Histogram
(c)

Figure 3. Four-level thresholding test image: (a) original image, (b) ACO-Otsu method, and (c) histogram of (b)

Table 3. Result comparisons among Otsu's, NM-PSO-Otsu and ACO-Otsu methods over the three standard test images

Standard Test Images	CPU Times (sec.)			Population Sizes (NA) × Iteration (NI)	
	Otsu	NM-PSO-Otsu	ACO-Otsu	Otsu and NM-PSO-Otsu	ACO-Otsu
Bi-level thresholding	0.000	0.000	0.009	4 × 10	10 × 10
Tri-level thresholding	0.281	0.015	0.044	7 × 20	20 × 20
Four-level thresholding	17.828	0.031	0.214	10 × 30	20 × 60

From the computational results listed in Table 3, we can generally conclude that the CPU time goes linearly up with the increasing level of thresholding; furthermore, the higher levels of thresholding will give rise to the population size and iteration. We also see that as the higher level of thresholding leads to the increasing computational complexity, our CPU time also increases from the lowest the lowest CPU time of 0.009 seconds at NA=10 and NI=10 (i.e. No. of evaluations = 100) to the highest CPU time of 0.214 seconds at NA=20 and NI=60 (i.e. No. of evaluations = 1200). When we compare with the Otsu's method, our ACO-Otsu method takes relatively less execution times to achieve 100% optimum but not when we compare with NM-PSO-Otsu method.

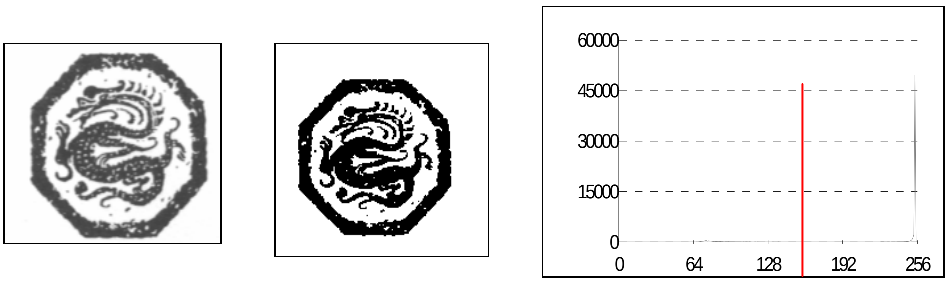
For evaluating the performance of our proposed method, three images (Dragon, Screws, and Blocks) are chosen; the threshold selection values and computation times for these three tested images are automatically determined according to different threshold levels. These results as listed in Table 4. Also Table 5 shows our comparison results of "Screws" image with the ones in [Zahara *et al.* 2005] and the ACO-Otsu takes relatively longer CPU time than the other two methods. However, we see that our method is capable of obtaining quality image segmentation results for the tri-level thresholding within a significantly short period of time and (shown in Figure 4 to Figure 6). For the 'Screws' and 'Blocks' test images (shown in Figures 5 and 6 below), both pictorial results appear to be better at the tri-level thresholding than at the bi-level thresholding.

Table 4. Computational results for images of Dragon, Screws, and Blocks at multi-level thresholdings

Images	Level	Optimal Thresholds	Optimal Objective Values	CPU Times (sec.)	Population Sizes (NA) × Iteration (NI)
Dragon	2	175	2615.11	0.068	20 × 20
Screws	2	210	338.82	0.042	20 × 20
	3	194,226	393.68	0.166	20 × 60
Blocks	2	201	230.36	0.009	5 × 20
	3	196, 226	274.12	0.106	20 × 40

Table 5. Computational results for Test Images of "Screws" at bi-level thresholding

"Screws" test image	Opt. Thresholds	Opt. Obj. Values	CPU Times (sec.)	Pop Sizes (NA)	Iteration (NI)
Otsu [Zahara *et al.* 2005]	210	90.05	0.000	4	10
NM-PSO-Otsu [Zahara *et al.* 2005]	210	90.05	0.000	4	10
ACO-Otsu [This study]	209	338.82	0.007	10	5



Original image (a) T=175 (b) Histogram (c)

Figure 4. Result of Original Image of "Dragon": (a) original image, (b) ACO-Otsu method, and (c) histogram of (b)

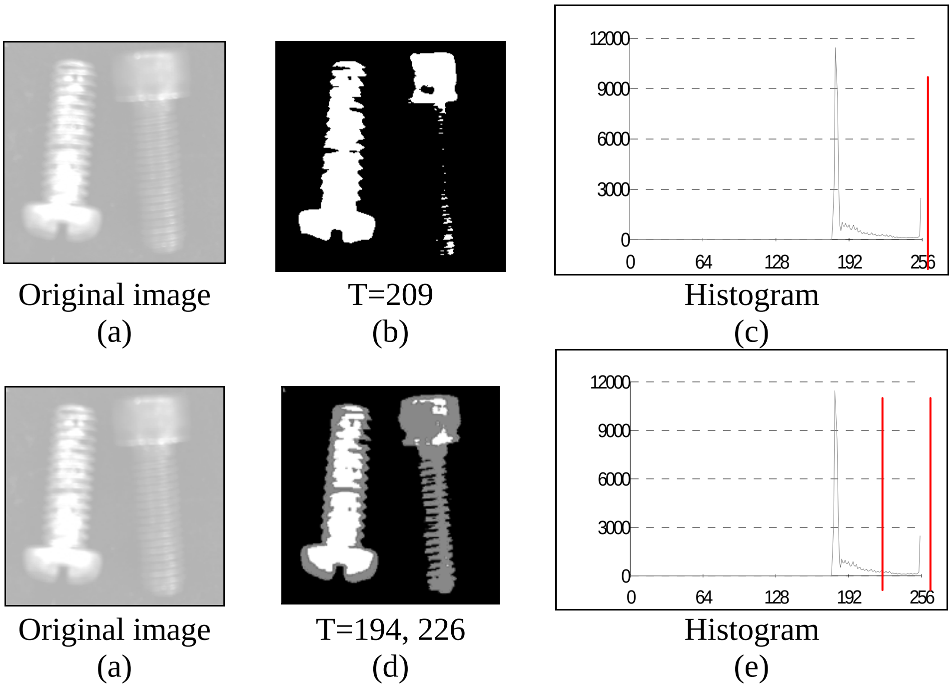


Figure 5. Thresholding Results of Original Image of “Screws”: (a) original image, (b) ACO-Otsu method at bi-level, (c) histogram of (b), (d) ACO-Otsu method at tri-level, and (e) histogram of (d).

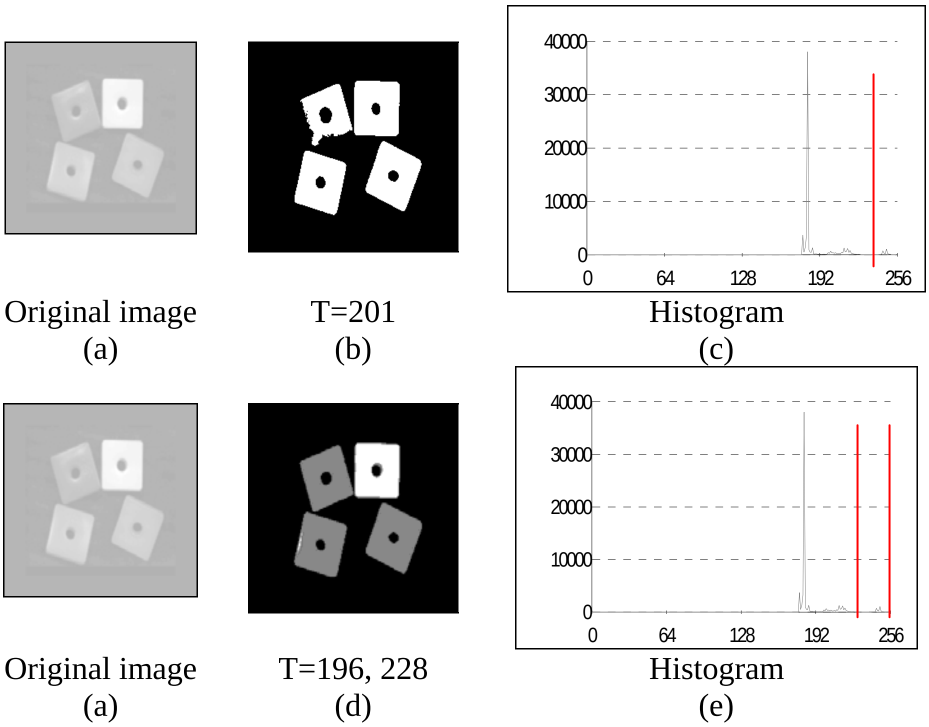


Figure 6. Result of Original Image of “Blocks”: (a) original image, (b) ACO-Otsu method at bi-level, (c) histogram of (b), (d) ACO-Otsu method at tri-level, and (e) histogram of (d).

5. Conclusion

In this study, we can draw several general conclusions at the end of this analysis. First of all, from our preliminary experiment, we have found that NA and NI have an inverse relation for all levels of thresholding. In other word, when NA increases, NI decreases; when NA decreases, NI increases. Second of all, we also find that when the level of thresholding increases, both NA and NI will increase. Then, the computational results for most images at bi-, tri-, and four-level thresholdings have shown number of iterations (NI) appears to be rather effective at population size (NA) of 20, NA regardless the complexity of the images. From the above findings, while the quality of image segmentation does not get compromised, we consider the ACO-Otsu method to be a potential method to accelerate the Otsu's method in multi-level thresholding for real-time applications.

Reference

Abutaleb A.S. (1989). Automatic thresholding of gray-level pictures using two-dimensional entropy, Computer Vision, Graphics, and Image Processing, 47, 22-32.

Fisher R.A. (1936). The use of multiple measurements in taxonomic problems, Annals of Eugenics, 7, 179-188.

Kapur J. N., Sahoo P. K., Wong A.K.C. (1985). A new method for gray-level picture thresholding using the entropy of the histogram, Computer Vision graphics Image Process, 29, 273-285.

Kittler J., Illingworth J. (1986). Minimum error thresholding, Pattern Recognition, 19, 41-47.

Lim Y. K., Lee S.U. (1990). On the color image segmentation algorithm based on the thresholding and the fuzzy c-means techniques, Pattern Recognition, 23, 935-952.

Mitra A. (2005). Restoration of noisy document images with an efficient bi-level adaptive thresholding, International Journal of Computer Intelligence, 2, 1304-2386.

Otsu N. (1979). A threshold selection method from gray-level histograms, IEEE Trans. Systems. Man Cybernet, SMC-9, 62-66.

Pun T. (1980). A new method of grey-level picture thresholding using the entropy of the histogram, Signal Processing, 2, 223-237.

Pun T. (1981). Entropy thresholding: a new approach, Computer Vision Graphics Image Process, 16, 210-239.

Tsai D.M., Chen Y. H. (1992). A fast histogram-clustering approach for multi-level thresholding, Pattern Recognition Letters, 13, 245-252.

Tsai D.M. (1995). A fast thresholding selection procedure for multimodal and unimodal histograms, Pattern Recognition Letter,16, 653-666.

Wang S. and Haralick R. (1984). Automatic threshold selection, Computer Vision Graphic Image Processing, 25, 46-47.

Yin P.-Y., Chen L.-H. (1993). New method for multilevel thresholding using the symmetry a duality of the histogram, Journal of Electronic Imaging, 2, 337-344.

Yin P.-Y., Chen L.-H. (1997). A fast iterative scheme for multi-level thresholding methods, Signal Processing, 60, 305-313.

Yin P.-Y. (1999). A fast scheme for optimal thresholding using genetic algorithms, Signal Processing, 72, 85-95.

Zahara E., Fan S.-K. S., Tsai D.M. (2005). Optimal multi-thresholding using a hybrid optimization approach, Pattern Recognition Letters, 26, 1082-1095.