



Title	Adaptive thresholding by variational method
Author(s)	Chan, FHY; Lam, FK; Zhu, H
Citation	IEEE Transactions on Image Processing, 1998, v. 7 n. 3, p. 468-473
Issued Date	1998
URL	http://hdl.handle.net/10722/42774
Rights	Creative Commons: Attribution 3.0 Hong Kong License

Correspondence

Adaptive Thresholding by Variational Method

Francis H. Y. Chan, F. K. Lam, and Hui Zhu

Abstract—When using thresholding method to segment an image, a fixed threshold is not suitable if the background is uneven. Here, we propose a new adaptive thresholding method using variational theory. The method requires only one parameter to be selected and the adaptive threshold surface can be found automatically from the original image.

Index Terms—Adaptive thresholding, image segmentation, variational method.

I. INTRODUCTION

Thresholding techniques are often used to segment images consisting of bright objects against dark backgrounds or *vice versa* [3], [9], [12]. When the background is uneven as a result of poor or nonuniform illumination conditions, a fixed (or global) gray-level threshold will not segment the image correctly. A way to deal with such cases is to match the background gray-level variations using a simple function such as a plane or biquadratic [4]. A more sophisticated method is to create a threshold surface so that different thresholds could be used for each individual pixel in the image [1]. In this method, first the image was divided into a regular grid of nonoverlapping subregions and a threshold was assigned to the center of each subregion which had bimodal histogram by a fixed thresholding method. Then the threshold surface was interpolated from these local threshold values. This method was further studied by Nakagawa and Rosenfeld with some new experiments [7]. Milgram *et al.* [5] considered using gradient or edge information to segment images and proposed “superslice” method. They assumed an individual object had a fixed threshold with respect to the background, but different objects may have different thresholds. Various of gray-level thresholds were tried to segment the image and the segmented objects were validated by the gradients along their boundaries. Encompassing advantages of both the “threshold surface” and the “superslice” methods, Yanowitz and Bruckstein suggested using the gray-level values at high gradient places as known data to interpolate the threshold surface of the image [14]. Their method is a heuristic algorithm that consists of the following seven steps.

- 1) Smooth the image by average filtering.
- 2) Derive the gray-level gradient magnitude.
- 3) Apply thresholding and a thinning algorithm to the gradient magnitude to find the object boundary points.
- 4) Sample the smoothed image at the boundary points as the local thresholds.
- 5) Interpolate the threshold surface by the sampled local thresholds.
- 6) Segment the image by the threshold surface.
- 7) Remove the “ghost” objects by a validation process.

Manuscript received October 30, 1996; revised March 10, 1997. This work was supported in part by research grants from The University of Hong Kong. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Guillermo Sapiro.

The authors are with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong (e-mail: fhychan@eee.hku.hk).
 Publisher Item Identifier S 1057-7149(98)01744-8.

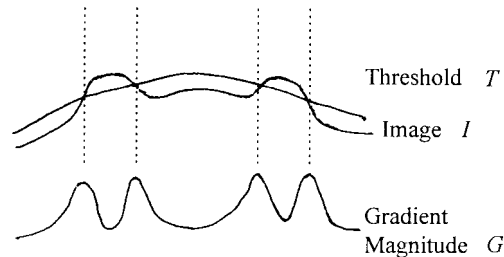


Fig. 1. Adaptive threshold surface to segment an image.

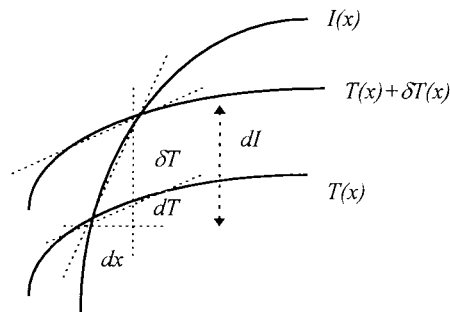


Fig. 2. Relationships among δT , dI , and dT .

The third and fourth steps are the keys to this method. The choices of the gradient magnitude threshold and thinning algorithms are difficult to determine. These factors influence the positions of the boundary point, and consequently affect the interpolation results.

Regularization approaches that use variational principles have been widely used in image processing [10]. Morel and Solimini [6] summarized that most of the heuristic segmentation algorithms can be translated into variational theory. They favored the translation by arguing that “the advantage of an *a priori* variational formulation is to reduce the number of parameters and thresholds and to fix them automatically.” But thresholding algorithms, an important class of image segmentation methods, were not mentioned in their work.

In this work, we will propose a new adaptive thresholding method by variational method. It can be viewed as a variational translation of Yanowitz and Bruckstein’s algorithm. But our method can obtain the threshold surface by deforming the original image gray-level surface so that steps 3 and 4 of [14] are integrated and all the uncertain decisions in step 3 are reduced to the choice of a regularization parameter. In next section, we will give the model of the adaptive threshold surface. In Section III, the implementation algorithm is presented. Experimental results are shown with some discussions in Section IV, and Section V is the conclusion.

II. VARIATIONAL FORMULATION OF ADAPTIVE THRESHOLDING

Let $I(x, y)$ be an image which is to be segmented by thresholding. $G(x, y)$ is the normalized gradient magnitude of $I(x, y)$.

$$G(x, y) = \frac{|\nabla I(x, y)|}{\max_{x, y} (|\nabla I(x, y)|)} \quad (1)$$

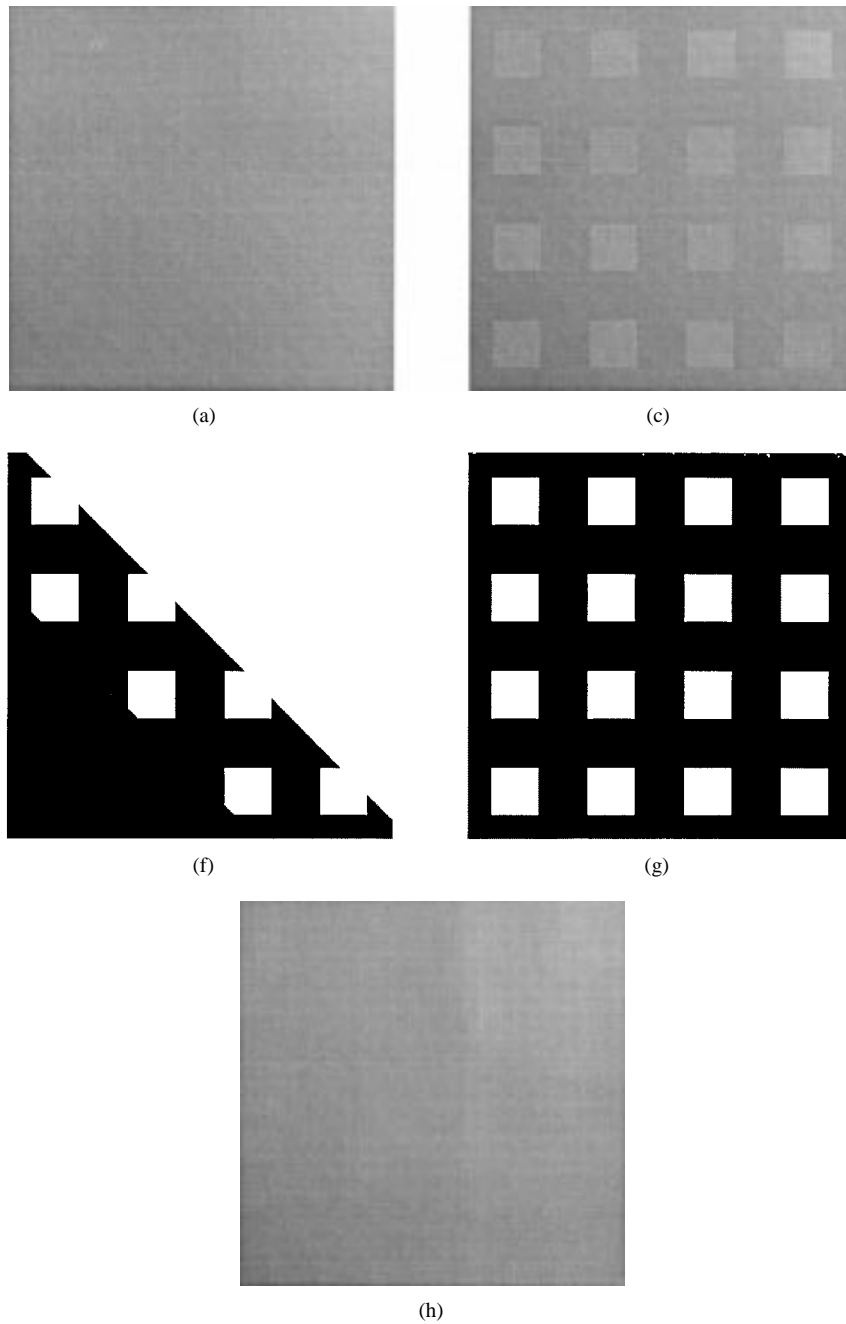


Fig. 3. (a) Simulated image background. (c) 16 objects being put on (a) by adding 20 gray levels to their local background. (f) Result of Otsu's algorithm to (c). (g) Result of our method to (c). (h) Adaptive thresholding surface found by our method.

$T(x, y)$ is the adaptive threshold to be found by which the image is segmented into a label image

$$L(x, y) = \begin{cases} 1, & \text{if } I(x, y) > T(x, y) \\ 0, & \text{if } I(x, y) \leq T(x, y) \end{cases} \quad (2)$$

$I(x, y)$ and $T(x, y)$ are two two-dimensional (2-D) surfaces which intersect at positions $c = \{(x, y) | I(x, y) = T(x, y)\}$. According to [14], c should be where the object boundaries are (see Fig. 1). That means if $T(x, y)$ is the right threshold surface, $\int_c G(x, y) dH^1(x, y)/C$ (H^1 is the Hausdorff measure supported by c [6], and C is the number of object boundary points) should be a maximum. This necessitates in finding T in a function space Ω to

$$\min_{T \in \Omega} \int_c F(T) dH^1(x, y)/C \quad (3)$$

in which, $F(T)$ is a functional of T and has a first derivative and $F(T) = -G(x, y), (x, y) \in c$. Obviously, this is an ill-posed problem.

To solve the problem, a penalty term should be introduced into the object function for regularization. Then the object function to be minimized is

$$\frac{1}{C} \int F(T) dH^1(x, y) + \lambda \int \left(\left| \frac{\partial T}{\partial x} \right|^2 + \left| \frac{\partial T}{\partial y} \right|^2 \right) dx dy \quad (4)$$

where λ is the regularization parameter. Minimizing (4) is equal to solving the following Poisson equation:

$$\begin{cases} \nabla^2 T(x, y) = \alpha \frac{\delta F(T)}{\delta T}, & \text{if } (x, y) \in c \\ \nabla^2 T(x, y) = 0, & \text{else} \end{cases} \quad (5)$$

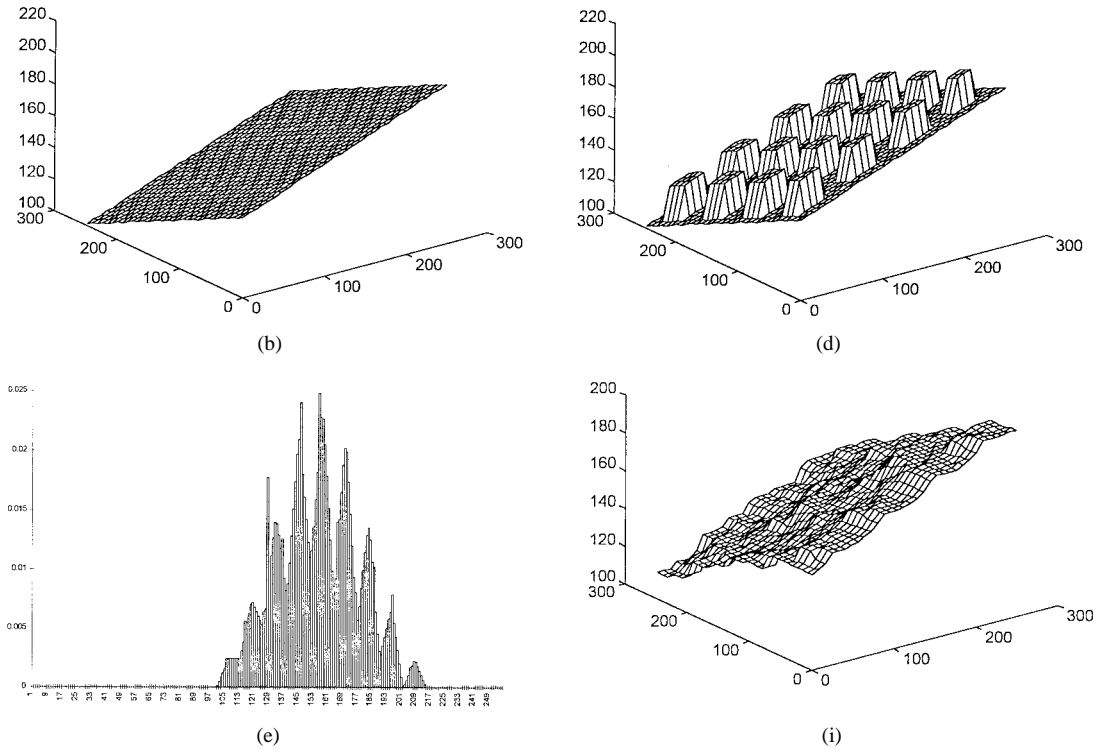


Fig. 3. (Continued.) (b) 3-D plot of (a). (d) 3-D plot of (c). (e) Gray-level histogram of (c). (i) 3-D plot of (h).

Here, $\alpha = 1/2\lambda$. Because $F(T)$ relates to c , $F(T)$ can be written as $F(x(T), y(T))$. By the chain rule

$$\frac{\delta F}{\delta T} = F_x \cdot \frac{\partial x}{\delta T} + F_y \cdot \frac{\partial y}{\delta T} = F_x \left/ \frac{\delta T}{dx} \right. + F_y \left/ \frac{\delta T}{dy} \right.$$

Note that here T is a function, δT is the variation of T , and $\delta T = dI - dT$ (see Fig. 2). So we have

$$\frac{\delta F}{\delta T} = \begin{cases} F_x/(I_x - T_x) + F_y/(I_y - T_y), & \text{if } (I_x \neq T_x) \text{ and } (I_y \neq T_y) \\ F_x/(I_x - T_x), & \text{if } (I_x \neq T_x) \text{ and } (I_y = T_y) \\ F_y/(I_y - T_y), & \text{if } (I_x = T_x) \text{ and } (I_y \neq T_y) \\ 0, & \text{if } (I_x = T_x) \text{ and } (I_y = T_y). \end{cases} \quad (6)$$

Now we have modeled the adaptive thresholding surface as the solution of a Poisson equation (5). The next step is to solve this equation. In next section, we will construct the numerical algorithm to tackle this problem.

III. IMPLEMENTATION

Similar to [14], which used successive overrelaxation (SOR) method [11] to solve Laplace equation for the interpolation, we use SOR to solve our Poisson equation (5). The SOR method here can be written as

$$T_{i,j}^{n+1} = T_{i,j}^n + \frac{\omega}{4} \xi_{i,j} \quad (7)$$

where $\xi_{i,j} = T_{i,j-1} + T_{i,j+1} + T_{i-1,j} + T_{i+1,j} - 4T_{i,j} - \alpha(\delta F \delta T)_{i,j}$, ω is the overrelaxation parameter and $1 < \omega < 2$. F_x, F_y, I_x, I_y, T_x , and T_y can be easily written in their finite-difference representations [11], which we will not repeat here. In order to keep track of the positions of c , we extend the two-value array $L(x, y)$ to a three-value array, in which boundary points of the segmented objects will be labeled as 2.

The first step in applying the iteration method to solve equations is to choose an initial solution. The choice is very important to the

optimization problem with a nonconvex objective function. Actually, our method tries to find the adaptive threshold surface from an initial surface which has to be given first. In most cases, this is not acceptable in thresholding algorithms; even if it can be accepted in some cases, to interactively draw a three-dimensional (3-D) surface on a computer is difficult. Fortunately, we find that the original image gray-level surface is a good start point for the threshold surface. At the first iteration, according to (2), all $L(x, y) = 0$, so $c = \phi$. The right-hand side of (5) is zero. So the first iteration behaves like smooth filtering which will let the threshold surface be smaller than the original image surface at some places. Then in the second iteration, $c \neq \phi$, and $\delta F/\delta T$ begins to take part in the deforming process of the threshold surface in the subsequent iterations.

If T changes, c is updated, and consequently $\delta F/\delta T$ changes. So the label array should be refreshed after each iteration. To arrive at a convergent solution, the number of iterations should be $O(J)$, if the image size is $J \times J$ [11]. We set a variable k to keep track of the number of points switched in/out of c between iterations. The iteration progress will stop if $k < K$ (a small integer number such as 10) or the iteration times exceeds a certain number (for example, J).

IV. EXPERIMENTS AND DISCUSSION

In this section, we will demonstrate some experimental results of our method. All these experiments are implemented on an SGI Indy PC workstation using C language. The SOR overrelaxation parameter we use is $\omega = 1.5$.

The first experiment is a simulated uneven background image. Fig. 3(a) is a sloping gray level background image generated by the computer. Fig. 3(b) is its 3-D plot. Sixteen objects are put on the background by adding 20 gray levels to their local positions [Fig. 3(c), (d)]. Fig. 3(e) shows Fig. 3(c)'s gray level histogram. It is hard to select a threshold directly from the histogram. The result of using Otsu's [8] fixed thresholding algorithm to Fig. 3(c)

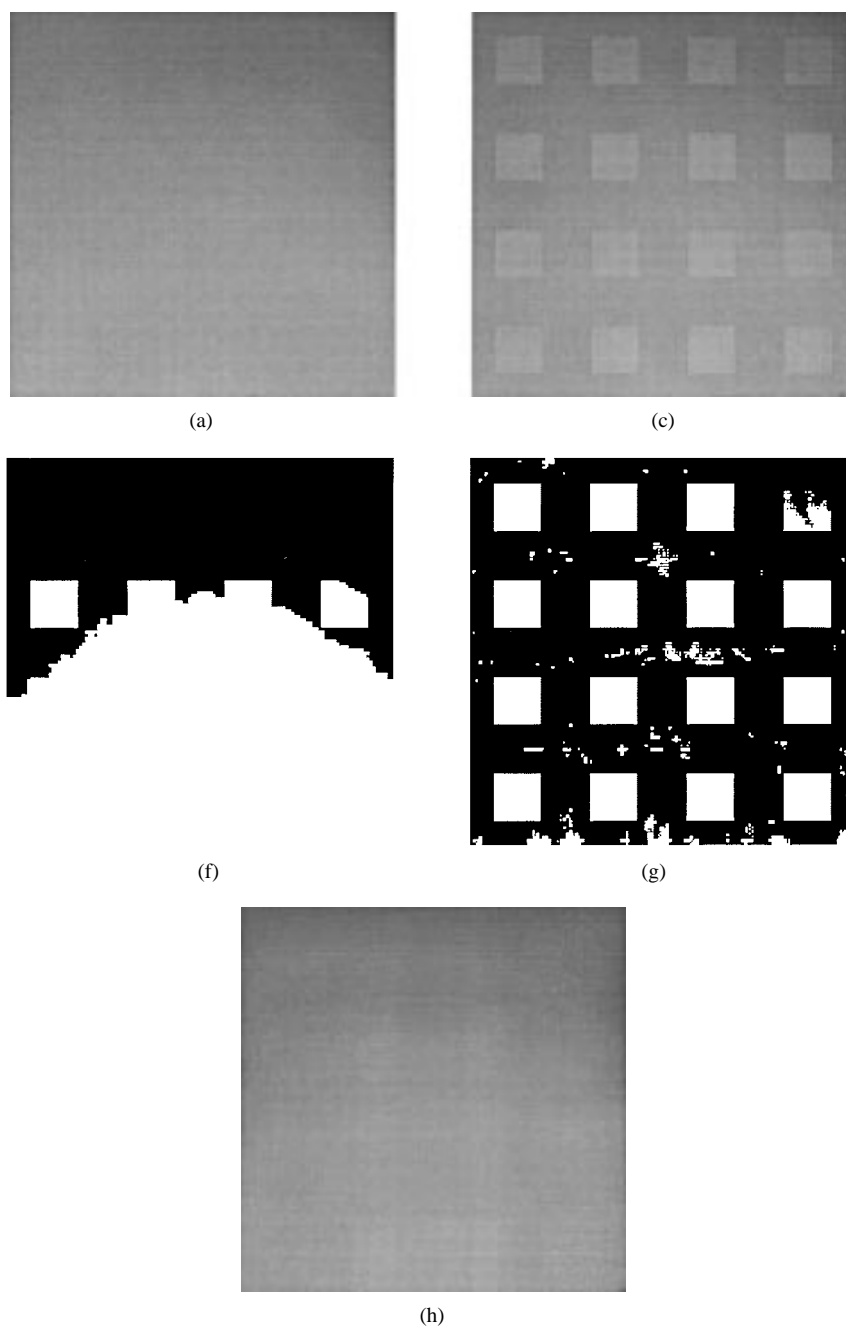


Fig. 4. (a) Real image background. (c) 16 objects being put on (a) by adding 20 gray levels to their local background. (f) Result of Otsu's algorithm to (c). (g) Result of our method to (c). (h) Adaptive thresholding surface found by our method.

is in Fig. 3(f). Fig. 3(g) is our method's result with $\alpha = 26$. The thresholding surface and its 3-D plot are shown in Fig. 3(h) and (i).

The next experiment (Fig. 4) is similar to the first one but the background is a real image of a nonuniformly illuminated white wall obtained from an Indycam of our workstation. The background has been processed by a 5×5 averaging filter to reduce noises. It can be seen the result of our method is good on the whole. In this experiment, $\alpha = 18$.

In Fig. 5, we use our method to segment a handwriting image in poorly illuminated environment. Also it is compared with fixed thresholding by Otsu's method.

The main difference between our method and [14] is that we embed the process of finding boundary points into the interpolation process so that the Laplace equation, when the right hand side of (5) is at

zero, used in [14] is changed to Poisson equation in our method. The right-hand side of (5) can be viewed as an external force to push the interactions of the threshold surface and the image surface to high image gray-level gradient places. The left hand side of (5) is an internal force which embodies suitable *a priori* knowledge of the threshold surface. We have found the results are more satisfactory when there are more objects in the image. Fig. 6 shows two images that have the same backgrounds as Figs. 3 and 4, respectively, but with fewer objects. There are more "unwanted objects" being segmented out. This is because in those areas with no objects nearby, no edge information is provided and the thresholding surface does not know where to move. On the other hand, we have little information about the background gray level surface. This is also the reason why in Fig. 5, the handwriting can be segmented well near the characters

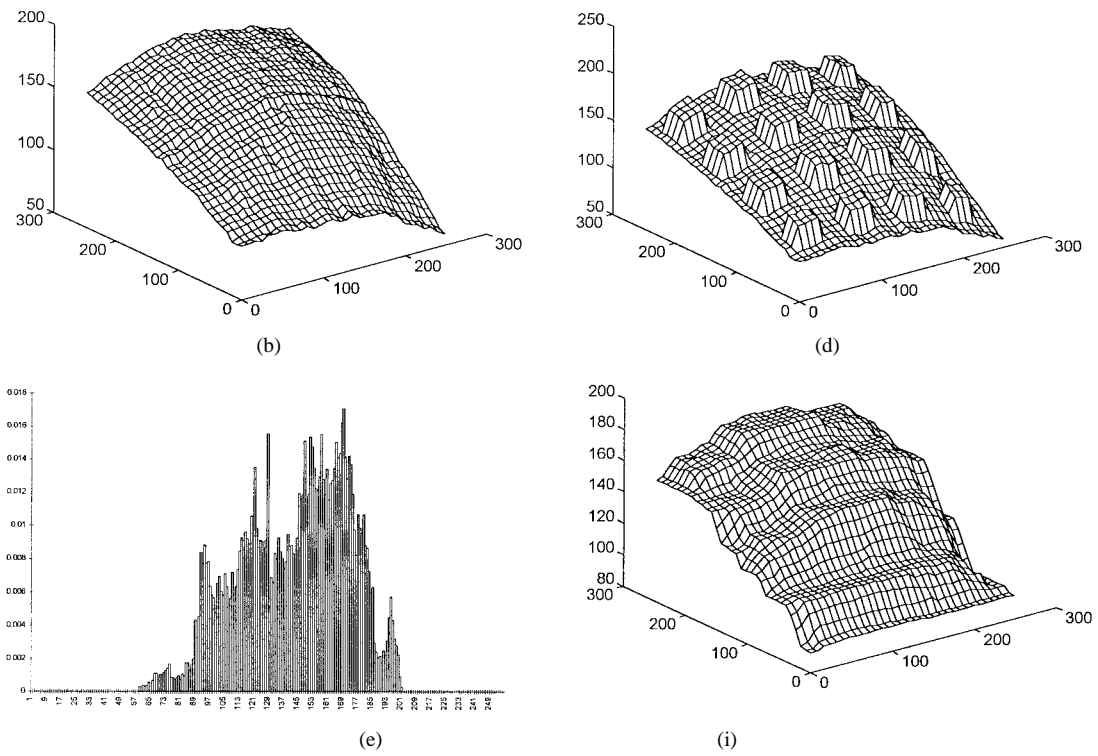


Fig. 4 (Continued.) (b) 3-D plot of (a). (d) 3-D plot of (c). (e) Gray-level histogram of (c). (i) 3-D plot of (h).

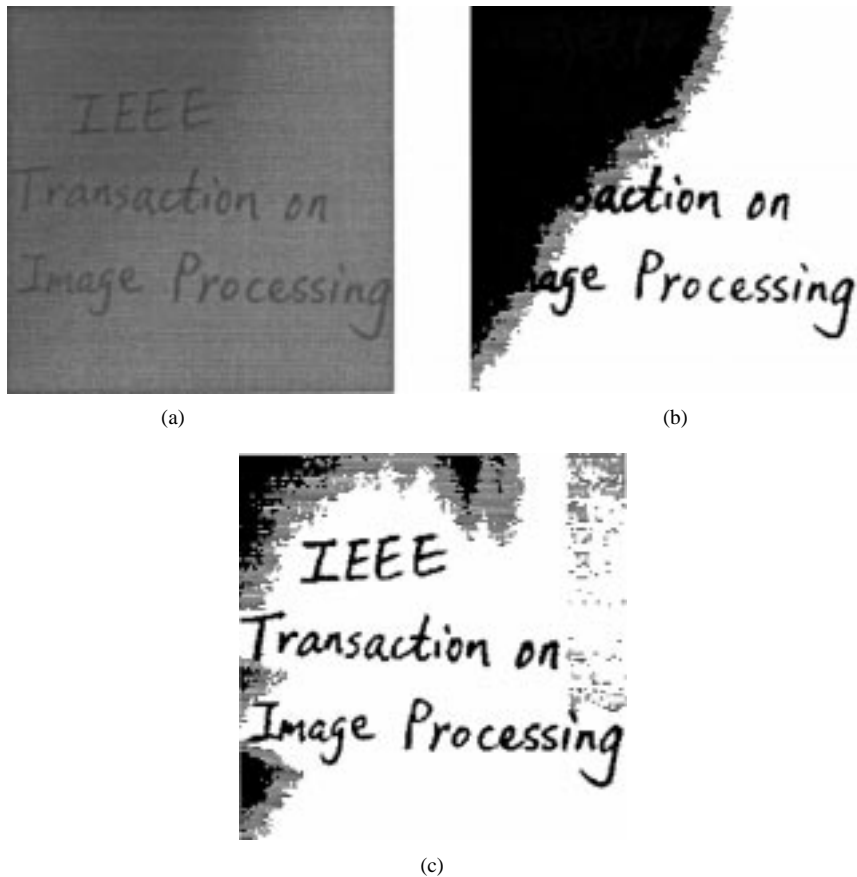


Fig. 5. (a) Poorly illuminated handwriting image. (b) Segmentation result by Otsu's method. (c) Segmentation result by our method.

but not in the surrounding background areas. These “ghost objects” were also be found in [14], and they can be eliminated by the step 7 of [14] using a validation process [5].

The size of all the images used above is 256×256 . The maximum number of iterations we use is 256. We find this is enough for convergence. It takes about 2 min for each image.

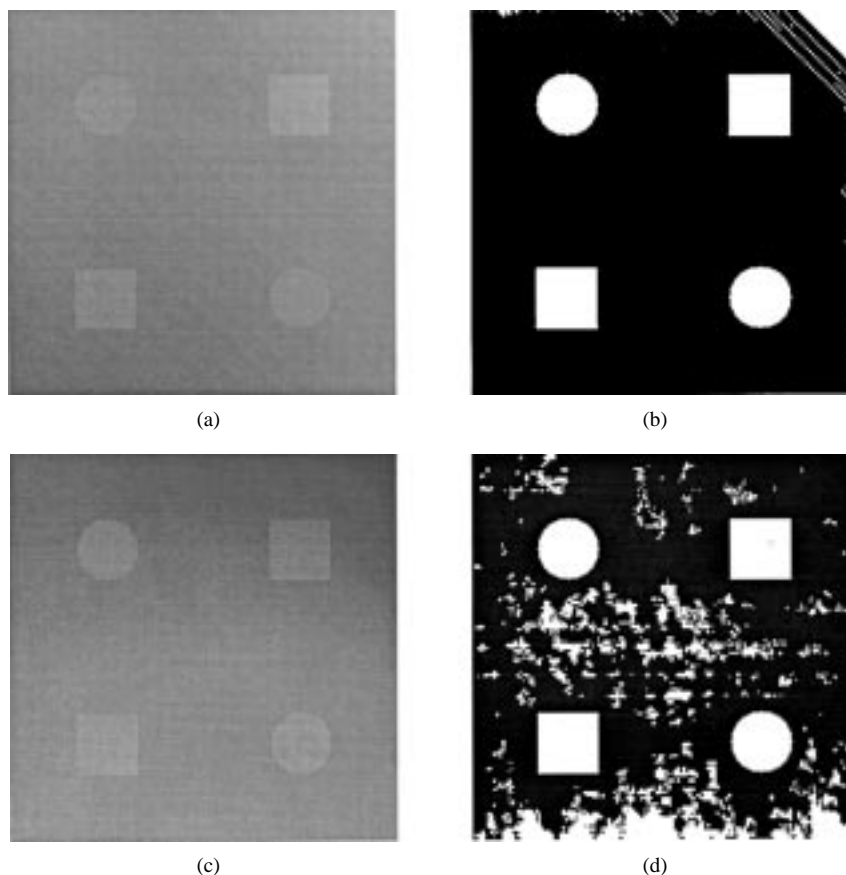


Fig. 6. (a) Same background but fewer objects image of the Fig. 3. (b) Segmentation result by our method to (a). (c) Same background but fewer objects image of the Fig. 4. (d) Segmentation result by our method to (c).

It should be pointed out that all variational methods in image segmentation, including ours, can be viewed as specific formulations from Mumford–Shah’s general model and its recent variants [6]. Besides image segmentation, this idea can be applied to other image processing methods, such as histogram modification [13].

V. CONCLUSION

In this work, we have proposed a new adaptive thresholding method for image segmentation. It can be viewed as a variational translation of [14]. The main advantage of our method is that the object boundary points selection which involves many parameter decisions and interpolation by solving a Laplace equation are integrated by solving a Poisson equation, in which only one coefficient α has to be set. Experimental results of simulated images and a real poorly illuminated handwriting image show the capability of this method. In our future work, the mathematical theory, the numerical implementation, and other application fields of our model will be further studied.

REFERENCES

- [1] C. K. Chow and T. Kaneko, “Automatic boundary detection of the left-ventricle from cineangiograms,” *Comput. Biomed. Res.*, vol. 5, pp. 388–410, 1972.
- [2] W. E. L. Grimson, “An implementation of a computational theory of visual surface interpolation,” *Comput. Vis., Graph., Image Process.*, vol. 22, pp. 39–69, 1983.
- [3] R. M. Haralick and L. G. Shapiro, “Image segmentation techniques,” *Comput. Vis., Graph., Image Process.*, vol. 29, pp. 100–132, 1985.
- [4] R. Jain, R. Kasturi, and B. G. Schunk, *Machine Vision*. New York: McGraw-Hill, 1995.
- [5] D. L. Milgram, A. Rosenfeld, T. Willet, and G. Tisdale, “Algorithms and hardware technology for image recognition,” *Final Report to U.S. Army Night Vision Laboratory*, Comput. Sci. Ctr., Univ. Maryland, College Park, 1978.
- [6] J.-M. Morel and S. Solimini, *Variational Methods in Image Segmentation*. Boston, MA: Birkhauser, 1995.
- [7] Y. Nakagawa and A. Rosenfeld, “Some experiments on variable thresholding,” *Pattern Recognit.*, vol. 11, pp. 191–204, 1979.
- [8] N. Otsu, “A threshold selection method from gray-level histogram,” *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-8, pp. 62–66, 1979.
- [9] N. R. Pal and S. K. Pal, “A review on image segmentation techniques,” *Pattern Recognit.*, vol. 26, pp. 1227–1249, 1993.
- [10] T. Poggio, V. Torre, and C. Koch, “Computational vision and regularization theory,” *Nature*, vol. 317, pp. 314–319, Sept. 1985.
- [11] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1992.
- [12] P. K. Sahoo, S. Soltani, A. K. C. Wong, and Y. C. Chen, “A survey of thresholding techniques,” *Comput. Vision, Graph., Image Process.*, vol. 41, pp. 233–260, 1988.
- [13] G. Sapiro and V. Caselles, “Histogram modification via partial differential equations,” *Proc. Int. Conf. Image Processing*, 1995, pp. 632–635.
- [14] S. D. Yanowitz and A. M. Bruckstein, “A new method for image segmentation,” *Comput. Vision, Graph., Image Process.*, vol. 46, pp. 82–95, 1989.