# A Fast 2D Otsu Thresholding Algorithm Based on Improved Histogram

**Conference Paper** · December 2009

DOI: 10.1109/CCPR.2009.5344078 · Source: IEEE Xplore



Gaobo Yang
Hunan University

# A Fast 2D Otsu Thresholding Algorithm Based on Improved Histogram

Ningbo Zhu[1], Gang Wang[1], Gaobo Yang[2,1], Weiming Dai[1]
1. School of computer and communication of Hunan University, Changsha, Hunan 410012, China
2. Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, Soochow, Jiangsu 215006, China
E-mail: wggg710@163.com

**Abstract:** Otsu adaptive thresholding is widely used in classic image segmentation. Two-dimensional Otsu thresholding algorithm is regarded as an effective improvement of the original Otsu method. To reduce the high computational complexity of 2D Otsu method, a fast algorithm is proposed based on improved histogram. Two-dimensional histogram is projected onto the diagonal, which forms 1D histgram with obvious peak and valley distribution. Then two-dimensional Otsu method is applied on a line that is vertical to the diagonal to find the optimal threshold. Furthermore, three look-up tables are utlilized to improve the computational speed by eliminating the redundant computation in original two-dimensional Otsu method. Theoretical analysis and experimental simulation show that the proposed approach greatly enhances the speed of thresholding and has better immunity to Salt and Pepper Noise.

**Key Words:** Image Segmentation, Thresholding，Otsu, Two-dimensional Histogram

## 1 INTRODUCTION

Image Segmentation is one of the basic techniques of image processing and computer vision. It is a key step for image analysis, comprehension and description. Among all the segmentation techniques, thresholding segmentation method [1] is the most popular algorithm and is widely used in image segmentation field. The basic idea of thresholding is to automatically select one or several optimal gray-level thresholds for separating objects of interest in an image from background based on their gray-level distribution. Basically thresholding methods can be categorized into two categories: global thresholding[1] and local thresholding[2]. Commonly used global thresholding methods are iterative method, Otsu method[3], maximum entropy method[4], and so on. In [1], the authors provided an exhaustive description and comparison of the performance measures over many image thresholding techniques.

Based on the gray-level histogram of an image, Otsu's thresholding method takes the variance between clusters as the criterion to select the optimal threshold. This method has been cited as an effective thresholding technique and widely used in real thresholding tasks. However, there are always uncertain disturbing factors in practical applications. These disturbing factors probably make one-dimensional (1D) histogram fail to possess obvious vally between two peaks. As a result, Otsu method can not obtain optimal segmentation results. To solve this problem, Liu et. al.[5] extended 1D histogram into two-dimensional(2D) histogram, which utilizes not only gray value distribution of an image, but also the dependency of pixel in its neighborhood. This method greatly improved the segmentation results, especially in low SNR condition. But the computation cost of 2D histogram is amazing. Many researches have been done to reduce the computation cost[6]-[11]. Hao et.al.[9] proposed a fast 2D Otsu algorithm, which changed the two dimensional threshold vector into one dimensional threshold and reduced much more computation cost. Huang et.al.[10] projected 2D histogram onto diagonal, which composing a new 1D histogram. This method reduced the searching space but could not get the optimal segmentation result with noisy image. Lang et.al.[11] utilized integral image to speed up the computation of 2D Otsu, but the method still had to search the entire 2D space.

In this paper, an improved fast algorithm is proposed to overcome the shortcomings of 1D Otsu and 2D Otsu methods. 2D histogram is projected onto diagonal. The searching space is simplified from 2D to 1D. Moreover, the image segmentation speed is improved by utilizing three look-up tables, which can greatly eliminate the redundant computations.

The rest of this paper is organized as follows. Section 2 gives a brief review of 2D Otsu algorithm. In section 3, we describe the proposed fast algorithm in detail. Then, section 4 presents theoretical analysis and experiment results. Finally, the paper is summarized and conclusions are drawn in section 5.

## 2 TWO-DIMENSIONAL OTSU METHOD[5]

Given an image f(x, y) represented by L gray levels, then its neighborhood average image g(x, y) can also be represented by L gray levels

$$g(x,y)=\frac{1}{2k+1}\sum_{\Delta x=-k}^{k}\sum_{\Delta y=-k}^{k}f(x+\Delta x,y+\Delta y) \qquad (1)$$

In this paper, k=1.

Then, for each pixel in the original image, we can obtain a pair $(i, j)$ , which is composed by original intensity i and average intensity j. Let $f_{ij}$ denotes the frequency of pair $(i, j)$ appeared in the image, let M denotes image size, and its joint probability can be expressed as

$$P_{ij} = \frac{f_{ij}}{M} \tag{2}$$

Where, $\sum_{i=0}^{L-1}\sum_{j=0}^{L-1} f_{ij} = M$ , $\sum_{i=0}^{L-1}\sum_{j=0}^{L-1} P_{ij} = 1$

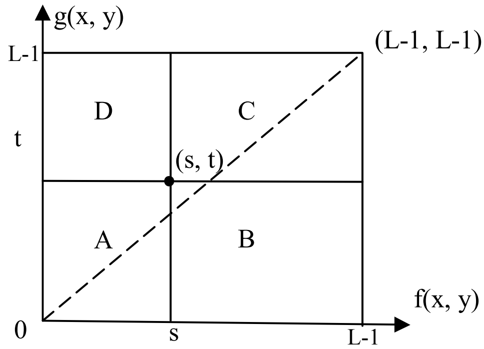



Fig. 1: two-dimensional histogram

The 2D histogram of the image is $\{ p_{ij} \}$. Fig.1 shows the top view of 2D histogram. It covers a square region with size $L \times L$ . The x-coordinate (i) represents gray level and the y-coordinate (j) represents the local average gray level. Given an arbitrary threshold pair $(s, t)$ , the 2D histogram can be divided into four regions. Regions A and C represent object and background respectively, and regions B and D represent edge and noise respectively. Let two clusters $C_0$ and $C_1$ represent object and background, then the probabilities of two clusters can be denoted as

$$\omega_0 = \sum_{i=0}^{s-1}\sum_{j=0}^{t-1} P_{ij} \qquad \omega_1 = \sum_{i=s}^{L-1}\sum_{j=t}^{L-1} P_{ij} \tag{3}$$

The intensity means value vectors of two clusters can be expressed as follows

$$\mu_0 = (\mu_{00}, \mu_{01})^T = \left( \sum_{i=0}^{s-1}\sum_{j=0}^{t-1} iP_{ij} / \omega_0, \sum_{i=0}^{s-1}\sum_{j=0}^{t-1} jP_{ij} / \omega_0 \right)^T \tag{4}$$

$$\mu_1 = (\mu_{11}, \mu_{10})^T = \left( \sum_{i=s}^{L-1}\sum_{j=t}^{L-1} iP_{ij} / \omega_1, \sum_{i=s}^{L-1}\sum_{j=t}^{L-1} jP_{ij} / \omega_1 \right)^T \tag{5}$$

The total mean vector of 2D histogram is

$$\mu_t = (\mu_{ti}, \mu_{tj})^T = \left( \sum_{i=0}^{L-1}\sum_{j=0}^{L-1} iP_{ij}, \sum_{i=0}^{L-1}\sum_{j=0}^{L-1} jP_{ij} \right)^T \tag{6}$$

In most cases, the probability that away from the diagonal can be negligible. So, it's easy to be verified that

$$\omega_0 + \omega_1 \approx 1 \qquad \mu_t \approx \omega_0\mu_0 + \omega_1\mu_1 \tag{7}$$

The between-class discrete matrix is defined as

$$S_b = \sum_{k=0}^{1} \omega_k \left[ (\mu_k - \mu_t)(\mu_k - \mu_t)^T \right] \tag{8}$$

The trance of discrete matrix could be expressed as

$$r_{trance}(S_b) = \omega_0 \left[ (\mu_{0i} - \mu_{ti})^2 + (\mu_{0i} - \mu_{tj})^2 \right] + \omega_1 \left[ (\mu_{1i} - \mu_{ti})^2 + (\mu_{1i} - \mu_{tj})^2 \right] \tag{9}$$

The optimal threshold of 2D Otsu method is the threshold that maximize $r_{trance}(S_b)$ .

## 3 IMPROVED FAST 2D OTSU ALGORITHM

### 3.1 Improved 2D histogram

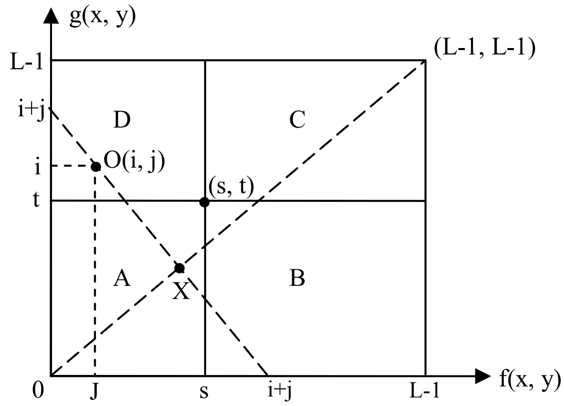



Fig. 2: projecting 2D histogram onto diagonal

Let L be the gray levels of image $f(x, y)$ , then the gray levels of the neighborhood average image $g(x, y)$ can also be L. The length of the diagonal of 2D histogram would be $\sqrt{2}(L-1)$ (as shown in Fig. 2). Scatter the diagonal

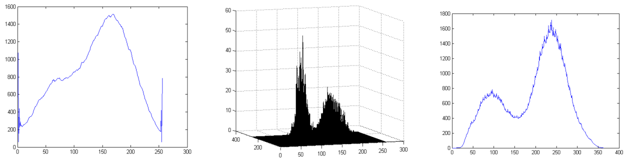
$$M = \left[ \sqrt{2}(L-1) \right] \qquad x = \left[ \frac{\sqrt{2}}{2}(i+j) \right] \tag{10}$$

Where M is length of the discrete diagonal, and x is discrete value for diagonal. $x \in M(0,1,2,\ldots,M-1)$ , i is gray level of original image and j is gray level of its neighborhood average image. $i \in L(0,1,2,\ldots,L-1)$ , $j \in L(0,1,2,\ldots,L-1)$ , $[\ \bullet\ ]$ is round function.

$h(i, j)$ is the statistic number of pixels at point O $(i, j)$ in 2D histogram. When we project the point O $(i, j)$ to the point X on diagonal, $h(i, j)$ is projected onto the diagonal too. As shown in follows, $H(x)$ is the projected value on diagonal.

$$H(x) = \sum_{x=[\sqrt{2}(i+j)/2]} h(i, j) \tag{11}$$

Therefore, the projected values on diagonal can be seen as a histogram with M levels. Fig. 3 shows the histogram results of which we added Gaussian noise to an image. (a) is 1D histogram of noisy image, it's evident that there is no marked valley in the histogram, so the Otsu's method can not perform well. (b) is 2D histogram of noisy image, and the projected result on diagonal is shown in (c). As shown in (c), we get an 1D histogram on diagonal, and the histogram is peak marked.



(a) (b) (c)

Fig. 3: (a) 1D histogram of noisy image (b) 2D histogram (c) Projected diagonal histogram

We utilize Otsu thresholding method on the 1D histogram that projected on diagonal to get a segmentation point K at the optimal threshold (as shown in Fig. 4). Then draw a line p passing the point K, and the line p is vertical to the diagonal. We can easily find that the object and background are in different side of line p. That's because K is the trough point of the projected histogram. So the optimal threshold should be on line p. The equation of the line p is defined as follows.

$$g(x,y)=-f(x,y)+2i \quad i\in L(0,1,2,..,L-1) \tag{12}$$

The line p intersect on the x axes and the y axes at points $(2i,0)$ and $(0,2i)$ respectively. When searching the optimal threshold on line p, we loop t from 0 to 2i, and by this, $s=-t+2i$ . Therefore, the threshold searching in two dimensions is transformed into one dimensional, saving lots of time.

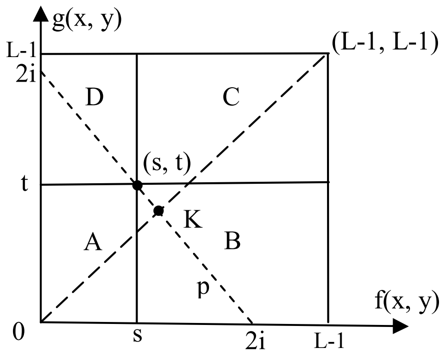



Fig. 4: Search the optimal threshold in improved histogram

### 3.2 Fast computation method

From expression (3)-(6), we find that the original 2D Otsu method contains a large amount of calculation of sum. And for each pair $(s,t)$ , all computation is recalculated, which waste a great deal of time. To reduce the redundant computation, we yield three look-up tables. For each pair $(s,t)$ , when calculating the trace of discrete matrix, we can simply use look-up table to find the corresponding sum results.

$$\begin{aligned}\mu_0=(\mu_{00},\mu_{01})^T &= \left(\sum_{i=0}^{s-1}\sum_{j=0}^{t-1} iP_{ij}/\omega_0, \sum_{i=0}^{s-1}\sum_{j=0}^{t-1} jP_{ij}/\omega_0\right)^T \\ &= \left(\sum_{i=0}^{s-1}\sum_{j=0}^{t-1} iP_{ij}, \sum_{i=0}^{s-1}\sum_{j=0}^{t-1} jP_{ij}\right)^T/\omega_0\end{aligned} \tag{13}$$

$$\begin{aligned}\mu_1=(\mu_{10},\mu_{11})^T &= \left(\sum_{i=s}^{L-1}\sum_{j=t}^{L-1} iP_{ij}/\omega_1, \sum_{i=s}^{L-1}\sum_{j=t}^{L-1} jP_{ij}/\omega_1\right)^T \\ &= \left(\sum_{i=s}^{L-1}\sum_{j=t}^{L-1} iP_{ij}, \sum_{i=s}^{L-1}\sum_{j=t}^{L-1} jP_{ij}\right)^T/\omega_1\end{aligned} \tag{14}$$

Let $X_0=\sum_{i=0}^{s-1}\sum_{j=0}^{t-1} iP_{ij}$ , $Y_0=\sum_{i=0}^{s-1}\sum_{j=0}^{t-1} jP_{ij}$

$$X_1=\sum_{i=s}^{L-1}\sum_{j=t}^{L-1} iP_{ij}\,,\quad Y_1=\sum_{i=s}^{L-1}\sum_{j=t}^{L-1} jP_{ij}$$

Therefore

$$\mu_0=(\mu_{00},\mu_{01})^T=(X_0,Y_0)^T/\omega_0 \tag{15}$$

$$\mu_1=(\mu_{10},\mu_{11})^T=(X_1,Y_1)^T/\omega_1 \tag{16}$$

We traverse the 2D histogram once to yield three look-up tables as follows

$$P(m,n)=\sum_{i=0}^{m}\sum_{j=0}^{n} f_{ij} \tag{17}$$

$$X(m,n)=\sum_{i=0}^{m}\sum_{j=0}^{n} if_{ij} \tag{18}$$

$$Y(m,n)=\sum_{i=0}^{m}\sum_{j=0}^{n} jf_{ij} \tag{19}$$

So we have

$$X_0=X(s,t)\,,\quad Y_0=Y(s,t) \tag{20}$$

$$X_1=X(L-1,L-1)-X(L-1,t)-X(s,L-1)+X(s,t) \tag{21}$$

$$Y_1=Y(L-1,L-1)-Y(L-1,t)-Y(s,L-1)+Y(s,t) \tag{22}$$

$$\omega_0=P(s,t) \tag{23}$$

$$\omega_1=P(L-1,L-1)-P(L-1,t)-P(s,L-1)+P(s,t) \tag{24}$$

From (20)-(24), probabilities and mean value vectors can be calculated using look-up table, and then the trace of discrete matrix is obtained easily. For every threshold pair $(s,t)$ , the one which maximize the trace of discrete matrix is regarded as the optimal threshold vector.

## 4 ALGORITHM COMPLEXITY ANALYSIS AND EXPERIMENT RESULTS

### 4.1 Algorithm complexity analysis

Supposing L is the grey levels of an image. In 2D Otsu algorithm, we have to calculate the trace of discrete matrix once for every pair $(s,t)$ , and get threshold when the trace is the largest. Furthermore, we have to do double cycle for s and t respectively. As a result, $L^2$ loops are needed. The algorithm proposed in this paper only need to search the threshold on two lines instead of the whole area. In the worst case, it only needs $2\sqrt{2}L$ loops.

In 2D Otsu algorithm, when we calculate the trace of discrete matrix, adding computing has to be done for $st+(L-s)(L-t)$ points each time, so the total adding times would be

$$A_1 = \sum_{s=0}^{L-1}\sum_{t=0}^{L-1}\left[s\cdot t+(L-s)(L-t)\right]=O(L^4) \tag{25}$$

The time complexity of the fast algorithm proposed in [9] is

$$A_2 = \sum_{s+t=0}^{2L}\left[N^2+2N(L-N)\right]=O(L^3) \tag{26}$$

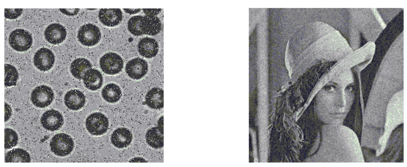
In the fast algorithm of this paper, we only need to add 9 times while calculating the trace of discrete matrix, and only $\sqrt{2}$ L in the worst case. When projecting the histogram and yielding the look-up table, we need $(L-1)\times(L-1)$ times of adding computing. The time complexity of Otsu on diagonal is $O(L^2)$. So, the total time complexity would be

$$A_3 = \sqrt{2}L\times 9+(L-1)\times(L-1)+O(L^2)=O(L^2) \tag{27}$$

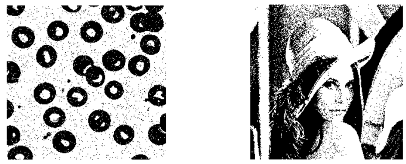
From the time complexity analysis above, we can see that the algorithm proposed in this paper enhanced the computing and searching speed greatly.

### 4.2 Experiment results

To evaluate the practical performance of the proposed method, we add Salt and Pepper Noise in the image of blood, and add Gaussian noise in the image of lena. We segment these images using 1D Otsu algorithm, 2D Otsu algorithm, algorithm in [9], and the algorithm proposed respectively, and compare there results, shown in Fig. 5. From the Fig. 5, we can see that 1D Otsu algorithm is very sensitive to noise, and don't have the effect of anti-noise neither to Salt and Pepper Noise nor to Gaussian noise. 2D Otsu algorithm and the algorithm in this paper perform similarly with Gaussian noise. But we can see difference in segmentation image between the method in [9] and the 2D Otsu method. That's because in [9], they changed the threshold pair $(s,t)$ into a number $(s+t)$. In the results of Salt and Pepper Noisy images, we can see that our method performs better than 2D Otsu. Furthermore, out method consumes the least time.



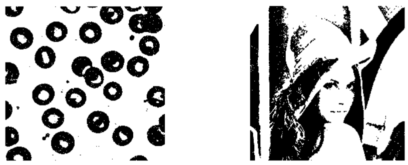
(Noisy images)



(Otsu)

(2D Otsu)

(Method in [9], N=50)



(Proposed)

Fig. 5: Segmentation results of noisy image

Table 1 shows the optimal threshold and time-consuming of different algorithm in segmenting noisy images of blood and lena. From the table we can see that the time-consuming of the algorithm proposed in this paper is not only better than 2D Otsu method, but also better than the fast algorithm in [9] and the algorithm in [11]. And the optimal threshold obtained by the algorithm proposed is nearly the same to the original 2D Otsu method.

Table 1: The optimal threshold and Time-consuming of different algorithm

| | | blood | | lena | |
|---|---|---|---|---|---|
| | | original | Noisy (Salt and Pepper) | original | Noisy (Gaussian) |
| 1d Otsu | Threshold | 112 | 111 | 117 | 119 |
| | Time /s | 0.012 | 0.012 | 0.012 | 0.012 |
| 2d Otsu | Threshold | (113,117) | (112,211) | (119,125) | (113,125) |
| | Time /s | 45.067 | 45.238 | 45.214 | 46.072 |
| Method in[9] | Threshold | 223 | 294 | 237 | 223 |
| | Time /s | 0.527 | 0.531 | 0.527 | 0.528 |
| Method in[11] | Threshold | (113,117) | (112,211) | (119,125) | (113,125) |
| | Time /s | 0.034 | 0.034 | 0.034 | 0.034 |
| proposed | Threshold | (114,115) | (111,118) | (118,124) | (114,123) |
| | Time /s | 0.025 | 0.025 | 0.025 | 0.025 |

## 5 CONCLUSIONS

In this paper, we proposed an improved image segmentation method to overcome the shortcomings of 1D Otsu and 2D Otsu algorithms. This method utilized the space characteristics and projector features of 2D histogram to reduce the threshold searching time. Furthermore, we eliminated the redundant computation in the 2D Otsu method. Theoretical analysis and experimental simulation show that the method greatly enhanced the speed of thresholding and has better noise immunity for Salt and Pepper Noise.

## ACKNOWLEDGEMENTS

This paper is supported by the open fund of Provincial Key Laboratory for Computer Information Processing Technology, Soochow University.

## REFERENCES

[1] M.Sezgin, B.Sankur. Survey over image thresholding techniques and quantitative performance evaluation. Journal of Electronic Imaging. 13(1): 146-165, 2004.

[2] J.Sauvola, M. Pietikainen. Adaptive Document Image Binarization. Pattern Recognition. 33: 225-236, 2000.

[3] Otsu N. A threshold selection method from gray-level histogram. IEEE Transactions on System Man and Cybemetic. 9(1): 62-66, 1979.

[4] PUN T. A new method for gray-level picture thresholding using the entropy of histogram. Signal Processing. 2(3): 233-237, 1980.

[5] J. Z. Liu, W. Q. Li. The automatic thresholding of gray-level picture via 2D Otsu method. Acta Automatica Sinica. 19(1): 101-105, 1993.

[6] J.Gong, L.J.Li, W.N.Chen. Fast recursive orithm for two-dimensional thresholding. Pattern Recognition. 31(3): 295-300, 1998.

[7] J.L.Fan,F.Zhao. Two-Dimensional Otsu's Curve Thresholding Segmentation Method for Gray-Level Images. Acta Electronica sinica. 35(4): 751-755, 2007.

[8] Y.Q.Wu, Z.Pan, W.Y.Wu. Image thresholding based on two-dimensional histogram oblique segmentation and its fast recurring algorithm. Journal on Communication 29(4): 77-89, 2008.

[9] Y. M. Hao, F. Zhu. Fast algorithm for twodimensional Otsu adaptive threshold algorithm. Journal of Image and Graphics. 10(4): 484-488, 2005.

[10] L.J.DONG, G.Yu. An Approach to Fast Image Segmentation Based on 2-D Histogram. Jouma1 of Northeastern University. 26(3): 220-223, 2005.

[11] X.P.Lang, F.Zhu, Y.M.Hao, J.J.Ou. Integral Image Based Fast Algorithm for Two-dimensional Otsu Thresholding. Proceedings of the 2008 Congress on Image and Signal Processing. 3(3): 667-681, 2008.