

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/309543000>

A Robust 2D Otsu's Thresholding Method in Image Segmentation

Article in *Journal of Visual Communication and Image Representation* · October 2016

DOI: 10.1016/j.jvcir.2016.10.013

CITATIONS

15

READS

368

3 authors, including:



Jian Hou

Bohai University

82 PUBLICATIONS 483 CITATIONS

SEE PROFILE

Elsevier Editorial System(tm) for Journal of
Visual Communication and Image Representation
Manuscript Draft

Manuscript Number: JVCI-15-417R1

Title: A Robust 2D Otsu's Thresholding Method in Image Segmentation

Article Type: Regular Article

Keywords: Otsu's method
thresholding
image segmentation

Corresponding Author: Prof. Jian Hou, Ph.D.

Corresponding Author's Institution: Bohai University

First Author: Chunshi Sha

Order of Authors: Chunshi Sha; Jian Hou; Hongxia Cui

Abstract: Otsu's method is a classic thresholding approach in image segmentation. While the two-dimensional (2D) Otsu's method performs better than the original one in segmenting images corrupted by noise, it is sensitive to Salt\&Pepper noise. In order to solve this problem, we present a robust 2D Otsu's thresholding method in this paper. Our method builds the 2D histogram based on the image smoothed by both median and average filters, in contrast to the traditional method using averaged image only. Then the optimal threshold vector is determined with two one-dimensional searches on the two dimensions of the 2D histogram. In addition, we introduce a region post-processing step to deal with the pixels of noise and edges. Compared with the traditional 2D Otsu's method, our method improves the robustness to Salt\&Pepper noise and Gaussian noise significantly. Experimental results on both synthetic and real images validate the effectiveness of the proposed MAOTSU_2D method.

Dear Editor,

Please find enclosed the revised version of our paper entitled "A Novel Robust 2D Otsu's Thresholding Method in Image Segmentation", authored by Chunshi Sha, Hongxia Cui and myself, submitted to be considered for publication in Journal of Visual Communication and Image Representation.

In this revised version, we made significant revisions according the reviewer's comments, including the organization, the expression and the language.

Should you have any questions as to the submission, please do not hesitate to contact us.

Best Regards,

Jian Hou

Response to the Reviews

We would like to thank the reviewers for their constructive and helpful comments, based on which we have carefully revised the manuscript.

Summary of the revisions:

1. According to the reviewer's suggestions, we carefully reformatted the paper.
2. We revised the paper to respond to the reviewer's confusion as to the derivation from Eq. (17) to Eq. (33).
3. We carefully revised the English of this paper and asked for the help of a native speaker in polishing the paper.

Response to reviewer #1:

1 The author is suggested to reformat the paper. For example, moving the equation(3)-(13) to section IV(b).

According to the reviewer's suggestion, we reformatted the paper and moved the Eq. (3)-Eq. (13) to section IV(b).

2. The author is suggested to revise the paper by Native speaker.

According to the reviewer's suggestion, we have carefully revised the paper to improve the English, and asked for the help of a native speaker in polishing the paper.

3 Typical, in image segmentation the two components are called as foreground vs background.

We are very sorry for the negligence in the original version. We have revised these two components as foreground and background in this revised version.

4 I was confused by the derivation from Eq(17)-Eq(33), what are the points that the author tried to explain? In particularly, what is the connection between the between-class variance with Salt&Pepper noises.

The derivation from Eq. (17)-Eq. (33) is intended to provide the ground of selecting the optimal thresholds on two dimensions separately. If the condition in Eq. (18) is satisfied, the optimal threshold vector obtained with Eq. (17) is the same as the one obtained with Eq. (16). In other words, the optimal threshold vector obtained by two 1D histogram searches is the same as the one obtained by traditional 2D search on the condition that the probability of the pixels of noise and edge is negligible.

We cannot find the connection between the between-class-variance with Salt&Pepper noises directly from the derivation Eq. (17)-Eq. (33). However, we find the following two disadvantages of the traditional 2D searches which lead to its sensitive to noises, especially Salt&Pepper noise. First, we note that the between-class variance in the traditional 2D search with Eq. (37) or Eq. (14)

is calculated between foreground and the combination of background, noise and edge, but not between foreground and background from Eq. (34)-Eq. (37). In this case, if the probability of the pixels of noise and edge is not negligible, the condition in Eq. (13) is not satisfied and the between-class variance represented by Eq. (37) or Eq. (14) is no longer applicable. Furthermore, Eq. (37) or Eq. (14) can not guarantee an appropriate division of region 2, 3 and 4, and a great number of pixels may be wrongly assigned to regions 2 and 3 if the two optimal thresholds differ significantly from each other. This effect is especially obvious in images corrupted by Salt&Pepper noise.

In contrast, our method using two 1D searches separately is less likely to be influenced by Salt&Pepper noise. Since the two 1D searches are conducted on the gray level histogram and the median-average histogram separately, either threshold is unlikely to be affected by noise significantly. As a result, the two thresholds tend to be quite close to each other, and only a small fraction of pixels appear in regions 2 and 3.

These explanation appears in Section 4.2, pp. 21-24.

Highlights

- We proposed to use median filtering followed by average filtering in building the 2D histogram.
- We proposed to search for the optimal thresholds on two dimensions separately of the 2D histogram.
- We presented a region post-processing step to deal with pixels of noise and edges.
- Our proposed method is shown to perform the best or the second best in comparison with the other 11 methods.

1
2
3
4
5 **1. Introduction**
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30

Image segmentation is an important step to extract foreground of interest for further analysis in image processing, pattern recognition and artificial intelligence, etc. While in general one image can be segmented into multiple parts (Hou et al., 2014), in this paper we focus on the case with only two parts, i.e., foreground and background. In existing image segmentation methods of this field, thresholding is one of the most frequently used, partly due to its simpleness and effectiveness (Sezgin and Sankur, 2004). Several classic thresholding methods include Otsu's method (Otsu, 1979), minimum error method (Kittler and Illingworth, 1986) and maximum entropy method (Pun, 1980). In this paper we focus on Ostu's method and aim to improve its robustness.

Otsu's method was proposed by Nobuyuki Otsu in 1979 and is widely used in document binarization (Moghaddam and Cheriet, 2012), computer vision (Xu et al., 2011) and pattern recognition (Sirisha et al., 2013). Ostu's method selects the optimal threshold automatically by maximizing the between-class variance of the segmented image. It is simple, efficient and parameter free. Due to these advantages, a large number of methods have been proposed to improve the original Otsu's method, including (Cheriet et al., 1998; Xue and Titterington, 2011; Alsaeed et al., 2012; Cai et al., 2014; Lai and Pl., 2014). However, these methods are usually unable to generate satisfactory segmentation results in the case that images are corrupted by noise. In order

1
2
3
4
5 to solve this problem, (Liu et al., 1991) presented a 2D Otsu's method to
6 extend Otsu's method with a 2D histogram. The 2D Ostu's method utilizes
7 the gray level of each pixel as well as the average gray level of pixels in its
8 neighborhood, and performs better than Otsu's method in case of low signal-
9 to-noise ratio (SNR). While effective, this method uses an exhaustive search
10 on 2D histogram to find the optimal threshold vector and is computationally
11 expensive.
12
13

14 Many works have been published to reduce the computation load of 2D
15 Otsu's method. In (Gong et al., 1998) a fast recursive version of 2D Otsu's
16 method was shown to reduce the computation complexity significantly. Then
17 in (Hao and Zhu, 2005) the authors changed the 2D threshold vector into
18 an 1D threshold and improve the computation efficiency further. Yue *et*
19 *al.* (Yue et al., 2009) proposed a decomposition based 2D Otsu's method
20 which calculates the optimal thresholds by two 1D searches separately, in
21 contrast to the traditional method selecting the optimal threshold vector in
22 2D space. This method performs better than or comparable to the traditional
23 methods in segmentation accuracy, while being more computationally and
24 memory efficient than the latter. Some other works in this field include
25 (Wei et al., 2007; Wang et al., 2007; Lang et al., 2008). In order to improve
26 segmentation quality, some post-processing and modification methods for
27 2D Otsu's method have also been proposed, including curve thresholding
28 segmentation (Fan and Zhao, 2007), gray level-gradient histogram based 2D
29 Otsu's algorithm (Chen et al., 2010), 2D Otsu's thresholding algorithm based
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

on local grid box filter (Guo et al., 2014) and others (Wang et al., 2011; Chen et al., 2012).

In addition to 2D Otsu's method, Jing *et al.* (Jing et al., 2003) proposed a 3D Otsu's method which introduces the median of the neighborhood of each pixel as the third characteristic and constructs a 3D histogram. This method produces better results than 2D Otsu's methods at the cost of larger computation load. Although some methods, e.g., (Fan et al., 2007; Wang et al., 2008, 2010; Sthitpattanapongsa and Srinark, 2012), have been proposed to reduce the computation complexity of 3D Otsu's method, this kind of methods are still rather inefficient as the optimal threshold vector is searched in 3D space.

As we reviewed above, there are many works devoted to improving the performance of 2D Otsu's method. However, the traditional 2D Otsu's methods are still afflicted by two major problems. First, they are sensitive to Salt&Pepper noise and are usually unable to generate satisfactory segmentation results in the case that images are corrupted by Salt&Pepper noise. Second, existing 2D Otsu's methods usually simply label all the noise and edges as in background or in foreground. This careless practice often leads to disastrous segmentation results in case of low SNR.

In order to overcome the above mentioned problems, we present a robust 2D Otsu's thresholding method, called Median-Average Otsu's method (MAOTSU_2D) in this paper. Our method builds the 2D histogram using the image smoothed by both median and average filtering, in contrast to tra-

ditional methods where images are smoothed by average filtering only. Then we determine the optimal threshold vector by two 1D searches on the two dimensions of the 2D histogram separately. In addition, we present a region post-processing method to deal with the pixels labeled as noise and edges. In experiments on both synthetic or real images our method is shown to be the most or second most robust one with respect to noise in twelve methods, while being computationally efficient than the fast recursive algorithm proposed in (Gong et al., 1998).

The remainder of this paper is organized as follows. In Section 2 we review the traditional 2D Otsu's method briefly. We then analyze the problems of the traditional 2D Otsu's method in Section 3 and present our MAOTSU_2D method in section 4. The experimental validation of our method is reported in Section 5. Finally, Section 6 concludes this paper.

2. Traditional 2D Otsu's Method

In this section we introduce the traditional 2D Otsu's method briefly. An image $f(x, y)$ of the size $M \times N$ is represented in L gray levels. Its corresponding averaged image $g(x, y)$ is defined by

$$g(x, y) = \frac{1}{k^2} \sum_{a=-(k-1)/2}^{(k-1)/2} \sum_{b=-(k-1)/2}^{(k-1)/2} f(x+a, y+b). \quad (1)$$

where k is set to be 3 in this paper. Denoting the gray level at pixel (x, y) in image $f(x, y)$ and $g(x, y)$ as i and j respectively, we obtain a gray level pair (i, j) for each pixel. Let F_{ij} be the frequency of the pair (i, j) , its joint

probability is given by

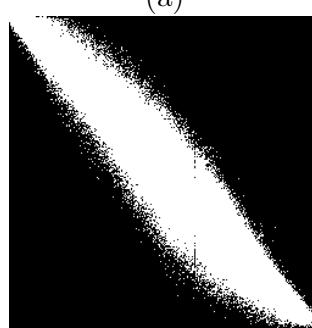
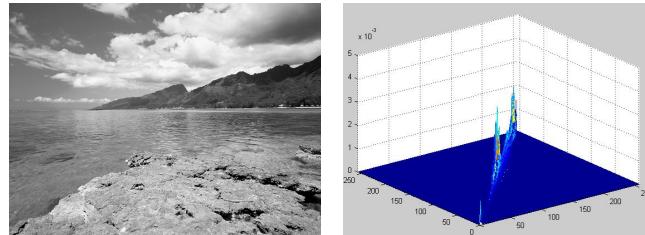
$$p_{ij} = \frac{F_{ij}}{M \times N}, \quad (2)$$

where $i, j = 0, 1, \dots, L - 1$ and $\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} p_{ij} = 1$.

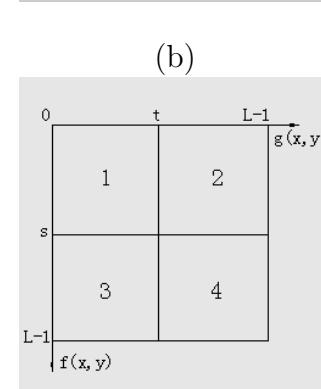
For one image we can build a 2D histogram with i and j as the two dimensions. For example, Figure 1(b) shows the 2D histogram of the image in Figure 1(a), and the projection of the 2D histogram is shown in Figure 1(c). By means of an arbitrary threshold vector (s, t) , the 2D histogram can be divided into four areas as illustrated in Figure 1(d). Pixels belonging to the background or the foreground should contribute mainly to the near-diagonal elements, as the areas of the background or the foreground are relatively smooth and there is little difference between the original gray level and the smoothed one. On the contrary, most pixels of noise and edges are away from the diagonal. Therefore, pixels in regions 1 and 4 can be considered as in foreground (or background) and background (or foreground) respectively, whereas those in regions 2 and 3 can be regarded as noise and edge. In this way, we obtain a segmentation of the image.

3. Problems of 2D Otsu's Method

One problem of the traditional 2D Otsu's method is that it is sensitive to Salt&Pepper noise. If an image is corrupted by Salt&Pepper noise, we might not be able to obtain the optimal threshold vector and reasonable segmentation results. An example of this problem is shown in Figure 2 where the image is corrupted by Salt&Pepper noise ($\delta = 0.1$). It is evident from



(a)



(b)

(c)

(d)

Figure 1: Illustration of the 2D histogram. (a) The original image, (b) 2D histogram, (c) projection of 2D histogram, (d) regional division.

Figure 2(h) that the thresholds s and t obtained with the traditional 2D Otsu's method differ widely. This enable a large number of near-diagonal pixels belonging to foreground or background to be located in region 3 (pointed by the red arrow) of the 2D histogram. From Section 2 we know that these pixels will be regarded as noise and edges incorrectly. In order to illustrate this problem more evidently, we show in Figure 2(d) the pixels in regions 1 and 4 by dark pixels, and those in regions 2 and 3 by white pixels. It is evident from Figure 2(d) that a large amount of foreground pixels are located in regions 2 and 3 and regarded as noise and edges by the traditional 2D Otsu's method. To accomplish image segmentation, we label the pixels in regions 2 and 3 as belonging to foreground and obtain the segmentation results illustrated in Figure 2(e). On the contrary, we obtain the segmentation result in Figure 2(f) if the pixels in regions 2 and 3 are labeled as belonging to background. While the segmentation in Figure 2(e) can be regarded as similar to the correct one in Figure 2(c), the segmentation in Figure 2(f) is a total failure. This implies that in order to obtain satisfactory segmentation results, we must determine the appropriate assignment of the pixels of regions 2 and 3, which is usually not a trivial task. The reason for this problem lies in that the traditional 2D Otsu's method yields an incorrect optimal threshold vector in the presence of Salt&Pepper noise. In other words, the traditional 2D Otsu's method is sensitive to Salt&Pepper noise.

In the example shown in Figure 2, if we label the pixels in regions 2 and 3 as in foreground, the segmentation result in Figure 2(e) is still acceptable.

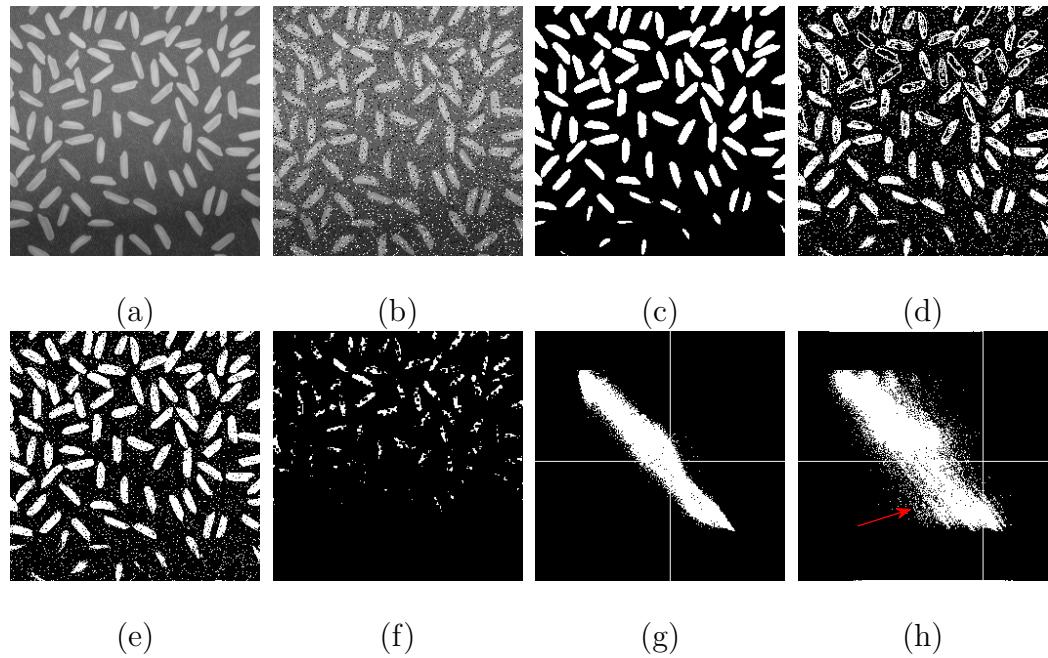


Figure 2: Illustration of the effect of Salt&Pepper noise on 2D Otsu's method. (a) The original image, (b) the image corrupted by Salt&Pepper noise ($\delta = 0.1$), (c) segmentation result of (a) using the traditional 2D Otsu's method, (d) the pixels in regions 2 and 3 with the traditional 2D Otsu's method, (e) segmentation result of (b) using the traditional 2D Otsu's method where pixels in regions 2 and 3 are assigned to foreground, (f) segmentation result of (b) using the traditional 2D Otsu's method where pixels in regions 2 and 3 are assigned to background, (g) projection of 2D histogram of (a), (h) projection of 2D histogram of (b).

1
2
3
4
5 However, we find that in case of low SNR, the segmentation performance of
6 the traditional 2D Otsu's method is not satisfactory with either assignment
7 of the pixels in regions 2 and 3. One example is shown in Figure 3, where the
8 pixels in regions 2 and 3 are labeled to be in foreground and in background,
9 yielding the segmentation results in Figure 3(e) and Figure 3(f) respectively.
10 Obviously both segmentation results are not satisfactory in comparison
11 with the correct one in Figure 3(c). The reason lies in the assignment of the
12 pixels in regions 2 and 3 in the traditional 2D Otsu's method. As shown in
13 Figure 3(d), the pixels in regions 2 and 3 are scattered in both foreground
14 and background widely. Since in the traditional 2D Otsu's method the pixels
15 in regions 2 and 3 are usually simply labeled to be in foreground or in back-
16 ground, we find that a large amount of pixels will be assigned incorrectly
17 with either assignment. In order to improve the segmentation performance
18 of the traditional 2D Otsu's method, we need a better method to deal with
19 the pixels in regions 2 and 3, rather than just labeling all of them to be in
20 background or in foreground.

21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40 **4. Our Method**
41
42

43 In this section, we present a robust 2D Otsu's method for image seg-
44 mentation, called MAOTSU_2D, to solve the problems of the traditional 2D
45 Otsu's method.
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

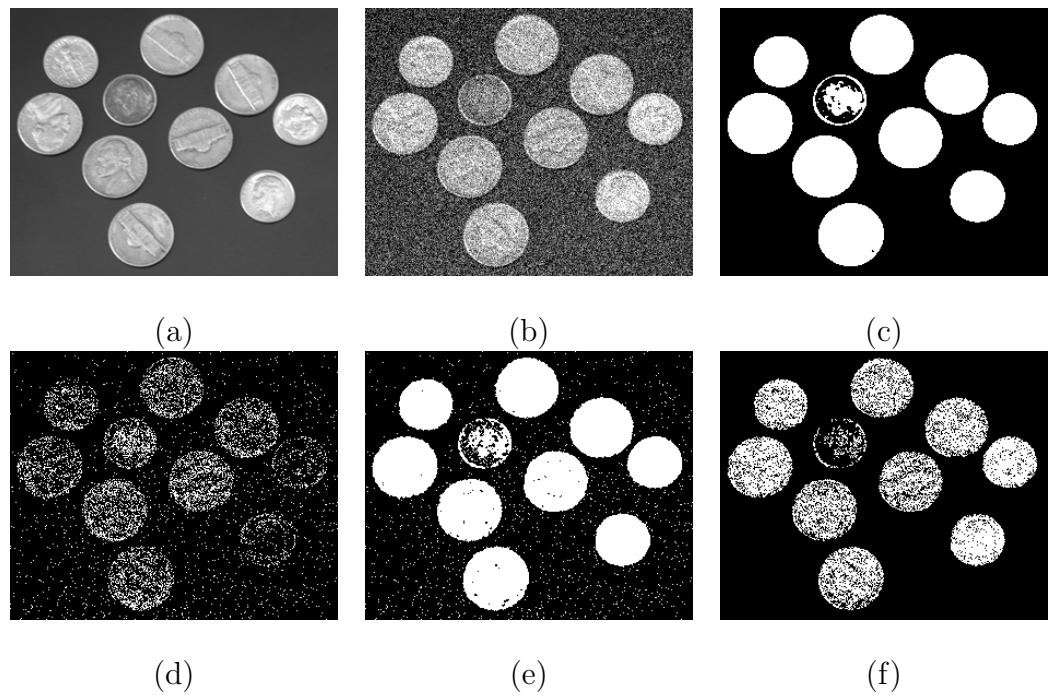


Figure 3: Segmentation results of the traditional 2D Otsu's method in case of low SNR.
 (a) The original image, (b) the original image corrupted by Gaussian noise ($\sigma^2 = 0.03$),
 (c) segmentation result of the original image, (d) the pixels in regions 2 and 3, (e) segmentation of the corrupted image by assigning pixels in regions 2 and 3 to be in foreground,
 (f) segmentation of the corrupted image by assigning pixels in regions 2 and 3 to be in background.

1
2
3
4
5 4.1. Building the New 2D Histogram
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27

In the traditional 2D Otsu's method we use the gray level and average gray level of pixels to build the 2D histogram. Since average filtering is suitable for removing Gaussian noise but not good at removing Salt&Pepper noise, the traditional 2D Otsu's method is not robust against Salt&Pepper noise. On the other hand, median filtering is able to remove Salt&Pepper noise effectively. This observation motivates us to use both average filtering and median filtering in 2D Otsu's method, in order to obtain the robustness to both Gaussian and Salt&Pepper noise. Therefore we call this method as Median-Average 2D Otsu's method (MAOTSU_2D). Our method is detailed as follows.

First, we calculate the median of pixels within $k \times k$ neighborhood in an image $f(x, y)$ and obtain the median image $m(x, y)$ as

$$m(x, y) = \text{med}\{f(x + a, y + b) | a, b = -\frac{k-1}{2}, -\frac{k-3}{2} \dots \frac{k-1}{2}\}. \quad (3)$$

In the next step we calculate the average image $G(x, y)$ of the median image $m(x, y)$ as

$$G(x, y) = \frac{1}{k^2} \sum_{c=-(k-1)/2}^{(k-1)/2} \sum_{d=-(k-1)/2}^{(k-1)/2} m(x + c, y + d). \quad (4)$$

Since we smooth the image with median filtering followed by average filtering, we call the combination of these filters as median-average filter, and the smoothed image as median-average image. With the median-average image $G(x, y)$ of the image $f(x, y)$, we are able to build the 2D histogram using

1
2
3
4
5 $f(x, y)$ and $G(x, y)$ as the two dimensions. Compared with the traditional
6 2D histogram based on $f(x, y)$ and $g(x, y)$, our histogram is less sensitive to
7 Gaussian noise and Salt&Pepper noise, as will be shown in Section 5. Next
8 we illustrate it with an example when an image is corrupted by Salt&Pepper
9 noise.
10
11

12 We firstly compare the robustness to Salt&Pepper noise of the average
13 filter with (1) and the median-average filter with (4). Impulse noise, includ-
14 ing Salt&Pepper noise and random-valued impulse noise, are added into the
15 image by replacing a part of the image pixels with noise values and leaving
16 the remainder unchanged. For Salt&Pepper noise, the gray levels of the noisy
17 pixels are equal to the minimum gray level or the maximum one with equal
18 probability (Hwang and Haddad, 1995; Xiong and Yin, 2012).
19
20
21
22
23
24
25
26
27
28
29

30 We divide pixels of an image corrupted by Salt&Pepper noise into three
31 types, i.e., uncorrupted pixels, center corrupted pixels and neighborhood cor-
32 rupted pixels. The center corrupted pixel refers to the pixel whose gray value
33 is changed as the one of noise. In contrast, the neighborhood corrupted pixel
34 itself is not corrupted by noise, but some pixel in its neighborhood are cor-
35 rupted. In order to compare the robustness to noise of the average filter and
36 the median-average filter, we take the 3×3 windows of one center corrupted
37 pixel and one neighborhood corrupted pixel, and show these two windows in
38 Figure 4(a) and Figure 4(b), respectively. With the center corrupted pixel,
39 the original gray value, average gray value and median-average gray value
40 are $f = 93.00$, $g = 109.22$ and $G = 89.33$, respectively. With the neighbor-
41
42
43
44
45
46
47
48
49
50
51
52
53
54

| | | |
|----|-----|-----|
| 89 | 87 | 82 |
| 92 | 255 | 97 |
| 96 | 83 | 102 |

$f = 93.00$

| | | |
|-----|-----|-----|
| 0 | 192 | 186 |
| 194 | 196 | 188 |
| 0 | 195 | 192 |

$f = 196.00$

(a) (b)

Figure 4: 3×3 window with Salt&Pepper noise. (a) The pixel itself is corrupted by noise, (b) the pixels in the neighborhood are corrupted by noise.

hood corrupted pixel, the three gray values are $f = 196.00$, $g = 149.22$ and $G = 189.44$, respectively. In both cases we see that the median-average filter provides a better estimation of the original gray levels than the average filter. This means that that median-average filter is more robust to Salt&Pepper noise than average filter. This observation can be further confirmed by the comparison of 2D histograms.

Figure 5 shows the traditional 2D histograms and our histograms of the original image in Figure 2(a) and the corrupted image in Figure 2(b) in the top row. For better illustration, we also show the projections of these 2D histograms in the bottom row. From Figure 5 we observe that with the original uncorrupted image, the traditional 2D Otsu's method performs very

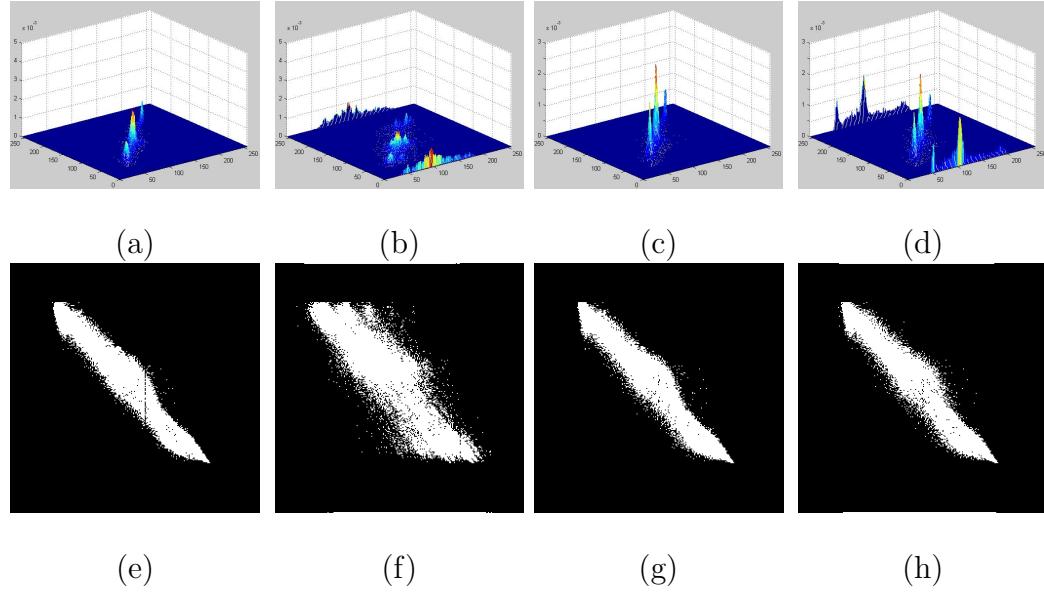


Figure 5: Comparison of the 2D histograms. (a) The traditional 2D histogram of Figure 2(a), (b) the traditional 2D histogram of Figure 2(b), (c) our 2D histogram of Figure 2(a), (d) our 2D histogram of Figure 2(b), (e) the projection of (a), (f) the projection of (b), (g) the projection of (c), (h) the projection of (d).

similar to ours, as shown in the comparison of Figure 5(e) and Figure 5(g). With the image corrupted by Salt&Pepper noise, the result of our method is very close to the one with the uncorrupted image, whereas the result of the traditional 2D Otsu's method differs widely from the one with the uncorrupted image. This confirms that our method is more robust to noise than the traditional 2D Otsu's method.

4.2. Calculating the Optimal Thresholds on 1D Histogram

After building the new 2D histogram, we need a threshold vector for regional division and then image segmentation. In order to obtain the best

segmentation results, we need to find out the optimal threshold vector. In the following we firstly introduce how to derive the optimal threshold vector in the traditional 2D Otsu's method.

Let's say that the pixels in an image are divided into two classes, i.e., foreground and background. Without loss of generality, we assume that the foreground class C_0 corresponds to the dark area in the image, and the bright area belongs to the background class C_1 . The foreground and background correspond to the pixels in region 1 and 4 in the 2D histogram respectively. The probabilities of C_0 and C_1 are denoted as

$$\omega_0 = P(C_0) = \sum_{i=0}^s \sum_{j=0}^t p_{ij} = \omega_0(s, t), \quad (5)$$

$$\omega_1 = P(C_1) = \sum_{i=s+1}^{L-1} \sum_{j=t+1}^{L-1} p_{ij} = \omega_1(s, t). \quad (6)$$

Correspondingly, we represent the class mean vectors as

$$u_0 = (u_{0i}, u_{0j})^T = \left(\sum_{i=0}^s \sum_{j=0}^t \frac{ip_{ij}}{\omega_0}, \sum_{i=0}^s \sum_{j=0}^t \frac{jp_{ij}}{\omega_0} \right)^T, \quad (7)$$

$$u_1 = (u_{1i}, u_{1j})^T = \left(\sum_{i=s+1}^{L-1} \sum_{j=t+1}^{L-1} \frac{ip_{ij}}{\omega_1}, \sum_{i=s+1}^{L-1} \sum_{j=t+1}^{L-1} \frac{jp_{ij}}{\omega_1} \right)^T, \quad (8)$$

and then the total mean vector of the 2D histogram as

$$u_T = (u_{Ti}, u_{Tj})^T = \left(\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} ip_{ij}, \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} jp_{ij} \right)^T. \quad (9)$$

The between-class variance matrix is defined as

$$\sigma_b = \sum_{k=0}^1 \omega_k \left[(u_k - u_T)(u_k - u_T)^T \right]. \quad (10)$$

1
2
3
4
5 We use the trace of σ_b to measure between-class variance, i.e.,
6
7
8
9

$$tr\sigma_b(s, t) = \sum_{k=0}^1 \left(\omega_k \left[(u_{ki} - u_{Ti})^2 + (u_{kj} - u_{Tj})^2 \right] \right). \quad (11)$$

10 In order to simplify calculation, we consider the following approximate
11 form of Eq. (11). In most cases, the probability of the pixels away from the
12 diagonal of the 2D histogram is negligible. Therefore it's easy to verify that
13
14
15
16
17

$$\omega_0 + \omega_1 \approx 1, \quad u_T \approx \omega_0 u_0 + \omega_1 u_1. \quad (12)$$

21 Based on Eq. (12), we rewrite Eq. (11) as
22
23

$$tr\sigma_b(s, t) = \frac{[u_{Ti}\omega_0 - u_i(s, t)]^2 + [u_{Tj}\omega_0 - u_j(s, t)]^2}{\omega_0(1 - \omega_0)}, \quad (13)$$

27 where
28
29

$$u_i(s, t) = \sum_{i=0}^s \sum_{j=0}^t ip_{ij}, \quad u_j(s, t) = \sum_{i=0}^s \sum_{j=0}^t jp_{ij}. \quad (14)$$

32 The optimal threshold vector (s^*, t^*) can then be obtained by maximizing
33
34 $tr\sigma_b(s, t)$, i.e.,
35

$$tr\sigma_b(s^*, t^*) = \max_{0 \leq s, t \leq L-1} \{tr\sigma_b(s, t)\}. \quad (15)$$

39 In order to further improve the robustness to Salt&Pepper noise, we
40 present a new method to find the optimal threshold vector. Instead of searching
41 for the optimal threshold vector in the 2D space, we choose to determine
42 the optimal threshold on each dimension of the 2D histogram respectively
43 using 1D Otsu's method. Compared with the traditional method, our
44 method is more robust to Salt&Pepper noise while being computationally
45 more efficient.
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64

It has been shown in (Yue et al., 2009) that Eq. (15) can be simplified as

$$\begin{aligned}
 tr\sigma_b(s^*, t^*) &= \max_{0 \leq s, t \leq L-1} \{tr\sigma_b(s, t)\} \\
 &= \max_{0 \leq s, t \leq L-1} \{tr\sigma_{b_i}(s) + tr\sigma_{b_j}(t)\} \\
 &= \max_{0 \leq s \leq L-1} \{tr\sigma_{b_i}(s)\} + \max_{0 \leq t \leq L-1} \{tr\sigma_{b_j}(t)\} \\
 &= tr\sigma_{b_i}(s^*) + tr\sigma_{b_j}(t^*),
 \end{aligned} \tag{16}$$

on condition that the probability of pixels in regions 2 and 3 is negligible,

i.e.,

$$\sum_{i=0}^s \sum_{j=t+1}^{L-1} p_{ij} = 0, \quad \sum_{i=s+1}^{L-1} \sum_{j=0}^t p_{ij} = 0. \tag{17}$$

In Eq. (16), we have

$$tr\sigma_{b_i}(s) = \omega_{i0}(U_{i0} - U_{ti})^2 + \omega_{i1}(U_{i1} - U_{ti})^2, \tag{18}$$

$$tr\sigma_{b_j}(t) = \omega_{j0}(U_{j0} - U_{tj})^2 + \omega_{j1}(U_{j1} - U_{tj})^2, \tag{19}$$

and

$$F_i = \sum_{j=0}^{L-1} p_{ij}, \quad G_j = \sum_{i=0}^{L-1} p_{ij}, \tag{20}$$

$$\omega_{i0} = \sum_{i=0}^s F_i, \quad \omega_{i1} = \sum_{i=s+1}^{L-1} F_i, \tag{21}$$

$$\omega_{j0} = \sum_{j=0}^s G_j, \quad \omega_{j1} = \sum_{j=s+1}^{L-1} G_j, \tag{22}$$

$$U_{i0} = \frac{\sum_{i=0}^s i F_i}{\omega_{i0}}, \quad U_{i1} = \frac{\sum_{i=s+1}^{L-1} i F_i}{\omega_{i1}}, \tag{23}$$

$$U_{j0} = \frac{\sum_{j=0}^s j G_j}{\omega_{j0}}, \quad U_{j1} = \frac{\sum_{j=s+1}^{L-1} j G_j}{\omega_{j1}}, \tag{24}$$

$$U_{ti} = \sum_{i=0}^{L-1} iF_i, \quad U_{tj} = \sum_{j=0}^{L-1} jG_j. \quad (25)$$

We present the derivation of Eq. (16) in the following briefly and refer interested reader to (Yue et al., 2009) for details. From Eq. (5) and Eq. (17) we have

$$\begin{aligned} \omega_0 &= \sum_{i=0}^s \sum_{j=0}^t p_{ij} \\ &= \sum_{i=0}^s \sum_{j=0}^t p_{ij} + \sum_{i=0}^s \sum_{j=t+1}^{L-1} p_{ij} \\ &= \sum_{i=0}^s \sum_{j=0}^{L-1} p_{ij} = \sum_{i=0}^s F_i = \omega_{i0}. \end{aligned} \quad (26)$$

With similar derivation we can obtain

$$\omega_0 = \omega_{j0}, \quad (27)$$

$$\omega_1 = \omega_{i1} = \omega_{j1}, \quad (28)$$

$$u_{0i} = U_{i0}, \quad u_{0j} = U_{j0}, \quad (29)$$

$$u_{1i} = U_{i1}, \quad u_{1j} = U_{j1}, \quad (30)$$

$$u_{Ti} = U_{ti}, \quad u_{Tj} = U_{tj}. \quad (31)$$

Substituting Eqs. (26) to (31) into Eq. (11), which is equivalent to Eq. (13)

on the condition of Eq. (12), we have

$$\begin{aligned}
tr\sigma_b(s, t) &= \sum_{k=0}^1 \left(\omega_k \left[(u_{ki} - u_{Ti})^2 + (u_{kj} - u_{Tj})^2 \right] \right) \\
&= \omega_{i0}(U_{i0} - U_{ti})^2 + \omega_{i1}(U_{i1} - U_{ti})^2 \\
&\quad + \omega_{j0}(U_{j0} - U_{tj})^2 + \omega_{j1}(U_{j1} - U_{tj})^2 \\
&= tr\sigma_{b_i}(s) + tr\sigma_{b_j}(t).
\end{aligned} \tag{32}$$

Then it is easy to obtain Eq. (16) from Eq. (32).

If the condition in Eq. (17) is satisfied, the optimal threshold vector obtained by Eq. (16) is the same as the one obtained by using Eq. (15). This provides the ground for determining the optimal thresholds on two dimensions separately. In practice, however, the condition in Eq. (17) is usually satisfied approximately but not exactly. As a result, there exist some differences between the optimal threshold vectors obtained by traditional 2D search with Eq. (15) and by two 1D searches with Eq. (16). Surprisingly, we find that the method using two 1D searches is more robust to noise, especially Salt&Pepper noise, than the one based on the traditional 2D search.

In the following we use one example to illustrate this observation. In Figure 2(b) the image is corrupted by Salt&Pepper noise. With the traditional 2D Otsu's method, the 2D search yields the optimal threshold vector as indicated in Figure 2(h). Obviously this optimal threshold vector is wrong as a large amount of near-diagonal pixels are divided in regions 2 and 3, and thus regarded as noise and edge. However, with the same image in Figure 2(b), the two 1D searches with Eq. (16) generate the optimal threshold vector as

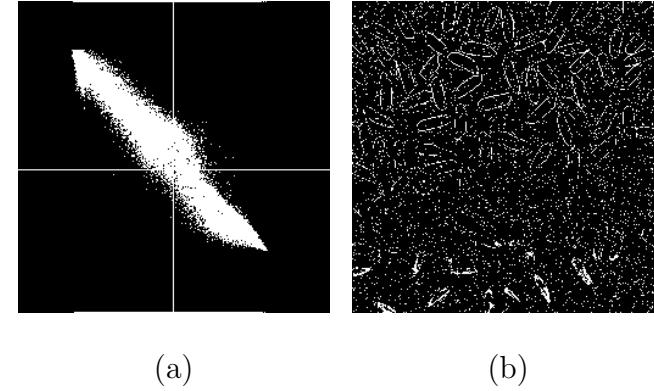


Figure 6: Illustration of the result of two 1D searches with Eq. (16). (a) The projection of the $f(x,y)$ - $G(x,y)$ histogram of Figure 2(b) using Eq. (16), (b) the pixels in regions 2 and 3 of (a).

indicated in Figure 6(a). With this correct optimal threshold vector, only a small amount of pixels are in regions 2 and 3 and labeled as noise and edge, as illustrated in Figure 6(b).

We try to explain why the method using two 1D searches has better robustness to noise as follows. The traditional 2D Otsu's method calculates the optimal threshold vector using Eq. (13), which is derived under the approximate condition Eq. (12). Let ω'_1 and u'_1 represent the approximations of ω_1 and u_1 obtained by using Eq. (12) respectively, we have

$$\omega_0 + \omega'_1 = 1, \quad u_T = \omega_0 u_0 + \omega'_1 u'_1, \quad (33)$$

and then

$$\begin{aligned} \omega'_1 &= 1 - \omega_0 \\ &= \sum_{i=s+1}^{L-1} \sum_{j=t+1}^{L-1} p_{ij} + \sum_{i=0}^s \sum_{j=t+1}^{L-1} p_{ij} + \sum_{i=s+1}^{L-1} \sum_{j=0}^t p_{ij}, \end{aligned} \quad (34)$$

$$\begin{aligned}
u'_1 &= (u'_{1i}, u'_{1j})^T \\
&= \frac{u_T - \omega_0 u_0}{\omega'_1} \\
&= \frac{1}{\omega'_1} \left(\sum_{i=s+1}^{L-1} \sum_{j=t+1}^{L-1} ip_{ij} + \sum_{i=0}^s \sum_{j=t+1}^{L-1} ip_{ij} + \sum_{i=s+1}^{L-1} \sum_{j=0}^t ip_{ij}, \right. \\
&\quad \left. \sum_{i=s+1}^{L-1} \sum_{j=t+1}^{L-1} jp_{ij} + \sum_{i=0}^s \sum_{j=t+1}^{L-1} jp_{ij} + \sum_{i=s+1}^{L-1} \sum_{j=0}^t jp_{ij} \right)^T.
\end{aligned} \tag{35}$$

Substituting Eq. (34) and Eq. (35) into Eq. (11), we obtain the equivalent form of Eq. (13) as

$$\begin{aligned}
tr\sigma_b(s, t) &= \left(\omega_0 \left[(u_{0i} - u_{Ti})^2 + (u_{0j} - u_{Tj})^2 \right] \right) \\
&\quad + \left(\omega'_1 \left[(u'_{1i} - u_{Ti})^2 + (u'_{1j} - u_{Tj})^2 \right] \right).
\end{aligned} \tag{36}$$

From Eq. (34) and Eq. (35) we find that ω'_1 and u'_1 are the probability and class mean vector of the pixels in regions 2, 3 and 4, but not in regions 4 only. In fact, the pixels in regions 2, 3 and 4 are regarded as in the same class C'_1 in the approximation in Eq. (12). This means that the class C'_1 includes not only the pixels in background but also those belonging to noise and edge. In other words, the image is divided into two classes, i.e., the foreground class C_0 , and the class C'_1 including background, noise and edge. As a result, the between-class variance in Eq. (36) or Eq. (13) is calculated between foreground and the combination of background, noise and edge, but not between foreground and background. If the probability of the pixels of noise and edge is not negligible, the condition in Eq. (12) is not satisfied and the between-class variance represented by Eq. (36) or Eq. (13) is no

longer applicable. Furthermore, Eq. (36) or Eq. (13) cannot guarantee an appropriate division of region 2, 3 and 4. As a result, a great number of pixels may be wrongly assigned to regions 2 and 3 when the two optimal thresholds differ significantly from each other. In images corrupted by Salt&Pepper noise, the gray values of noisy pixels are equal to the minimum or maximum gray values, and are significantly different from those of neighboring pixels. Therefore these pixels will be located far away from the main diagonal of the 2D histogram. In this case, the probability of noisy pixels and edges can not be neglected and they have evident influence on the optimal threshold vector in the traditional 2D search. Consequently, a great number of pixels may be wrongly assigned to regions 2 and 3 and thus the traditional 2D search is likely to yield incorrect optimal threshold vector, as illustrated in Figure 2(h).

In contrast, the method using two 1D searches separately is less likely to be influenced by Salt&Pepper noise. Since the two 1D searches are conducted on the gray level histogram and the median-average histogram separately, either threshold is unlikely to be affected by noise significantly. As a result, the two thresholds tend to be quite close to each other, and only a small fraction of pixels appear in regions 2 and 3, as illustrated in Figure 6(a).

Another advantage of the method using two 1D searches lies in the computation efficiency. Compared with the traditional 2D Otsu's method using 2D search, this method reduces the computation complexity remarkably.

1
2
3
4
5 *4.3. Region Post-processing*
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

With the obtained optimal threshold vector (s, t) , we are able to divide the 2D histogram into four regions. Here s is the gray level threshold, and t is the threshold of median-average gray level. Without loss of generality, we assume that the pixels in region 1 belong to background and those in region 4 belong to foreground. As stated in Section 3, simply labeling all the pixels in regions 2 and 3 as in foreground or background is not a good option. We need a better region post-processing method to deal with the pixels in regions 2 and 3.

The pixels in regions 2 and 3 are usually of noise and edge, and distributed in both foreground and background. If we label all of them as in foreground or background, we will find many noisy points in the segmentation result, as illustrated in Figure 3(e) and Figure 3(f). In order to remove these noisy points, we should label each pixel in regions 2 and 3 based on the assignments of its neighbors. Our method to achieve this aim is as follows. For each pixel (x, y) in regions 2 and 3, we calculate its median-average gray level $G(x, y)$ using Eq. (4) and then compare $G(x, y)$ with the threshold s to determine the label of this pixel. If $G(x, y) < s$, then the pixel (x, y) is labeled as in background. Otherwise, this pixel is labeled as in foreground.

We illustrate the effect of our region post-processing method on the images corrupted by Gaussian noise and Salt&Pepper noise in Figure 7 and Figure 8, respectively. From Figure 7(b-c) and Figure 8(b-c), we notice that there are a large number of noisy points in foreground or background area of

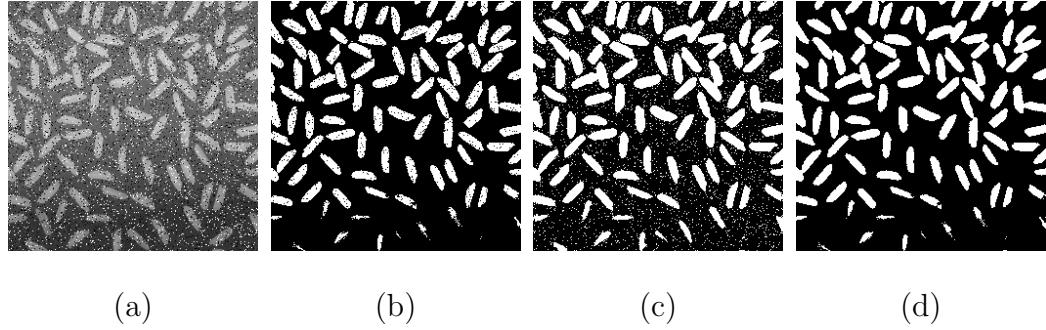


Figure 7: Illustration of the effect of region post-processing on Salt&Pepper noise. (a) Figure 2(a) corrupted by Salt&Pepper noise ($\delta = 0.1$) (b) segmentation result of the corrupted image by assigning pixels in regions 2 and 3 to be in background, (c) segmentation of the corrupted image by assigning pixels in regions 2 and 3 to be in foreground, (d) segmentation of the corrupted image with region post-processing.

the image by simply labeling all the pixels in regions 2 and 3 to background or foreground respectively, although the segmentation results are already better than those of the traditional 2D Otsu's method. In contrast, Figure 7(d) and Figure 8(d) show that with our region post-processing method, the noisy points in both foreground and background are removed effectively. This validates the effectiveness of our region post-processing method.

4.4. Other Methods Derived from the General Framework

MAOTSU_2D can be regarded as a general framework. In addition to the method presented above, some other typical 2D Otsu's methods can be derived from the framework by building the 2D histogram with different filters. If we replace the median-average filter with average filter only, MAOTSU_2D reduces into AOTSU_2D, which is the traditional 2D Otsu's method. On the

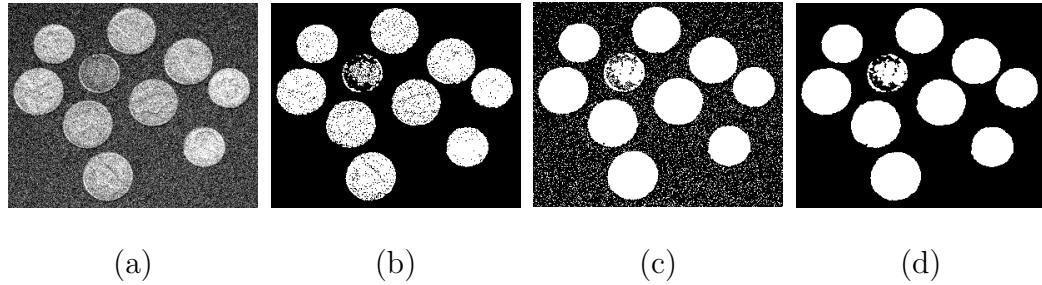


Figure 8: Illustration of the effect of region post-processing on Gaussian noise. (a) Figure 3(a) corrupted by Gaussian noise ($\sigma^2 = 0.03$) (b) segmentation result of the corrupted image by assigning pixels in regions 2 and 3 to be in background, (c) segmentation of the corrupted image by assigning pixels in regions 2 and 3 to be in foreground, (d) segmentation of the corrupted image with region post-processing.

other hand, MAOTSU_2D reduces to MOTSU_2D if the median-average filter is replaced by median filter.

These methods are suitable for different application. AOTSU_2D is adapted to images corrupted by Gaussian noise as average filter is able to remove Gaussian noise effectively, whereas MOTSU_2D is suitable for images corrupted by impulse noises like Salt&Pepper noise since median filter is adopted. However, it is generally difficult to choose between AOTSU_2D and MOTSU_2D without the prior knowledge of the noise. Compared with AOTSU_2D and MOTSU_2D, MAOTSU_2D is independent of the type of noise and suitable for images corrupted by unknown noise.

It is important to select the proper method according to the particular application. In the following we illustrated it with two examples. In order to observe the robustness to noises of different types and different levels,

we apply the three methods to images corrupted by low noise and heavy noise respectively. First, we test them on images corrupted by low Gaussian noise and low Salt&Pepper noise respectively as shown in Figure 9. We note that MAOTSU_2D has the best robustness to both types of noise, while AOTSU_2D and MOTSU_2D show comparable or inferior robustness to MAOTSU_2D in the presence of low Gaussian noise and low Salt&Pepper noise respectively. On the other hand, AOTSU_2D and MOTSU_2D preserve more details (pointed by the red arrows) of the segmented image than MAOTSU_2D, as shown in Figure 9(c) and Figure 9(f). Therefore, in the case that the noise type is known and the noise level is low, AOTSU_2D is a superior option for Gaussian noise and MOTSU_2D is an appropriate choice for Salt&Pepper noise, as they make better balance between robustness to noise and effectiveness of preserving details.

Then the three methods are applied to images corrupted by heavy noise and the segmentation results are shown in Figure 10. We observe that AOTSU_2D and MOTSU_2D are sensitive to noise types, and they are outperformed by MAOTSU_2D evidently in the robustness to both heavy Gaussian and heavy Salt&Pepper noise. This implies that MAOTSU_2D is a preferable selection in case of noise of unknown types or heavy noise. Since in practice the noise types are usually unknown and images are generally corrupted by mixed noise, MAOTSU_2D seems to be a better choice than the other two methods.

In addition, our general framework can also be applied to the 2D maxi-

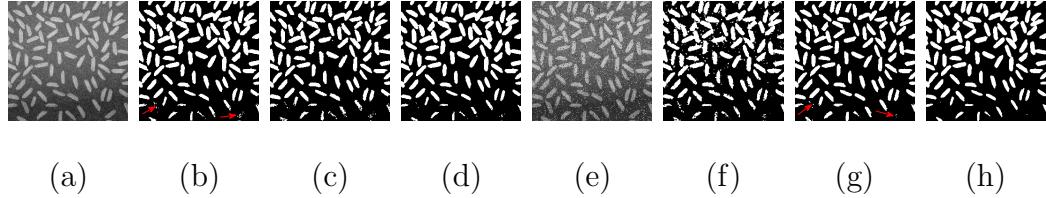


Figure 9: Comparison of segmentation results on the image corrupted by low noise. (a) The image corrupted by low Gaussian noise ($(\sigma^2 = 0.003)$), (b) segmentation result of (a) using AOTSU_2D, (c) segmentation of (a) using MOTSU_2D, (d) segmentation of (a) using MAOTSU_2D, (e) the image corrupted by low Salt&Pepper noise ($\delta = 0.05$), (f) segmentation of (e) using AOTSU_2D, (g) segmentation of (e) using MOTSU_2D, (h) segmentation of (e) using MAOTSU_2D.

mum entropy method. By following the same procedures as presented bavoe, a robust Median-Average 2D maximum entropy method can be obtained.

5. Experimental Results

In this section we test our MAOTSU_2D method in experiments on both synthetic and real images with different types of noise. In order to evaluate its performance, we also compare the results of our method with those of some other methods, including Otsu's method (OTSU) (Otsu, 1979), iterative triclass Otsu's method (ITOTSU) (Cai et al., 2014), the traditional 2D Otsu's method (2D OTSU) (Gong et al., 1998), modified 2D Otsu's algorithm (MOTSU_2D) (Chen et al., 2012), a decomposition based 2D Otsu's method (DBOTSU_2D) (Yue et al., 2009), an equivalent 3D Otsu's method (EOTSU_3D) (Sthitpattanapongsa and Srinark, 2012), maximum entropy (ENTROPY) (Pal and Pal, 1989), fast recursive 2D maximum entropy (2D

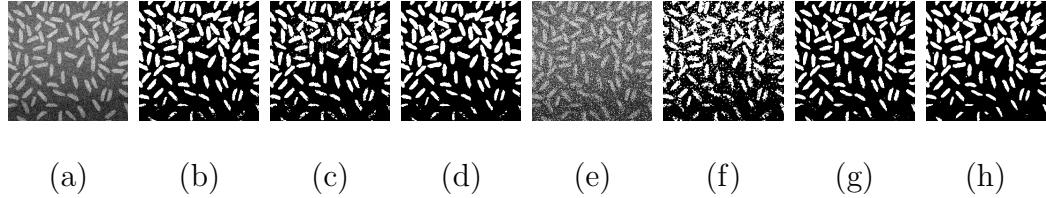


Figure 10: Comparison of segmentation results on the image corrupted by heavy noise.

(a) The image corrupted by heavy Gaussian noise ($(\sigma^2 = 0.01)$), (b) segmentation result of (a) using AOTSU_2D, (c) segmentation of (a) using MOTSU_2D, (d) segmentation of (a) using MAOTSU_2D, (e) the image corrupted by heavy Salt&Pepper noise ($\delta = 0.2$), (f) segmentation of (e) using AOTSU_2D, (g) segmentation of (e) using MOTSU_2D, (h) segmentation of (e) using MAOTSU_2D.

ENTROPY) (Gong et al., 1998), k-means (K-MEANS) (Macqueen, 1967), fast generalized fuzzy c-means algorithm (FGFCM) (Cai et al., 2007) and fuzzy local information c-means (FLICM) (Krinidis and Chatzis, 2010).

In all the following experiments, we adopt the parameters $c = 2$, $m = 2$, $\varepsilon = 0.00001$, $N_R = 8$ (a 3×3 window centered around each pixel, except for the central pixel itself), the maximum iteration count $maxiter = 100$ for both FGFCM and FLICM, and also $\lambda_s = 3$, $\lambda_g = 6$, $\alpha = 6$ for FGFCM. The pixels in regions 2 and 3 are all assigned to background with 2D OTSU, MOTSU_2D, DBOTSU_2D, EOTSU_3D and 2D ENTROPY. The iteration of ITOTSU is terminated when the thresholds of two consecutive iterations are less than 1.

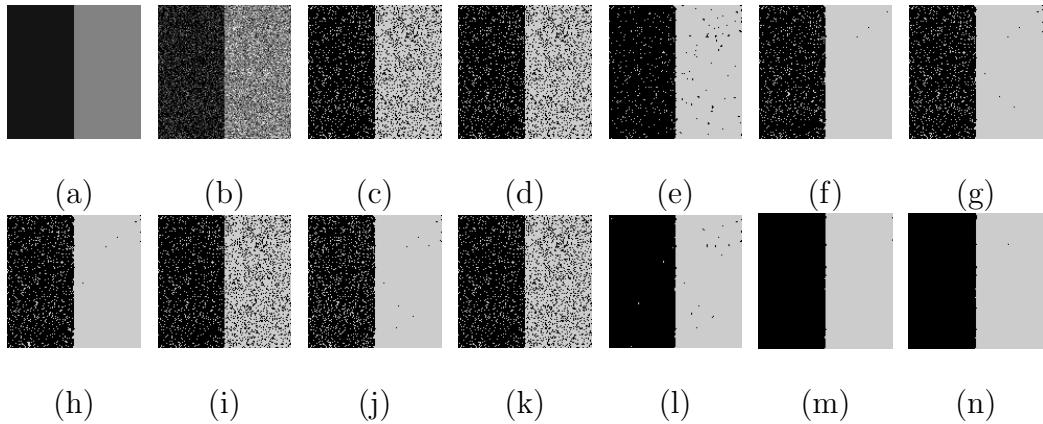


Figure 11: Comparison of segmentation results from different methods on a synthetic image corrupted by Gaussian noise ($\sigma^2 = 0.04$). (a) The original image, (b) the same image corrupted by Gaussian noise ($\sigma^2 = 0.04$), (c) using OTSU, (d) using ITOTSU, (e) using 2D OTSU, (f) using MOTSU_2D, (g) using DBOTSU_2D, (h) using EOTSU_3D, (i) using ENTROPY, (j) using 2D ENTROPY, (k) using K-MEANS, (l) using FGFCM, (m) using FLICM, (n) using MAOTSU_2D.

5.1. Results on Synthetic Image

First, we apply all these 12 methods to a synthetic image corrupted by Gaussian noise and Salt&Pepper noise respectively. As shown in Figure 11(a), the synthetic image is of size 128×128 and the gray level values of the two classes are 20 and 130 respectively. The size of neighborhood window k is all set to 3 in the rest of this paper. Figure 11 and Figure 12 illustrate the segmentation results on the image corrupted by Gaussian noise ($\sigma^2 = 0.04$) and Salt&Pepper noise ($\delta = 0.1$) respectively. It is evident that our method generates the best segmentation results in the presence of Salt&Pepper noise, and the second best segmentation results in the presence of Gaussian noise.

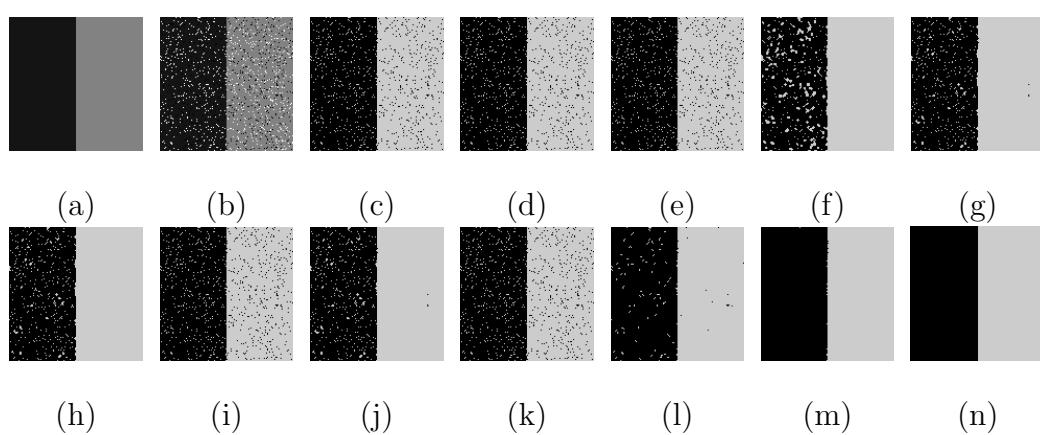


Figure 12: Comparison of segmentation results from different methods on a synthetic image corrupted by Salt&Pepper noise ($\delta = 0.1$). (a) The original image, (b) the same image corrupted by Salt&Pepper noise ($\delta = 0.1$), (c) using OTSU, (d) using ITOTSU, (e) using 2D OTSU, (f) using MOTSU_2D, (g) using DBOTSU_2D, (h) using EOTSU_3D, (i) using ENTROPY, (j) using 2D ENTROPY, (k) using K-MEANS, (l) using FGFCM, (m) using FLICM, (n) using MAOTSU_2D.

Table 1: Correct Segmentation Rate (CSR %) of Twelve Methods on Synthetic Image

| | OTSU | ITOTSU | 2D OTSU | MOTSU_2D | DBOTSU_2D | EOTSU_3D | ENTROPY | 2D ENTROPY | K-MEANS | FGFCM | FLICM | MAOTSU_2D |
|---------------------------------|-------|--------|---------|----------|-----------|----------|---------|------------|---------|-------|--------------|--------------|
| Gaussian ($\sigma^2 = 0.02$) | 93.34 | 93.35 | 99.73 | 97.86 | 97.51 | 97.50 | 93.31 | 96.87 | 93.34 | 99.94 | 99.96 | 99.95 |
| Gaussian ($\sigma^2 = 0.04$) | 85.47 | 85.51 | 97.87 | 95.69 | 95.09 | 95.00 | 85.38 | 92.94 | 85.48 | 99.61 | 99.85 | 99.76 |
| Gaussian ($\sigma^2 = 0.06$) | 80.16 | 80.29 | 84.65 | 94.52 | 93.26 | 92.99 | 80.10 | 90.28 | 80.15 | 98.60 | 99.65 | 99.13 |
| Gaussian ($\sigma^2 = 0.08$) | 76.91 | 77.00 | 77.70 | 92.68 | 91.50 | 91.11 | 76.88 | 87.67 | 76.92 | 97.21 | 99.28 | 98.13 |
| Gaussian Average | 83.97 | 84.04 | 89.99 | 95.19 | 94.34 | 94.15 | 83.92 | 91.94 | 83.97 | 98.84 | 99.69 | 99.24 |
| Salt&Pepper ($\delta = 0.05$) | 97.45 | 97.45 | 97.47 | 97.83 | 98.62 | 98.62 | 97.45 | 84.32 | 97.45 | 99.69 | 99.97 | 99.99 |
| Salt&Pepper ($\delta = 0.10$) | 94.98 | 94.98 | 95.00 | 94.70 | 97.06 | 97.09 | 94.98 | 97.16 | 94.98 | 98.88 | 99.93 | 99.97 |
| Salt&Pepper ($\delta = 0.15$) | 92.46 | 92.46 | 92.49 | 90.93 | 95.22 | 95.28 | 92.46 | 95.36 | 92.46 | 97.72 | 99.89 | 99.92 |
| Salt&Pepper ($\delta = 0.20$) | 90.11 | 90.11 | 90.12 | 86.50 | 93.02 | 93.12 | 90.11 | 94.11 | 90.11 | 96.18 | 99.82 | 99.84 |
| Salt&Pepper Average | 93.75 | 93.75 | 93.77 | 92.49 | 95.98 | 96.03 | 93.75 | 92.74 | 93.75 | 98.12 | 99.90 | 99.93 |

For quantitative comparison, we use the evaluation measure called correct segmentation rate (CSR), which is a modification of misclassification error (ME) (Yasnoff et al., 1977; Sezgin and Sankur, 2004) and also a special case of the segmentation accuracy (SA) (Ahmed et al., 2002; Zhang and Chen, 2004; Cai et al., 2007; Krimidis and Chatzis, 2010). CSR is defined as the number of correctly classified pixels divided by the total number of pixels, i.e.,

$$CSR = \frac{|B_A \cap B_R| + |O_A \cap O_R|}{|B_R| + |O_R|} \times 100\%, \quad (37)$$

where B_A and O_A represent the set of pixels belonging to background and foreground respectively found by the segmentation algorithm, B_R and O_R represent the set of pixels belonging to background and foreground respectively in the reference image, and $|.|$ is the cardinality of the set. The CSR varies from 0 for a totally wrongly segmented image to 100% for a perfectly segmented image. We apply these methods to the synthetic image corrupted by noise of different levels and report the CSR's in Table 5.1. In this table each CSR is the average of CSR's from ten experiments.

From Figure 11, Figure 12 and Table 5.1 we find that in all the 12 methods, the proposed MAOTSU_2D method performs the best in robustness to Salt&Pepper noise, and the second best in robustness to Gaussian noise. In the 11 methods for comparison, OTSU, ITOTSU, ENTROPY and K-MEANS are the least robust ones to Gaussian noise as they fail to take spatial correlation into account. In the case of Gaussian noise, DBOTSU_2D, MOTSU_2D and EOTSU_3D benefit from the improved calculation method and perform better than 2D Otsu and 2D ENTROPY. However, they are still inferior to MAOTSU_2D due to the absence of region post-processing. On the other hand, FGFCM, FLICM and MAOTSU_2D remove almost all the added noise. As a result, MAOTSU_2D performs slightly inferior to FLICM and ranks as the second best one in the 12 methods. In the case of Salt&Pepper noise, none of 2D OTSU, MOTSU_2D and 2D ENTROPY performs satisfactorily as all the methods find the optimal threshold vector with 2D search. Although DBOTSU_2D and EOTSU_3D search the two optimal thresholds separately and show better robustness to Salt&Pepper noise, they are still outperformed by our MAOTSU_2D method. Our explanation for this observation is that DBOTSU_2D dose not take the neighborhood median into consideration and neither of them conducts region post-processing. In addition, our MAOTU_2D surpasses FGFCM and FLICM and shows the best robustness to Salt&Pepper noise in the 12 methods.

1
2
3
4
5 *5.2. Results on Real Image*
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

In this part we apply these methods to segment real images corrupted by Gaussian noise and Salt&Pepper noise and report the results in Figure 13, Figure 14, Figure 15 and Figure 16. Then in Table 5.2 we present a quantitative comparison of the segmentation results on Figure 13(a), Figure 14(a), Figure 15(a) and Figure 16(a), where the comparison score (Masulli and Schenone, 1999; Zhang and Chen, 2004; Cai et al., 2007; Krinidis and Chatzis, 2010) is used as the evaluation criterion. All the test images are corrupted by Gaussian noise at different levels, i.e., $\sigma^2 = 0.02$ to 0.12 with step 0.02 and by Salt&Pepper noise at different levels, i.e., $\delta = 0.05$ to 0.30 with step 0.05. The average comparison scores of all the levels of Gaussian noise and of Salt&Pepper noise with each image are calculated respectively and reported in Table 5.2. In addition, each comparison score is the average of comparison scores from ten experiments in this table.

The comparison score is defined as

$$S = \frac{1}{c} \sum_{i=0}^c \frac{A_i \cap R_i}{A_i \cup R_i}, \quad (38)$$

where A_i represents the amount of pixels belonging to the i th class in the segmentation and R_i represents the amount of pixels belonging to the i th class in the reference segmented image. The comparison score S indicates the similarity between the segmentation and the ground truth. The larger S is, the better the segmentation is.

It is evident from Figure 14, Figure 13, Figure 15, Figure 16 and Table

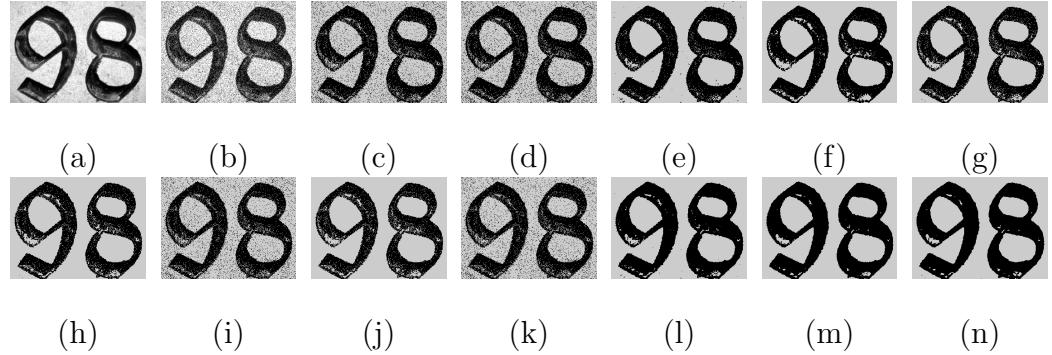


Figure 13: Comparison of segmentation results on a real image corrupted by Gaussian noise ($\sigma^2 = 0.06$). (a) The original image, (b) the same image corrupted by Gaussian noise ($\sigma^2 = 0.06$), (c) using OTSU, (d) using ITOTSU, (e) using 2D OTSU, (f) using MOTSU_2D, (g) using DBOTSU_2D, (h) using EOTSU_3D, (i) using ENTROPY, (j) using 2D ENTROPY, (k) using K-MEANS, (l) using FGFCM, (m) using FLICM, (n) using MAOTSU_2D.

5.2 that our MAOTSU_2D method performs better than the other eleven methods in experiments on all the four real images, except that its result is slightly inferior to FLICM on the image in Figure 13(a). Our method has the best average comparison score in case of Gaussian noise. With respect to Salt&Pepper noise, our MAOTSU_2D method generates the best result, except that on Figure 15(a) it generates the second best average comparison score, slightly behind FLICM. From this observation and the one in the case of synthetic image, it can be concluded that our MAOTSU_2D is superior or comparable to FLICM, and it is the most robust or second most robust to both Gaussian noise and Salt&Pepper noise in the 12 methods.

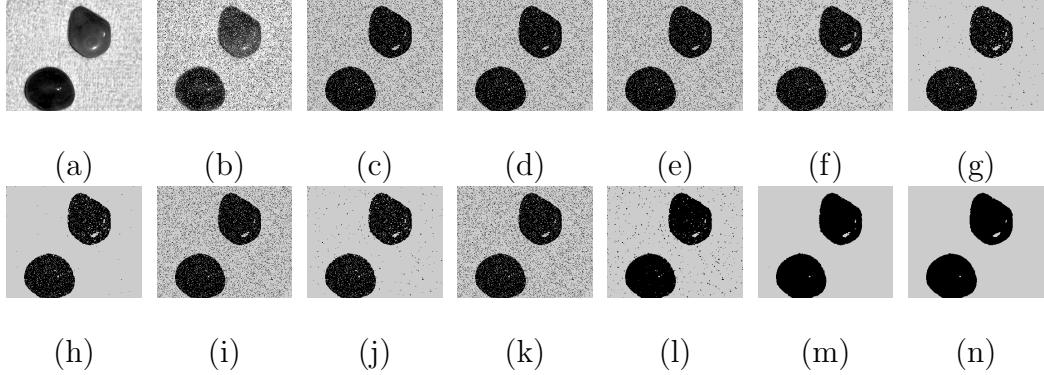


Figure 14: Comparison of segmentation results on a real image corrupted by Salt&Pepper noise ($\delta = 0.20$). (a) The original image, (b) the same image corrupted by Salt&Pepper noise ($\delta = 0.20$), (c) using OTSU, (d) using ITOTSU, (e) using 2D OTSU, (f) using MOTSU_2D, (g) using DBOTSU_2D, (h) using EOTSU_3D, (i) using ENTROPY, (j) using 2D ENTROPY, (k) using K-MEANS, (l) using FGFCM, (m) using FLICM, (n) using MAOTSU_2D.

Table 2: Comparison Scores (S) for Segmentation Results of Real Images

| | Figure 13(a) | | Figure 14(a) | | Figure 15(a) | | Figure 16(a) | | Total Average | |
|------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| | Gaussian | Salt&Pepper |
| OTSU | 0.7822 | 0.7935 | 0.7408 | 0.7919 | 0.4239 | 0.3896 | 0.6375 | 0.7535 | 0.6461 | 0.6821 |
| ITOTSU | 0.7816 | 0.8054 | 0.7149 | 0.7943 | 0.3910 | 0.4712 | 0.6332 | 0.7940 | 0.6302 | 0.7162 |
| 2D OTSU | 0.8763 | 0.7974 | 0.8555 | 0.7930 | 0.5405 | 0.4327 | 0.6748 | 0.7632 | 0.7368 | 0.6966 |
| MOTSU_2D | 0.8603 | 0.7847 | 0.9510 | 0.8225 | 0.6950 | 0.3837 | 0.7770 | 0.6623 | 0.8208 | 0.6633 |
| DBOTSU_2D | 0.8518 | 0.8535 | 0.9205 | 0.9084 | 0.6406 | 0.5455 | 0.7525 | 0.7897 | 0.7913 | 0.7743 |
| EOTSU_3D | 0.8468 | 0.8551 | 0.9176 | 0.9153 | 0.6279 | 0.5861 | 0.7406 | 0.8030 | 0.7832 | 0.7899 |
| ENTROPY | 0.7821 | 0.7934 | 0.7425 | 0.7919 | 0.4261 | 0.3888 | 0.6382 | 0.7524 | 0.6472 | 0.6816 |
| 2D ENTROPY | 0.8184 | 0.8649 | 0.8701 | 0.9111 | 0.5464 | 0.6781 | 0.5894 | 0.5133 | 0.7061 | 0.7418 |
| K-MEANS | 0.7822 | 0.7934 | 0.7407 | 0.7919 | 0.4238 | 0.3899 | 0.6375 | 0.7537 | 0.6460 | 0.6822 |
| FGFCM | 0.9271 | 0.9167 | 0.9647 | 0.9414 | 0.7200 | 0.6737 | 0.8433 | 0.8433 | 0.8638 | 0.8438 |
| FLICM | 0.9384 | 0.9411 | 0.9756 | 0.9766 | 0.7078 | 0.7040 | 0.8372 | 0.8514 | 0.8647 | 0.8683 |
| MAOTSU_2D | 0.9404 | 0.9428 | 0.9744 | 0.9777 | 0.7835 | 0.6634 | 0.8625 | 0.8768 | 0.8902 | 0.8652 |

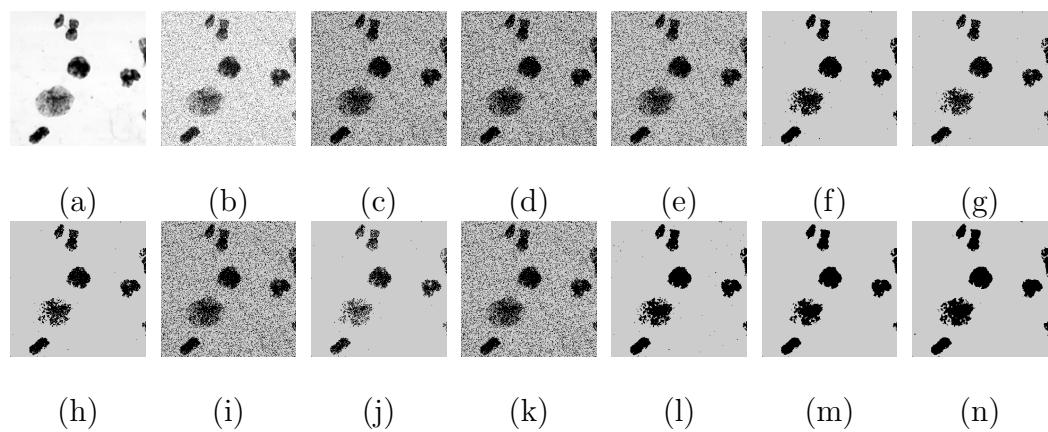


Figure 15: Comparison of segmentation results on a real image corrupted by Gaussian noise ($\sigma^2 = 0.1$). (a) The original image, (b) the same image corrupted by Gaussian noise ($\sigma^2 = 0.1$), (c) using OTSU, (d) using ITOTSU, (e) using 2D OTSU, (f) using MOTSU_2D, (g) using DBOTSU_2D, (h) using EOTSU_3D, (i) using ENTROPY, (j) using 2D ENTROPY, (k) using K-MEANS, (l) using FGFCM, (m) using FLICM, (n) using MAOTSU_2D.

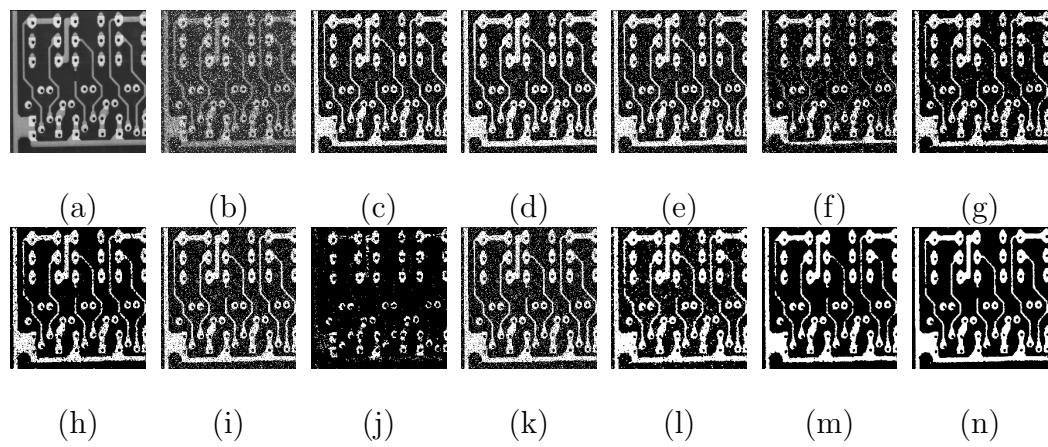


Figure 16: Comparison of segmentation results on a real image corrupted by Salt&Pepper noise ($\delta = 0.2$). (a) The original image, (b) the same image corrupted by Salt&Pepper noise ($\delta = 0.2$), (c) using OTSU, (d) using ITOTSU, (e) using 2D OTSU, (f) using MOTSU_2D, (g) using DBOTSU_2D, (h) using EOTSU_3D, (i) using ENTROPY, (j) using 2D ENTROPY, (k) using K-MEANS, (l) using FGFCM, (m) using FLICM, (n) using MAOTSU_2D.

1
2
3
4
5 *5.3. Comparison of Computation Time*
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

Finally we compare the computation efficiency of the 12 methods. The experiments are conducted on one hundred images with different sizes from 100×100 to 1000×1000 . Each image size has been tested with ten different images and each running time is the average from ten experiments. The average running time comparison is illustrated in Figure 17. From Figure 17 we observe that FLICM is the least efficient method. When the image size is less than or equal to 500×500 , DBOTSU_2D, EOTSU_3D and MAOTSU_2D are evidently faster than MOTSU_2D, 2D OTSU and 2D ENTROPY, and this advantage in computation efficiency is attributed to the search of optimal thresholds in 1D space. While OTSU is the most efficient method, it also generates the worst segmentation results. Although calculating the extra neighborhood median and conducting region post-processing add to the computation load, the proposed MAOTSU_2D still runs faster than the fast recursive algorithm of the 2D maximum entropy method (2D ENTROPY), ITOTSU and K-MEANS and close to ENTROPY, EOTSU_3D and FGFCM. Furthermore, the average computation time of the proposed MAOTSU_2D is only 0.37 seconds even for the image size of 1000×1000 and no more than 0.21% of FLICM's. This means that our MAOTSU_2D has an absolute advantage over FLICM in computation efficiency.

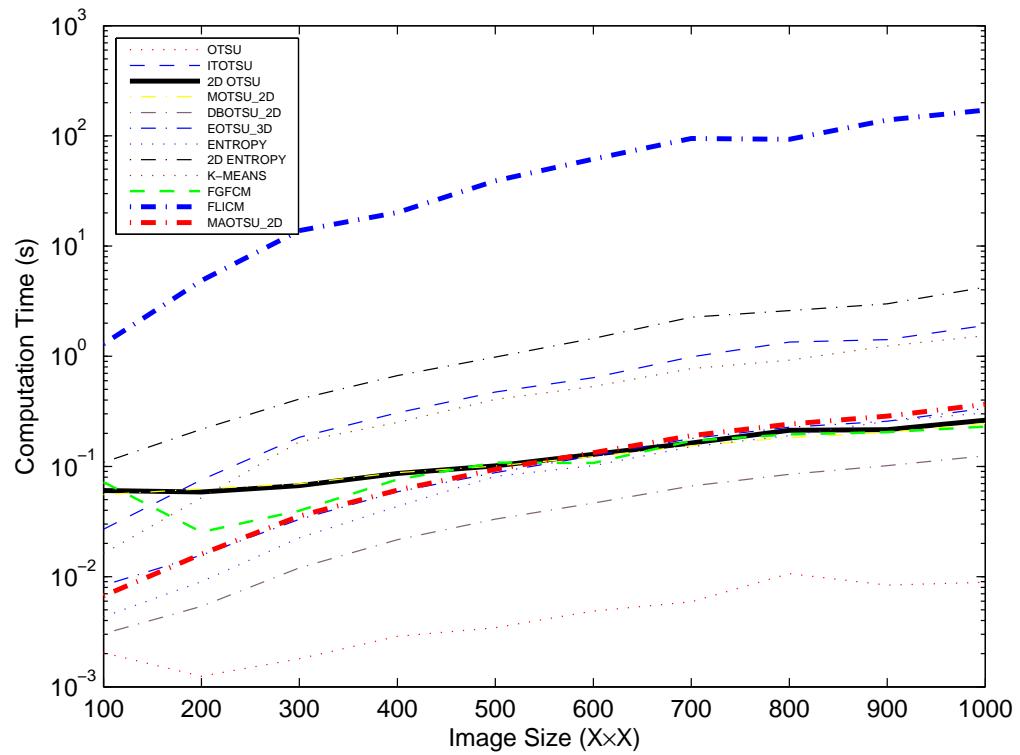


Figure 17: Comparison of the computation time.

1
2
3
4
5 **6. Conclusion**
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

In this paper we present a robust 2D Otsu's thresholding method called MAOTSU_2D for image segmentation. This method builds the 2D histogram based on the image smoothed by both median and average filters, in contrast to the traditional methods where the images are smoothed by average filter only. We select the optimal threshold vector by two 1D searches on the two dimensions separately, different from the 2D search in traditional methods. Finally, MAOTSU_2D introduces a region post-processing step to remove the noisy points in segmented images. We compare our method to 11 other methods in experiments on both synthetic and real images with Salt&Pepper noise and Gaussian noise. Both qualitative and quantitative comparisons indicate that our method has the best or second best robustness to Salt&Pepper noise and Gaussian noise. Furthermore, our method, ranking as one of the most efficient methods, is dramatically faster than FLICM and more efficient than or close to the fast recursive algorithm for 2D Otsu (2D OTSU) and MOTSU_2D.

41 **Acknowledgement**
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

This work is supported in part by the National Natural Science Foundation of China under Grant No. 61473045 and No. 41371425, and by the Program for Liaoning Innovative Research Team in University (LT2013023).

References

- Ahmed, M., Yamany, S., Mohamed, N., Farag, A., Moriarty, T., 2002. A modified fuzzy c-means algorithm for bias field estimation and segmentation of mri data. *Medical Imaging, IEEE Transactions on* 21, 193–199.
- Alsaeed, D.H., Bouridane, A., Elzaart, A., Sammouda, R., 2012. Two modified otsu image segmentation methods based on lognormal and gamma distribution models, in: *Information Technology and e-Services (ICITEs)*, 2012 International Conference on, IEEE. pp. 1–5.
- Cai, H., Yang, Z., Cao, X., Xia, W., Xu, X., 2014. A new iterative triclass thresholding technique in image segmentation. *Image Processing, IEEE Transactions on* 23, 1038–1046.
- Cai, W., Chen, S., Zhang, D., 2007. Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation. *Pattern Recognition* 40, 825–838.
- Chen, Q., Zhao, L., Lu, J., Kuang, G., Wang, N., Jiang, Y., 2012. Modified two-dimensional otsu image segmentation algorithm and fast realisation. *Image Processing, IET* 6, 426–433.
- Chen, Y., Chen, D., Yang, X., Chen, L., 2010. Otsu's thresholding method based on gray level-gradient two-dimensional histogram, in: *Informatics in Control, Automation and Robotics (CAR)*, 2010 2nd International Asia Conference on, IEEE. pp. 282–285.

- Cheriet, M., Said, J.N., Suen, C.Y., 1998. A recursive thresholding technique for image segmentation. *Image Processing, IEEE Transactions on* 7, 918–921.
- Fan, J., Zhao, F., 2007. Two-dimensional otsu's curve thresholding segmentation method for gray-level images. *Dianzi Xuebao(Acta Electronica Sinica)* 35, 751–755.
- Fan, J., Zhao, F., Zhang, X., 2007. Recursive algorithm for three-dimensional otsu's thresholding segmentation method. *Dianzi Xuebao(Acta Electronica Sinica)* 35, 1398–1402.
- Gong, J., Li, L., C, W., 1998. Fast recursive algorithms for two-dimensional thresholding. *Pattern Recognition* 31, 295–300.
- Guo, W., Wang, X., Xia, X., 2014. Two-dimensional otsu's thresholding segmentation method based on grid box filter. *Optik-International Journal for Light and Electron Optics* 125, 5234–5240.
- Hao, Y., Zhu, F., 2005. Fast algorithm for two-dimensional otsu adaptive threshold algorithm. *Journal of Image and Graphics* 4, 014.
- Hou, J., Sha, C., Chi, L., Xia, Q., Qi, N., 2014. Merging dominant sets and dbscan for robust clustering and image segmentation, in: *Proceedings of The 21st IEEE International Conference on Image Processing (ICIP 2014)*, pp. 4422–4426.

- 1
2
3
4 Hwang, H., Haddad, R., 1995. Adaptive median filters: new algorithms and
5 results. *Image Processing, IEEE Transactions on* 4, 499–502.
6
7 Jing, X., Li, J., Liu, Y., 2003. Image segmentation based on 3-d maximum
8 between-cluster variance. *Acta Electronica Sinica* 31, 1281–1285.
9
10 Kittler, J., Illingworth, J., 1986. Minimum error thresholding. *Pattern Recog-*
11
12 nition
- 13 19, 41–47.
14
15 Krinidis, S., Chatzis, V., 2010. A robust fuzzy local information c-means
16 clustering algorithm. *Image Processing, IEEE Transactions on* 19, 1328–
17 1337.
18
19 Lai, Y.K., Pl., R., 2014. Efficient circular thresholding. *Image Processing,*
20
21 *IEEE Transactions on* 23, 992–1001.
22
23
24 Lang, X., Zhu, F., Hao, Y., Ou, J., 2008. Integral image based fast algorithm
25 for two-dimensional otsu thresholding, in: *Image and Signal Processing,*
26
27 2008. CISP '08. Congress on, pp. 677–681.
28
29
30 Liu, J., Li, W., Tian, Y., 1991. Automatic thresholding of gray-level pictures
31 using two-dimension otsu method, in: *Circuits and Systems, 1991. Con-*
32
33 *ference Proceedings, China., 1991 International Conference on, IEEE.* pp.
34
35 325–327.
36
37
38 Macqueen, J., 1967. Some methods for classification and analysis of multi-
39
40 variate observations, in: *Proceedings of the Fifth Berkeley Symposium on*
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

1
2
3
4 Mathematical Statistics and Probability, Volume 1: Statistics, pp. 281–
5
6 297.
7
8

9
10 Masulli, F., Schenone, A., 1999. A fuzzy clustering based segmentation sys-
11 tem as support to diagnosis in medical imaging. Artificial Intelligence in
12 Medicine 16, 129–147.
13
14

15
16 Moghaddam, R., Cheriet, M., 2012. Adotsu: An adaptive and parameterless
17 generalization of otsu’s method for document image binarization. Pattern
18 Recognition 45, 2419–2431.
19
20

21
22 Otsu, N., 1979. A threshold selection method from gray-level histograms.
23
24 IEEE Transactions on Systems, Man and Cybernetics 9, 62–66.
25
26

27
28 Pal, N.R., Pal, S.K., 1989. Object-background segmentation using new def-
29 initions of entropy. Computers & Digital Techniques Iee Proceedings E
30
31 136, 284–295.
32
33

34
35 Pun, T., 1980. A new method for grey-level picture thresholding using the
36 entropy of the histogram. Signal Processing 2, 223–237.
37
38

39 Sezgin, M., Sankur, B., 2004. Survey over image thresholding techniques and
40 quantitative performance evaluation. J. Electronic Imaging 13, 146–168.
41
42

43 Sirisha, P., Naga Raju, C., Pradeep Kumar Reddy, R., 2013. An efficient
44 fuzzy technique for detection of brain tumor. Journal on Software Engi-
45 neering 7.
46
47

- 1
2
3
4
5 Sthitpattanapongsa, P., Srinark, T., 2012. An equivalent 3d otsus threshold-
6
7
8
9
10
11
12 Wang, H., Pan, D., Xia, D., 2007. A fast algorithm for two-dimensional otsu
13
14 adaptive threshold algorithm. *Acta Automatica Sinica* 33, 968–971.
15
16
17 Wang, L., Duan, H., Wang, J., 2008. A fast algorithm for three-dimensional
18
19 otsu's thresholding method, in: IT in Medicine and Education, 2008. ITME
20
21 2008. IEEE International Symposium on, pp. 136–140.
22
23
24 Wang, N., Li, X., Chen, X., 2010. Fast three-dimensional otsu thresholding
25
26 with shuffled frog-leaping algorithm. *Pattern Recognition Letters* 31, 1809–
27
28 1815.
29
30
31 Wang, Q., Wan, S., Yue, L., 2011. A novel robust algorithm for image
32
33 segmentation, in: Image and Graphics (ICIG), 2011 Sixth International
34
35 Conference on, pp. 238–243.
36
37
38 Wei, K., Zhang, T., Shen, X., Liu, J., 2007. An improved threshold selection
39
40 algorithm based on particle swarm optimization for image segmentation,
41
42 in: Natural Computation, 2007. ICNC 2007. Third International Confer-
43
44 ence on, pp. 591–594.
45
46
47 Xiong, B., Yin, Z., 2012. A universal denoising framework with a new impulse
48
49 detector and nonlocal means. *Image Processing, IEEE Transactions on* 21,
50
51 1663–1675.
52
53
54
55
56
57
58
59
60
61
62
63
64
65

- 1
2
3
4 Xu, X., Xu, S., Jin, L., Song, E., 2011. Characteristic analysis of otsu
5 threshold and its applications. Pattern Recognition Letters 32, 956–961.
6
7
8 Xue, J., Titterington, D., 2011. t-tests, f-tests and otsu's methods for image
9 thresholding. Image Processing, IEEE Transactions on 20.
10
11
12 Yasnoff, W.A., Mui, J.K., Bacus, J.W., 1977. Error measures for scene
13 segmentation. Pattern Recognition 9, 217–231.
14
15
16
17 Yue, F., Zuo, W., Wang, K., 2009. Decomposition based two-dimensional
18 threshold algorithm for gray images. Zidonghua Xuebao/Acta Automatica
19 Sinica 35, 1022–1027.
20
21
22
23
24
25
26
27 Zhang, D., Chen, S., 2004. A novel kernelized fuzzy c-means algorithm
28 with application in medical image segmentation. Artificial Intelligence in
29 Medicine 32, 37–50.
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54