

## A Practical Approach of Selecting the Edge Detector Parameters to Achieve a Good Edge Map of the Gray Image

<sup>1</sup>Akram A. Moustafa and <sup>2</sup>Ziad A. Alqadi

<sup>1</sup>Department of Computer Science, Al Al-Bayt University, P.O. Box 922283, Mafrag, Jordan 11192

<sup>2</sup>Faculty of Engineering, Al-Balqa Applied University, Amman, Jordan

**Abstract: Problem statement:** practical approach of detecting edge map was proposed. **Approach:** The methodology of this approach was presented and tested in order to select the best value of the operator, used to smooth and get the gradient, and the threshold value used to convert the gray gradient to binary edge map, so a practical value of the threshold and edge operator coefficient was investigated, these values used to calculate the gradient in order to get a better edge-map. **Results:** While increasing the value of C (constant C the first parameter in our practical approach of detecting the edge), we narrow the range of t and at the same time the value of the suitable t will be increased toward 1. **Conclusion:** This approach can be used to get the best edge map and to get a clear edge map, which can be used later in image segmentation and object extraction.

**Keywords:** Gradient, detecting, edge detection methodology, original image, edge direction, pixels

### INTRODUCTION

As we a gray image is a set of pixels which are arranged in a 2D matrix, the intersection of a certain row and column defines the x and y coordinates of a certain pixel and the corresponding value of the pixel reflects the intensity value of the point which ranges from the minimum gray level (0) to the maximum gray level (for example 255)<sup>[1]</sup>.

Edge detection is point processing method used to segment the image and to separate the image into different discontinuous objects which can be used for farther image processing operation<sup>[2]</sup>.

Edge detection techniques are based<sup>[3-7]</sup> on the Properties of the gradient which includes:

- The magnitude of gradient provides information about the strength of the edge.
- The direction of gradient is always perpendicular to the direction of the edge as shown in Fig. 1

So the main idea of getting the gradient at each point is to:

- Compute derivatives in x and y directions
- Find gradient magnitude
- Threshold gradient magnitude

Thus we can estimate the gradient with finite differences:

$$\frac{\partial f}{\partial x} = \lim_{h \rightarrow 0} \frac{f(x+h, y) - f(x, y)}{h} \quad (1)$$

$$\frac{\partial f}{\partial y} = \lim_{h \rightarrow 0} \frac{f(x, y+h) - f(x, y)}{h} \quad (2)$$

Setting h to 1 we can obtain:

$$\begin{aligned} \frac{\partial f}{\partial x} &= \frac{f(x+h_x, y) - f(x, y)}{h_x} \\ &= f(x+1, y) - f(x, y), (h_x = 1) \end{aligned} \quad (3)$$

$$\begin{aligned} \frac{\partial f}{\partial y} &= \frac{f(x, y+h_y) - f(x, y)}{h_y} \\ &= f(x, y+1) - f(x, y), (h_y = 1) \end{aligned} \quad (4)$$

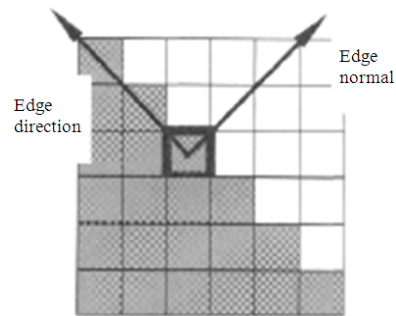


Fig. 1: Point gradient

**Corresponding Author:** Akram A. Moustafa, Department of Computer Science, Al Al-Bayt University, P.O. Box 922283, Mafrag, Jordan 11192

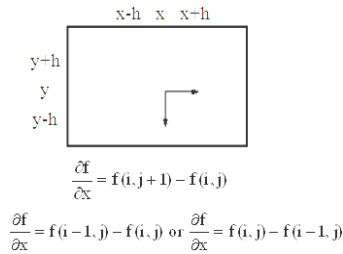


Fig. 2: Pixel coordinates

Using pixel-coordinate notation (j corresponds to the x direction and i to the negative y direction) as shown in Fig. 2.

Let us now Consider the arrangement of pixels about the pixel (i, j) as follows:

$$\begin{bmatrix} a_0 & a_1 & a_2 \\ a_7 & a[i, j] & a_3 \\ a_6 & a_5 & a_4 \end{bmatrix}$$

The partial derivatives can be computed by:

$$M_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6) \quad (5)$$

$$M_y = (a_6 + ca_5 + a_4) - (a_0 + ca_1 + a_2) \quad (6)$$

We call the constant C the first parameter in our practical approach of detecting the edge.

By setting C to 1 we can obtain the Prewitt operator<sup>[3-7]</sup>:

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

$$M_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

And by setting C to 2 we can get the Sobel operator<sup>[3-7]</sup>:

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$M_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

## MATERIALS AND METHODS

We proposed the following steps in order to get the edge map of the original gray image:

- Get the original gray image
- Select X and Y gradient operators by setting C to a defined number
- Smooth the gray image and get the smoothed X and Y gradients by performing the following 2 steps:
  - Correlate the original image with X operator to get the X gradient
  - Correlate the original image with Y operator to get the Y gradient
- Compute the total gradient
- Convert the total gray gradient to binary in order to get the edge map by threshold the gray gradient using a threshold value t (t is a second operator in our practical approach of detecting the edge map)
- Let us illustrates these steps by example using mat lab
- Get the original image:  
a = imread('rice.png')  
In Fig. 3, imshow(a); title 'Original image'
- Select the gradient operators(let C = 1):
 

```
x = [-1 0 1; -1 0 1; -1 0 1]
x = -1 0 1
     -1 0 1
     -1 0 1
>> y = x'
y = -1 -1 -1
     0 0 0
     1 1 1
```
- Get a smoothed X gradient by correlating the original image with X operator:  
Gx = filter2(x, a)
- Get a smoothed Y gradient by correlating the original image with Y operator:  
Gy = filter2(y, a)
- Compute the total gray gradient:  
G = sqrt(Gx.\*Gx+Gy.\*Gy)
- Select the threshold value and convert the gray gradient to binary in order to get the edge map:  
t = 0.3  
e\_map = im2bw(G/255,t)
- In Fig. 4, imshow(e\_map), title 'Edge Map'.

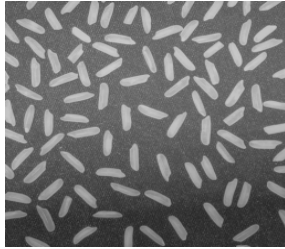


Fig. 3: Original image



Fig. 4: Edge Map:

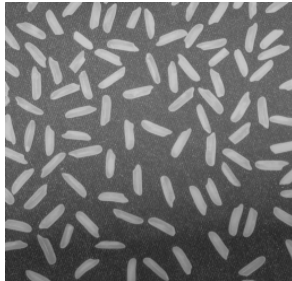


Fig. 5: Original gray image

### RESULTS

Now we will try to use various values of the operator C and for each value of C we will select different values of t, then we apply the methodology of detecting the edge map. The obtained results will be analyzed in order to get the exact rang of t values which can be used as a best results to get the best clear edge map.

Here are some practical results (Fig. 5-45).

C = 1(Prewitt detector)

```
gx = -1 -1 -1
      0  0  0
      1  1  1
```

```
>> gy = gx'
gy   = -1  0  1
      -1  0  1
      -1  0  1
```

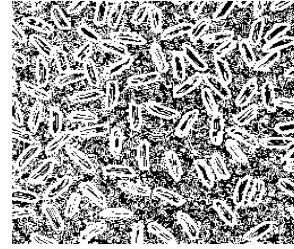


Fig. 6: Values of t = 0.1

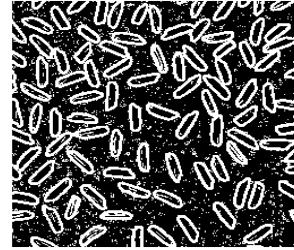


Fig. 7: Values of t = 0.2

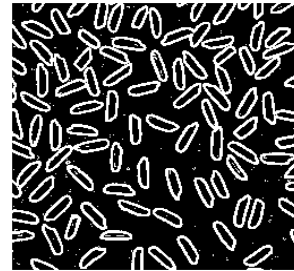


Fig. 8: Values of t = 0.3



Fig. 9: Values of t = 0.4



Fig. 10 Values of t = 0.5

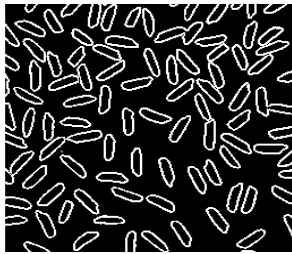


Fig. 11: Values of  $t = 0.6$



Fig. 12: Values of  $t = 0.7$



Fig. 13: Values of  $t = 0.8$

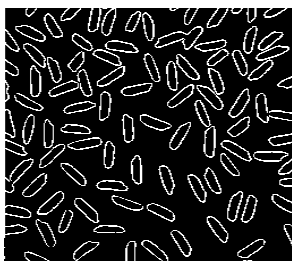


Fig. 14: Values of  $t = 0.9$

Good for  $0.4 \leq t \leq 0.9$   
 $C = 2$  (Sobel detector)  
 $gx = \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}$   
 $gy = \begin{matrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{matrix}$



Fig. 15: Values of  $t = 1$

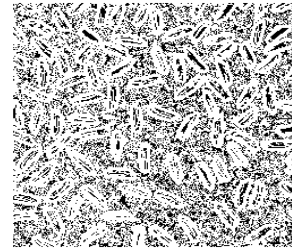


Fig. 16: Edge with  $t = 0.1$

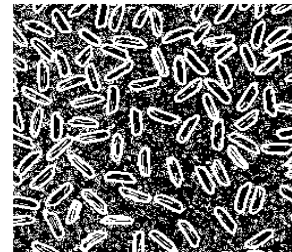


Fig. 17: Edge with of  $t = 0.2$



Fig. 18: Edge with  $t = 0.3$

Good for  $0.4 \leq t \leq 0.8$

$C = 3$   
 $gxz = \begin{matrix} -1 & -3 & -1 \\ 0 & 0 & 0 \\ 1 & 3 & 1 \end{matrix}$

$\gg gyz = gxz'$   
 $gyz = \begin{matrix} -1 & 0 & 1 \\ -3 & 0 & 3 \\ -1 & 0 & 1 \end{matrix}$

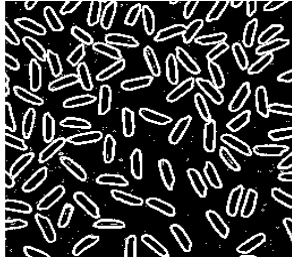


Fig. 19: Edge with of t = 0.4



Fig. 20: Edge with t = 0.5



Fig. 21: Edge with of t = 0.6



Fig. 22: Edge with t = 0.7

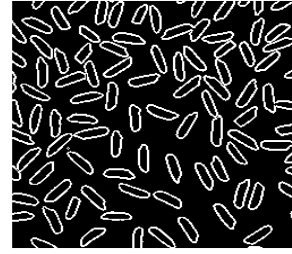


Fig. 23: Edge with of t = 0.8

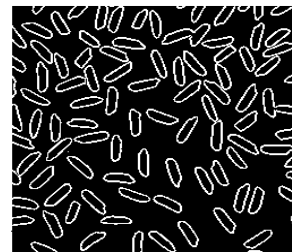


Fig. 24: Edge with t = 0.9

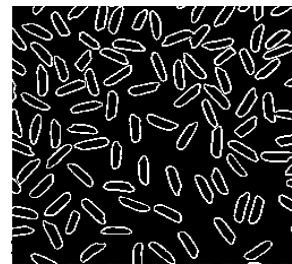


Fig. 25: Edge with of t = 1

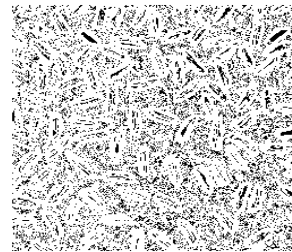


Fig. 26: Values of t = 0.1

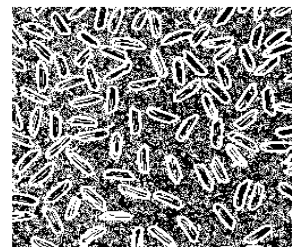


Fig. 27: Values of t = 0.2

Good for  $0.6 \leq t \leq 0.9$

C = 4

```
gx = -1  0  1
     -4  0  4
       -1  0  1
```

>> gy = gx'

```
gy  = -1  -4  -1
      0   0   0
      1   4   1
```

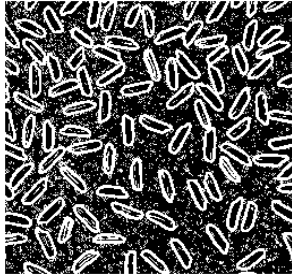


Fig. 28: Values of  $t = 0.3$

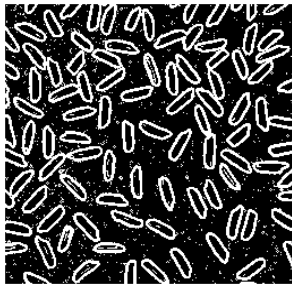


Fig. 29: Values of  $t = 0.4$



Fig. 30: Values of  $t = 0.5$

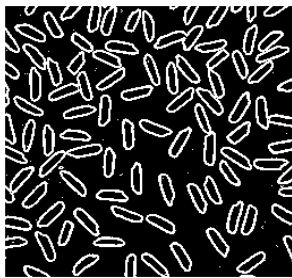


Fig. 31: Values of  $t = 0.6$

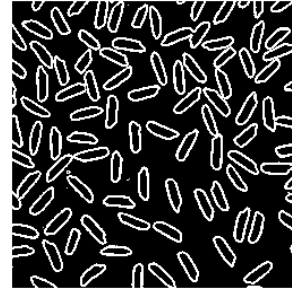


Fig. 32: Values of  $t = 0.7$

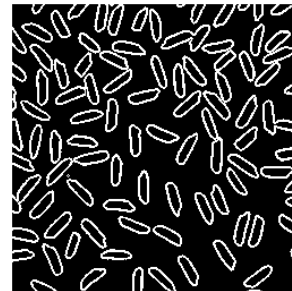


Fig. 33: Values of  $t = 0.8$



Fig. 34: Values of  $t = 0.9$



Fig. 35: Values of  $t = 1$

Good for  $0.8 \leq t \leq 1$   
C = 5  
gx = -1 0 1  
-5 0 5  
-1 0 1

>> gy = gx'  
gy = -1 -5 -1  
0 0 0  
1 5 1  
Good for  $0.9 \leq t \leq 1$

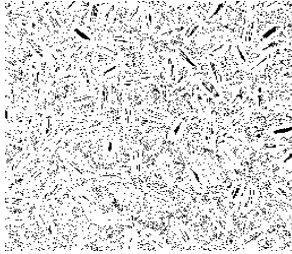


Fig. 36: Values of  $t = 0.1$

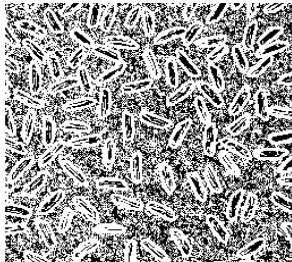


Fig. 37 Values of  $t = 0.2$

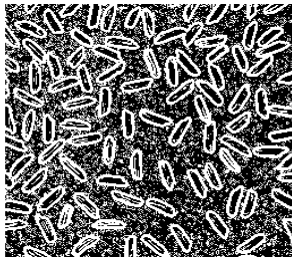


Fig. 38: Values of  $t = 0.3$

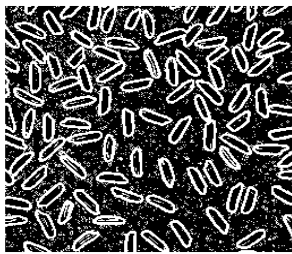


Fig. 39: Values of  $t = 0.4$

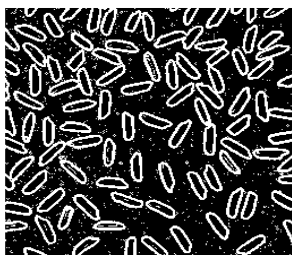


Fig. 40: Values of  $t = 0.5$

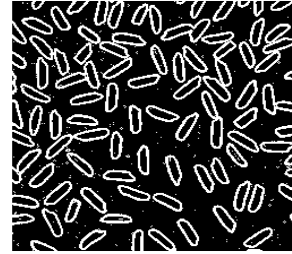


Fig. 41: Values of  $t = 0.6$

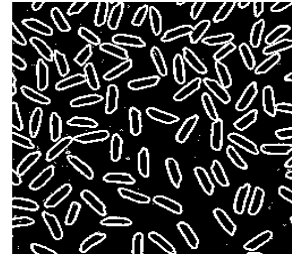


Fig. 42: Values of  $t = 0.7$

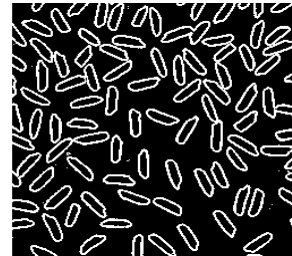


Fig. 43: values of  $t = 0.8$



Fig. 44: Values of  $t = 0.9$



Fig. 45: Values of  $t = 1$

### **DISCUSSION**

As we see from the results while increasing the value of C we narrow the range of t and at the same time the value of the suitable t will be increased toward 1.

### **CONCLUSION**

In order to get a clear edge map we have to select the C parameter which is to be used as a constant to define the X and Y operators to get the X and Y gradients. Increasing of C value leads to narrow the range of t, thus we have to select a high value of t which is closed to 1.

### **REFERENCES**

1. Canny, J., 1986. A computational approach to edge detection. *IEEE Trans. Patt. Anal. Mach. Intell.*, 8: 679-714.  
<http://portal.acm.org/citation.cfm?id=11275>
2. Deriche, R., 1987. Using Canny's criteria to derive an optimal edge detector recursively implemented. *Int. J. Comput. Vis.*, 1: 167-187. DOI: 10.1007/BF00123164
3. Lindeberg, T., 1998. Edge detection and ridge detection with automatic scale selection *Int. J. Comput. Vis.*, 30: 117-154.  
<http://portal.acm.org/citation.cfm?id=305297.305299>
4. Lindeberg, T., 1997. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, USA., ISBN: 0-7923-9418-6, pp: 435.
5. Pathegama, M. and Ö. Göl, 2005. Edge-end pixel extraction for edge-based image segmentation. *Proceedings of the World Academy of Science, Engineering and Technology*, Jan. 2005, pp: 161-163.  
<http://www.waset.org/pwaset/v2/v2-41.pdf>
6. Zhang, W. and F. Bergholm, 1997. Multi-scale blur estimation and edge type classification for scene analysis. *Int. J. Comput. Vis.*, 24: 219-250.  
<http://portal.acm.org/citation.cfm?id=264981>
7. Ziou, D. and S. Tabbone, 1998. Edge detection techniques an overview. *Int. J. Patt. Recog. Image Anal.*, 8: 537-559.  
<http://en.scientificcommons.org/85470>