Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.017

IJCSMC, Vol. 5, Issue. 11, November 2016, pg.29 – 36

Extracting Isolated Words from an Image of Text

Bassam M.Subaih[1], Prof. Ziad A.Alqadi[2], Mazen A.Hamdan[3]

1, 2, 3 Albalqa Applied University Jordan, Faculty of Engineering Technology

Department of Computer Engineering

Abstract: Edge detection is a process of locating an edge of an image. Detection of edges in an image is a very important step towards understanding image features. This paper will proposed a special edge operator which can be implemented to find, detect and extract isolated words in a text image. The proposed method will be implemented using different types of images and different images which contain isolated horizontal, vertical, and both horizontal and vertical words. The proposed method will be compared with other edge operators in order to check the efficiency of the proposed method.

Key words: Edge, Edge operator, Color image, gray image, Binary image.

I- Introduction

Edge detection is a process of locating an edge of an image. Detection of edges in an image is a very important step towards understanding image features. Edges consist of meaningful features and contain significant information. It significantly reduces the image size and filters out information that may be regarded as less relevant, thus preserving the important structural properties of an image.[4],[5] Most images contain some amount of redundancies that can sometimes be removed when edges are detected and replaced during reconstruction. This is where edge detection comes into play. [6],[7]Also, edge detection is one of the ways of making images not take up too much space in the computer memory. Since edges often occur at image locations representing object boundaries, edge detection is extensively used in image segmentation when images are divided into areas corresponding to deferent objects.

A recent development in edge detection techniques takes a frequency domain approach to finding edge locations. Phase congruency methods attempt to find locations in an image where all sinusoids in the frequency domain are in phase. These locations will generally correspond to the location of a perceived edge, regardless of whether the edge is represented by a large change in intensity in the spatial domain. A key benefit of this technique is that it responds strongly to Mach bands, and avoids false positives typically found around roof edges. A roof edge, is a discontinuity in the first order derivative of a grey-level profile.[10],[11],and [12].

I-1 Edge definition

Edges is defined as a sudden changes of discontinuities in an image or a significant transitions in an image [1],[2]. Generally edges are of three types:

- Horizontal edges
- Vertical Edges
- Diagonal Edges

Most of the shape information of an image is enclosed in edges. So first we detect these edges in an image and by using these filters and then by enhancing those areas of image which contains edges, sharpness of the image will increase and image will become clearer. Here are some of the masks (operators)for edge detection that we will discuss in this section: Prewitt Operator, Sobel Operator, Laplacian Operator, Canny operator, and Roberts operator, each of these operators is used to obtain the gradient magnitude by applying convolution operation using the operator and the input image. Table 1 shows the value for different edge operator.[1], [2], and [3].

I-2 Digital image

Digital image can be color, gray or binary image. An RGB image, sometimes referred to as a *true-color* image,[9] is stored as an *m*-by-*n*-by-3 data array that defines red, green, and blue color components for each individual pixel. RGB images do not use a palette. The color of each pixel is determined by the combination of the red, green, and blue intensities stored in each color plane at the pixel's location. Graphics file formats store RGB images as 24-bit images, where the red, green, and blue components are 8 bits each. Grayscale image It is also known as an intensity, gray scale, or gray level image. Array of class uint8, uint16, int16, single, or double whose pixel values specify intensity values. For single or double arrays, values range from [0, 1]. For uint8, values range from [0,255]. For uint16, values range from [0, 65535]. For int16, values range from [-32768, 32767]. Binary image is a Logical array containing only 0s and 1s, interpreted as black and white, respectively.

Table 1:Edge operators values

Operator	Operator value(mask)		
Laplace[4], [5]	0	1	0
	1	-4	1
	0	1	0
Canny[2],[3]	-2	-2	0
	-2	0	2
	0	2	2
Prewitt[5],[6]	-2	-1	0
	-1	0	1
	0	1	2
Sobel[4]	0	2	2
	-2	0	2
	-2	-2	0
Roberts	1	1	
	-1	-1	

Different methods are now available to convert RGB image to gray image, and gray image to binary image[9] and this operation is important specially if there is a need to find the image edges.[9].

II- The proposed method of extracting words from a text image

The proposed method of extracting isolated words from an image of text is based on edge detection (calculation) . Edge detection requires an operator(Mask) [1],[4],[5] , which can be used to find the gradient that points to the local extremes by applying convolution between the mask and the input image[6],[7],[8]. The proposed method of extracting word from the text image is based on selecting the following edge operator :

1 0 1

0 -4 0

1 0 1

Which means that each pixel p(x,y) in the edge is to be calculated using the following formula:

New p(x,y)=p(x-1,y-1)+p(x-1,y+1)+p(x+1,y-1)-4p(x,y)

Using this proposed operator the method of extracting isolated words from text image can be implemented applying the following sequences of steps:

1. Get the original text image (imagen).
2. If imagen is color image convert it to gray.
3. Convert the resulting image in 2 to binary image.
4. Filter the image by removing small objects from the binary image.
5. Generate the proposed operator.
6. Apply convolution of the operator and filtered binary image.
7. Apply bwlabel to get the connected objects.
8. Retrieve the objects which represent the isolated words in input text image.

III- Method implementation

The following matlab code was written and executed several times using the proposed operator and other operators using different text images:

```
%% Read Image
imagen=imread('C:\Users\User\Desktop\im6.png');
%% Show image
figure(1)
imshow(imagen);
title('INPUT IMAGE WITH NOISE')
%% Convert to gray scale
if size(imagen,3)==3 % RGB image
  imagen=rgb2gray(imagen);
end
%% Convert to binary image
threshold = graythresh(imagen);
imagen =~im2bw(imagen,threshold);
%% Remove all object containing fewer than 30 pixels

imagen = bwareaopen(imagen,30);
pause(1)
%% Show image binary image
figure(2)
imshow(~imagen);
title('INPUT IMAGE WITHOUT NOISE')
BW=imagen;figure
  imshow(~BW)


g1=[1 0 1;0 -4 0;1 0 1];
gg1=filter2(g1,BW);
[n1 n2]=bwlabel(gg1,4);

for n=1:n2
  ob = [n1==n];
  imshow(~ob);
  pause(0.5)
end
n2
```


III-1 Using Binary text image with horizontal and vertical words

The following binary text image shown in figure 1 was treated as an input image for the proposed method.

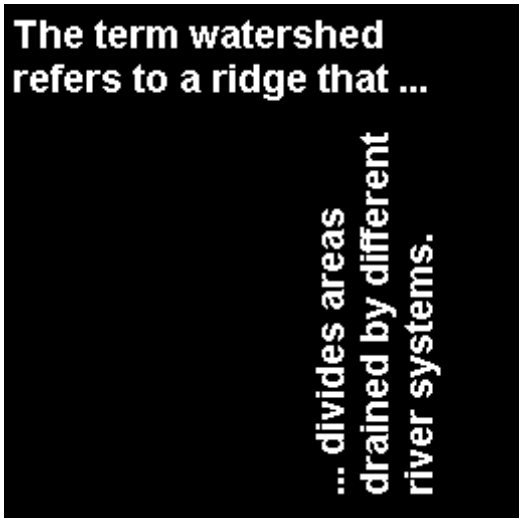

Figure 1: Binary input image(17 words)

Using the proposed operator figure 2 shows some of the outputs:

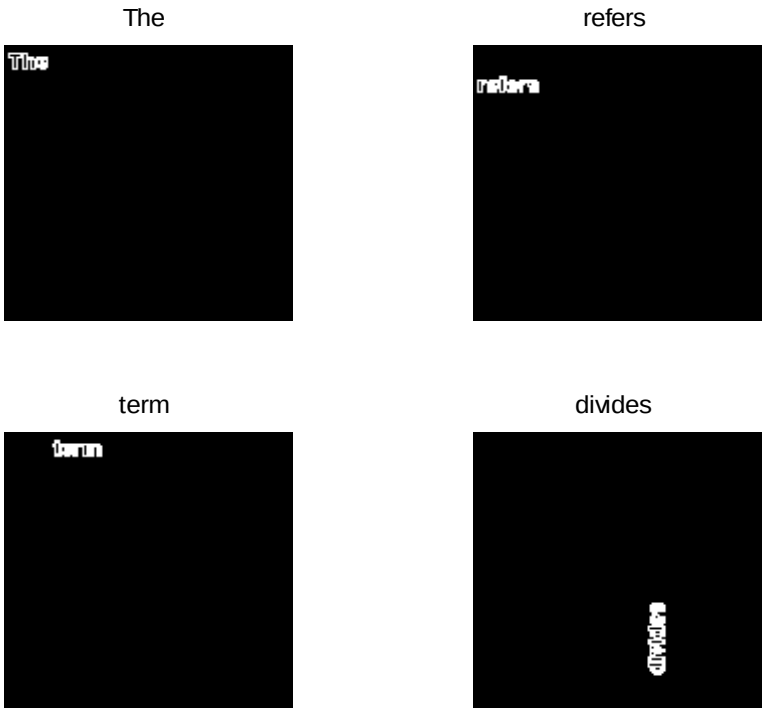

Figure 2: Some output words using proposed operator

Table 2 shows a comparisons results by applying other edge operators.

Table 2: Results using binary image with 17 horizontal and vertical words

Operator	# of objects(words) retrieved	Remarks
Proposed	17	Excellent for horizontal and vertical words
Laplace	18	Good for horizontal words
Canny	94	Poor
Prewitt	34	Poor
Sobel	82	Poor
Roberts	360	Poor

III-2 Using color text image with horizontal words

The same method was executed using a color text image as an input image, which is shown in figure 3.

Figure 3: Color text input image(9 horizontal words)

The proposed method was implemented using the proposed operator and other operators and the execution results are shown in table 3.

Table 3: Results using color image with 9 horizontal words

Operator	# of objects(words) retrieved	Remarks
Proposed	9	Excellent for vertical words
Laplace	10	Good for vertical words
Canny	24	Poor
Prewitt	22	Poor
Sobel	19	Poor
Roberts	213	Poor

III-3 Using color text image with vertical words

The same method was executed using a color text image as an input image, which is shown in figure 4

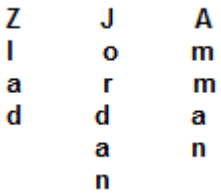

Figure 4: Color text input image(3 vertical words)

The proposed method was implemented using the proposed operator and other operators and the execution results are shown in table 4.

Table 4: Results using color image with 3 vertical words

Operator	# of objects(words) retrieved	Remarks
Proposed	3	Excellent for vertical words
Laplace	15	Poor
Canny	16	Poor
Prewitt	16	Poor
Sobel	18	Poor
Roberts	47	Poor

From the experimental results shown in tables 2, 3, and 4 we can conclude the following:

- The proposed operator is efficient to detect and extract words in all types of images.
- The proposed operator efficiently detect horizontal words in the text image.
- The proposed operator efficiently detect vertical words in the text image.
- The proposed operator efficiently detect both horizontal and vertical words in the text image.
- Laplacian operator is good to detect only horizontal words.
- All other operators are not efficient in detecting words and give poor results.

IV-Conclusions

A new method of detecting and extracting isolated words in a text image which used a special edge operator was proposed, implemented and tested. It was shown that the proposed method is very efficient in finding and extracting vertical and horizontal words in any type of digital image.

References

[1] Akram A. Moustafa and Ziad A. Alqadi, A Practical Approach of Selecting the Edge Detector Parameters to Achieve a Good Edge Map of the Gray Image, Journal of Computer Science 5 (5): 355-362, 2009 ISSN 1549-3636.
[2] 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,
[3] 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
[4] 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=305297.3052 99
[5] Lindeberg, T., 1997. Scale-Space Theory in Computer Vision. Kluwer Academic Publishers, USA., ISBN: 0-7923-9418-6, pp: 435.

[6] 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
[7] 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
[8] 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
[9] Majed O. Al-Dwairi, Ziad A. Alqadi, Amjad A. AbuJazar and Rushdi Abu Zneit, Optimized True-Color Image Processing, World Applied Sciences Journal 8 (10): 1175-1182, 2010 ISSN 1818-4952.
[10] H.G. Barrow and J.M. Tenenbaum (1981) "Interpreting line drawings as three-dimensional surfaces", Artificial Intelligence, vol 17, issues 1–3, pages 75–116.
[11] *Lindeberg, Tony (2001), "Edge detection", in Hazewinkel, Michiel, Encyclopedia of Mathematics, Springer, ISBN 978-1-55608-010-4*
[12] T. Pajdla and V. Hlavac (1993) "Surface discontinuities in range images," in Proc IEEE 4th Int. Conf. Comput. Vision, pp. 524–528.

