Text-zone Detection and Rectification in Document Images Captured by Smartphone

Sophea PRUM

Mimos Berhad, Kuala Lumpur, Malaysia,
sophea.prum@mimos.my,
www.mimos.my

Abstract. Detection text-zone in the document images captured by smartphone is one of the main challenges in document image processing research domain. Unlike scanned document images, the document im-ages captured by smartphones, especially under unconstrained condition and environment, have some additional challenges. The document frame may appear on various backgrounds, bent, rotated, light reflection, etc. To deal with these problems, we present a pre-processing method that allows to detect and rectify text-zone in the document images captured by a smartphone. The proposed method consists of three steps: bina-rization, skew angle detection/correction and text-zone detection. The experimental results based on the recognition rates given by two end-to-end OCR systems have shown a significant improvement when applying the proposed pre-processing methods before applying any OCR system.

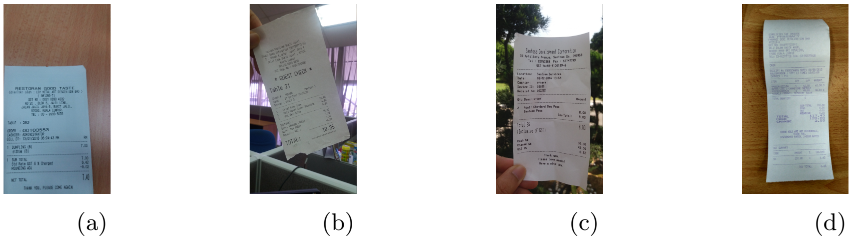
Key words: Document image processing, text-zone detection, binariza-tion, skew correction, smartphone document capture, mobile camera doc-ument capture

1 Introduction

Nowadays, smartphone and tablet are used not only as a personal camera, but also as a personal scanner. It allows users to capture any documents on hard support and convert them into a digital image instantly. However, unlike scanned document images, using these new devices, the images are captured under uncon-trolled environment. Therefore, these images may have some additional noises such as: heterogeneous illumination, distortion, out of focus blur, motion blur, light reflection etc. Physical document paper might be bent or folded and/or having low contrast text. Furthermore, the document paper can be appeared on various backgrounds. Some examples are illustrated in Fig. 1.

Since the mid 1950's, Optical Character Recognition (OCR) has been one of the most active research subjects in the domain of pattern recognition and machine learning [1, 2]. Different systems/algorithms have been presented in the literature [3, 4, 5, 6]. Different open-source [7, 8, 9] and commercial OCR systems have been used and commercialized.

These systems are generally designed for scanned document images. They provide a promising results when the document quality is good enough. However, when applying these OCR systems directly on the document images captured by smartphone, the recognition rate decreases dramatically due to the problems mentioned above. Therefore, converting these document images into editable text is a new challenge.



(a) (b) (c) (d)

Fig. 1: Example of the document images captured by smartphone. (1a) document paper does not fully appear in the image, (1b) document image captured in the office, (1c) document image captured outside, (1d) document image containing bent paper.

In this paper, we present a method for text-zone detection and rectification in document images captured by a smartphone or similar device (tablet or camera). Our proposed method is specifically designed to deal with five main problems:(1) document frame appears on various backgrounds (see Fig. 1), (2) document frame does not fully appear in the image (see Fig. 1a), (3) document image containing bent paper (see Fig. 1d), (4) light reflection and (5) low contrast text. Our contributions are mainly on binarization, skew angle detection/correction and text-zone detection methods.

This paper is organized as follows. Session 2 presents our proposed method while session 3 introduces experiment protocol and experimental results. Finally, conclusion and future work are presented in session 4.

2 Proposed system

The overview of our system is illustrated in Fig. 2. The system is divided into two steps: pre-processing and text transcription. The pre-processing step consists of 3 sub-steps: binarization, skew detection/correction and text-zone detection.

As mentioned earlier, this paper focuses the pre-processing step. Therefore, in the following sub-sections, we will focus only on the methods using in the pre-processing step. Two end-to-end OCR systems are used in the experiment session (see section 3.2) in order to evaluate the impact of the proposed pre-processing methods on the OCR recognition result.

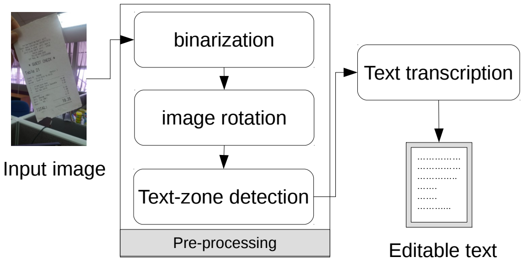


Fig. 2: Global overview of an OCR system.

2.1 Binarization

In document image processing, binarization is a crucial step which aims at ex-tracting pixels of text strokes from their background. The input image is con-verted into a matrix of two values 0 and 1, where 0 represents the text stroke pixel and 1 represents the background pixel (or vice versa). In the literature, we can classify binarization methods into two approaches: global and local ap-proaches [13].

A method using global approaches consists in choosing only one threshold value for the entire image. Otsu method [14] is one of the most popular global binarization methods which allows to compute the threshold value automatically. However, the global approaches can be only applied when the image illumination is homogeneous with high contrast image. Therefore, these approaches cannot be applied on the document image captured by smartphone since in general, the illumination is heterogeneous and some documents have very low contrast, as illustrated in Fig. 1.

Unlike global approaches, a method relying on local approaches consists in selecting a threshold value for each pixel or sub-region. Different methods have been presented in the literature [15, 16, 17, 18]. The methods presented by Niblack [15], Sauvola [16] and Wolf [17] rely on the mean and standard deviation of each sub-region of the input image. These methods are very sensitive to the coefficient value of the standard deviation, especially, when applying on low contrast image. The method presented by Su [18] relies on Canny edge map. This method provides a good performance only when the edge map is successfully extracted.

In this paper, we present a new binarization method which is able to deal with low contrast image. This proposed method used the combination of gradient image and local adaptive binarization method, as illustrated in Fig. 3. First, the input color image is converted into a grayscale image. Then, we apply a local binarization method which relies on Gaussian weighted in order to extract text stroke pixels. For each pixel $I_{(x,y)}$ of the input image I, the threshold value $T_(x, y)$ is the average of Gaussian weighted in the sliding window. This method allows to extract text pixels in a very low contrast image. However, the binary

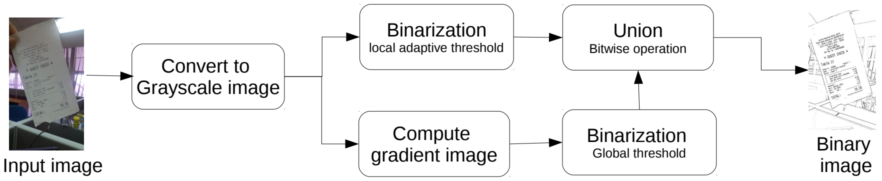


Fig. 3: Workflow of proposed binarization method.

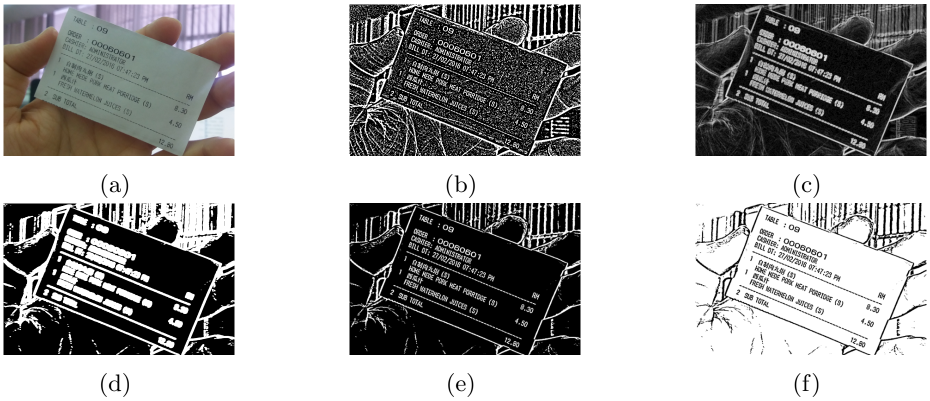


Fig. 4: Example of binarization result.

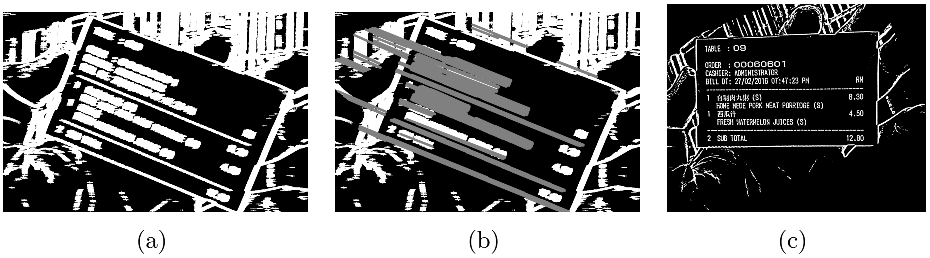
image contains a lot of noises since some background pixels are also extracted (see Fig. 4b).

In order to remove these noise pixels, we compute the second binary image to extract the approximate text stroke position. This second binary image will be used as a mask image to extract the text stroke pixels from the first binary image. More specifically, given the grayscale image, we compute gradient image by combining the horizontal and vertical Sobel Operator [19], as illustrated in Fig. 4c. Then, we apply Otsu binarization method in order to remove the background pixels. A dilatation operation is applied in order to add the missing text strokes pixels (see Fig. 4d). Finally, to extract text pixel from the first binary image, bitwise 'AND' operation is applied. The final result of our binarization is illustrated in Fig. 4e. Fig. 4f is the negative image of Fig. 4e.

2.2 Skew angle detection and correction

The document images captured by smartphone (tablet and camera) are more or less rotated according the position of the document and smartphone camera. Therefore, skew angle detection and correction is an unavoidable step.

In our research, we focus on receipts. Therefore, we consider that text-lines in a document image are approximately on a parallel line. As a consequence, skew angle is the average of all the angles composing by text-lines and the x-axis.



(a) (b) (c)

Fig. 5: Example of skew detection and correction.

Given the binary image resulted from the previous step (see Fig. 4e), the dilatation method is applied in order to create a straight approximate line on each text-line (see Fig. 5a). Then, Hough-line transform [20] is applied in order to detect all the possible straight lines in the images. In our system, we are in-teresting only on the straight lines that pass through the text-lines. We assume that the text-lines angles comprise between 0 to 45 degree, or 135 to 225 degree, or 315 to 360 degree. Therefore, all the detected straight that satisfy this condition will be considered as Potential Straight Lines (PSLs). Fig. 5b illustrated an example of detected PSLs.

However, some of these PSLs can be noise since they do not pass through the text-lines. In addition, the angles of these PSLs with x-axis may comprise between 0 to 45 degree, or 135 to 225 degree, or 315 to 360 degree. Therefore, compute the average of these angles and used as skew angle will have a high bias.

In order to compute a precise skew angle, we create an angle histogram of these detected PSLs, the objective of which is to count the number PSLs that have approximately the same angle. More specifically, for each PSL (l_i), the angle α_i composing by the line l_i and x-axis is computed. A histogram is crated by considering the gap of 5 degree (see an example in Fig. 6). The gap that has the highest number of PSLs is considered as the potential gap (β). In our example, the potential gap $\beta =$[20, 24] degree. The skew angle δ can be computed by:

$$\delta = \frac{1}{L}\sum_{i=1}^{L}\alpha_i \qquad (1)$$

where $\alpha_i \in \beta$ and L is the total number of PSLs having the angles belong to β

Once the skew angle δ is detected, Affine transform is used for skew correc-tion. Given the input image I and and skew angle δ, the corrected images I' can be computed by:

$$I'_{(x,y)} = I_{(M_{(11)}x+M_{(12)}y+M_{(13)},M_{(21)}x+M_{(22)}y+M_{(23)})} \qquad (2)$$

where M is the rotation 2D matrix defined by:

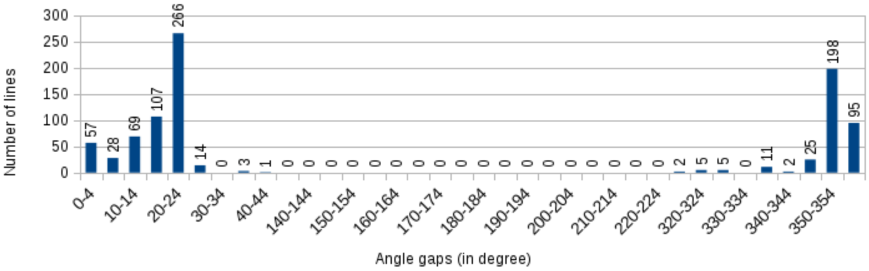


Fig. 6: Example of angle histogram.

$$M = \begin{bmatrix} a & b & (1-a) * c.x - b * c.y \\ -b & a & b * c.x + (1-a) * c.y \end{bmatrix}$$

where $a = s * cos(\delta)$, $b = s * sin(\delta)$, and s is the scale value. In our case, we initial $s = 1$. c is the central point of I' which can be estimated by: $c = (E/2, E/2)$, $E = max\{I.height, I.width\}$.

Finally, we obtained skew corrected image as illustrated in Fig 5c.

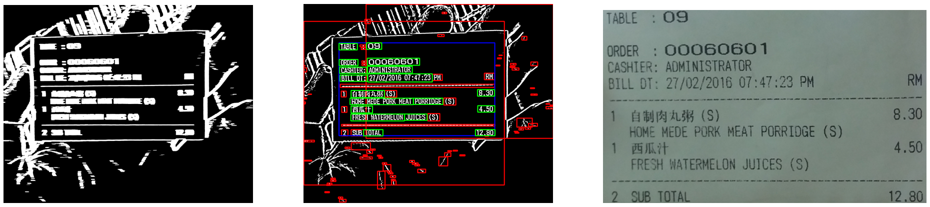
2.3 Text-zone detection

Due to background complexity in the input image, some background pixels can-not be removed in the binarization step (see sub-section 2.1). Therefore, text-zone detection steps is required in order to crop the text-zone from the input image.

As mention earlier, the document paper can be bent and appeared on differ-ent and heterogeneous backgrounds. Therefore, detection text-zone in this kind of document images is one of the main challenges. To deal with this similar prob-lem, based on our best Knowledge, two approaches have been presented in the literature.

The first approaches consists in detecting and cropping the edge of docu-ment frame in the given image. In a recent competition [10], all the participant methods in challenge-2 try to detect the four corners of the document frame by computing the intersection positions of four straight lines (upper, lower, left and right) on the edge of the document frame. Then, skew and distortion correction methods are applied in order to crop the document paper. However, this kind of methods can only be applied in the case that the document paper is flat, the document frame fully appears in the image, and with homogeneous background.

The second approaches aims at removing the marginal noise in the document image. The system presented in [11] relies on edge density. This method relies on the assumption that text areas have a low density of edges while marginal noise areas have a very high density of edges. Unfortunately, this assumption cannot be applied in our document images. The system presented in [12] consists in detecting page border using the combination of profile projection and connected



(a) (b) (c)

Fig. 7: Result of text-blocks and text-zone detection.

component labeling process. However, this method can only be applied on the image having black marginal noise.

In order to deal with this problem, we proposed a new text-zone detection method. We consider that the text-zone is the zone which contains all the text-block in the document image.

Given the skew corrected image resulted from the previous step (see Fig. 5c), dilation method is applied in order to link the characters that belongs to the same text-block (see Fig. 7a). Then, all the connected components are extracted. For each connected component c_i, its width (w_i) and high (h_i) can be computed. The connected component c_i is considered as a text-block if $min_h < h_i < max_h$ and $w_i > min_w$, where the value of min_h, max_h and min_w can be roughly fixed for each document type.

After applying this filter rules, we obtain a list of main text-blocks (T) as illustrated in green boxes in Fig. 7b. The red boxes are not considered as text-block. Then, the bounding box of text-zone represented by the top-left corner point P_1 and the bottom-right corner point P_2, can be defined by:

$$P_1 = (\min_{c_j \in T}(c_j.x), \min_{c_j \in T}(c_j.y)) \; and \; P_2 = (\max_{c_j \in T}(c_j.x), \min_{c_j \in T}(c_j.y)) \qquad (3)$$

An example of detected text-zone is illustrated by a blue box in Fig. 7b.

3 Experimentation

3.1 Data sets

In this experiment, we use 20 receipts issued by different shops and supper-markets. The document papers can be bent, folded and/or with low contrast text. The Samsung J5 smartphone is used to capture all the images by using two different methods:

We captured 2 sets of document images. The images in the first set (set 1) are captured by placing the document paper on a desk (homogeneous background) inside the building. The images in the second set (set 2) are captured with heterogeneous background, inside and outside the building.

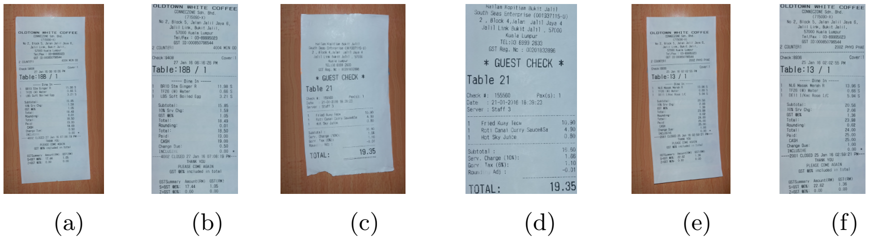

(a) (b) (c) (d) (e) (f)

Fig. 8: Example of cropped images given by our text-zone detection method from set 1. Fig. 8a, 8c and 8e are the input images and Fig. 8b, 8d and 8f are the cropped images resulted from our pre-processing step.

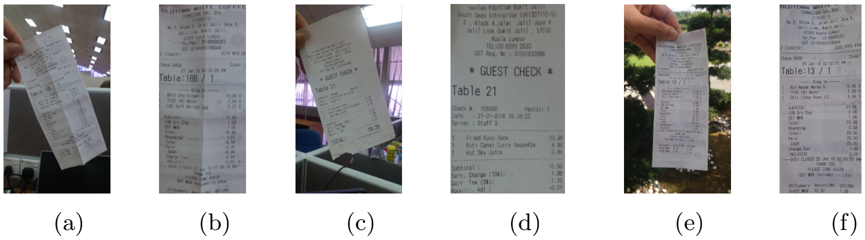

(a) (b) (c) (d) (e) (f)

Fig. 9: Example of cropped images given by our text-zone detection method from set 2. Fig. 9a, 9c and 9e are the input images and Fig. 9b, 9d and 9f are the cropped images resulted from our pre-processing step.

3.2 Experimental results

In this experiment, we rely on OCR recognition rates to evaluate the impact of our pre-processing methods. Two end-to-end OCR systems are used: Tesser-act [7] (open source OCR system) and ABBYY finereader (commercial OCR system). It is worth to mention that both systems have their own page-layout analysis method that allows to identify text and image zones before applying the recognition process. These two OCR systems are applied on the raw document images and on the cropped document images resulted from our pre-processing method. Some examples images are illustrated in Fig. 8 and 9.

Table 1 shows the recognition rates given by Tesseract and Finereader OCR systems applying on raw and cropped images. We can notice that:

- On set 1, when applying Tesseract and Finereader directly on the raw images, the recognition rate is 67.48% and 73.21% respectively. However, when applying these OCR systems on the cropped images resulted from our pre-processing method, it shows an improvement of recognition rate of 9% on both Teserract and Finereader OCR.
- Idem for set 2. Applying Tesseract on cropped images shows a significant improvement from 39.01% to 60.65% (+ 21%) while applying Finereader on cropped images, the recognition rate improves from 61.60% to 72.69% (+11%).

- As mentioned earlier, both OCR systems have their own text detection method allowing to localize text-zone in the document images. The significant im-provement when applying these OCR systems on the cropped images has shown the efficiency of the proposed pre-processing method compared to the pre-processing methods used in both OCR systems.
- Although, there is significant improvement in the recognition rates, we can notice that the recognition rates remain low. This low recognition rate can explain the complexity of our document images. As mentioned in 3.1, the document papers used in our experiments are bent, folded and or with low contrast text. In addition, both OCR systems are not specifically trained to recognize receipts. Usually, fonts using in receipts are different from fonts using in ordinary printed documents. Therefore, training these OCR systems with receipt documents will improve the recognition rates.

Data set	Tesseract		Finereader	
	raw images (%)	cropped images(%)	raw images(%)	cropped images(%)
Set 1	67.48	76.39	73.21	82.94
Set 2	39.01	60.65	61.60	72.69

Table 1: Recognition rate given by Tesseract and ABBYY finereader.

4 Conclusion and future works

In this paper, we proposed a pre-processing method for text-zone detection and correction in the document images captured by a smartphone. This method consists of three steps: binarization, skew detection/correction and text-zone de-tection. The proposed binarization method relies on the combination of gradient image and local adaptive threshold based on Gaussian weighted. This method allows to extract text stroke pixels from its background and can be applied on low contrast image. The skew detection and correction method consists in de-tecting the skew angle of the document paper in the image and correcting it in order to ensure that text-lines are in the horizontal position. Finally, the text-zone detection method consists in detecting all text-zone of the document image and removing its background.

The experimental results presented in section 3 have shown a significant im-provement when applying our pre-processing method before applying any OCR method.

However, in this experiment, only one smartphone is used to capture the images. The resolution of the smartphone camera might have a great impact on the method. In our future work, different kind of smartphones will be considered.

References

1. S. Mori and C. Y. Suen and K. Yamamoto: Historical review of OCR research and development, Proceedings of the IEEE, pp 1029–1058, vol. 80 (1992)
2. Øivind Due Trier and Anil K. Jain and Torfinn Taxt: Feature extraction methods for character recognition-A survey, Pattern Recognition Journal. 641–662 (1996)
3. S. IMPEDOVO, L. OTTAVIANO, and S. OCCHINEGRO: OPTICAL CHARAC-TER RECOGNITION A SURVEY, International Journal of Pattern Recognition and Artificial Intelligence, vol. 5, pp. 1-24 (1991)
4. S. Mori and H. Nishida and H. Yamada: Optical Character Recognition, John Wiley & Sons, Inc., New York, USA
5. U. Pal and B.B. Chaudhuri: Indian script character recognition: a survey, Journal of Pattern Recognition, vol. 37, pp. 1887—1899 (2004)
6. G. Debashis and D. Tulika and P.S. Adamane: Script Recognition: A Review, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, pp. 2142–2161 (2010)
7. R. Smith: An Overview of the Tesseract OCR Engine, Proceedings of the Ninth International Conference on Document Analysis and Recognition, vol. 2, pp. 629–633, Washington. DC, USA (2007)
8. Thomas M. Breuel: The OCRopus open source OCR system, Proc. SPIE 6815, Document Recognition and Retrieval XV (2008)
9. A.Graves: RNNLIB: a recurrent neural network library for sequence learning prob-lems. http://sourceforge.net/projects/rnnl/
10. J. Burie and J. Chazalon and M. Coustaty and S. Eskenazi and M. M. Luqman and M. Mehri and N. Nayef and J. Ogier and S. Prum and M. Rusinol: Competition on smartphone document capture and OCR (SmartDoc), Document Analysis and Recognition (ICDAR), pp. 1161–1165, Nancy, France (2015)
11. W. Peerawit, A. Kawtrakul: Marginal noise removal from document images using edge density. In: 4th Information and Computer Engineering Postgraduate Work-shop, Phuket, Thailand (2004)
12. N. Stamatopoulos, B. Gatos and A. Kesidis, "Automatic Borders Detection of Camera Document Images", 2nd International Workshop on Camera-Based Docu-ment Analysis and Recognition, 2007, pp. 71-78.
13. S. S. Lokhande and N. A. Dawande, A Survey on Document Image Binarization Techniques, International Conference on Computing Communication Control and Automation (ICCUBEA), pp. 742-746, 2015
14. N.A. Otsu, A Threshold Selection Method from Gray-Level Histograms, IEEE Transactions on Systems, Man, and Cybernetics, pp. 62-6, vol. 9, 1979.
15. W. Niblack, An Introduction to Digital Image Processing, Englewood Cliffs, N.J.: Prentice Hall, pp. 115-116, 1986.
16. J. Sauvola and M. Pietikinen, Adaptive document image binarization, Pattern Recognition Journal, pp. 225–236, vol. 33, 2000
17. C. Wolf and J.-M. Jolion and F. Chassaing, Text Localization, Enhancement and Binarization in Multimedia Documents, International Conference on Pattern Recog-nition, pp. 1037-1040, 2002
18. B. Su and S. Lu and C. L. Tan, Robust Document Image Binarization Technique for Degraded Document Images, IEEE Transactions on Image Processing, pp. 1408- 1417, vol. 22, 2013
19. I. Sobel, History and Definition of the so-called "Sobel Operator" (Report)
20. J. Illingworth and J. Kittler, A survey of the hough transform, Computer Vision, Graphics, and Image Processing, pp. 87-116, vol. 44, 1988.