Non-Visual Document Recognition for Blind Reading Assistant System

[*1]Ching-Tang Hsieh, [1]Cheng-Hsiang Yeh, [2]Tsun-Te Liu and [2]Ku-Chen Huang
[1] *Dep. of Electrical Engineering, Tamkang University, New Taipei, Taiwan, R.O.C*
[2] *Dep. of Physical Education, Tamkang University, New Taipei, Taiwan, R.O.C*
[*]*Corresponding Author: hsieh@ee.tku.edu.tw*

Abstract

As time goes on, a huge mass of progressive knowledge is developed and then the e-book is built for paper reduction and environmental protection. Therefore, many historical documents over the past don't exist. Research on blind reading aid device is a popular topic gradually, but there are some drawbacks on these methods. For example, electronic documents of many old dated of historical documents don't read because format should be limited to e-document. Users can't read any specified region of the document as he wishes. A novel blind reading aid device is proposed without e-document and the user only need to point the document with his finger. This system is composed of third parts. First, we use rectangle detection method to catch the region for document under the Microsoft Kinect. Next, dilation method and projection profile methods are used in order to execrate text and constructed coordinate database. Finally, skin detection, BEA method and depth image of Microsoft Kinect can get the coordinates of user's finger when user wishes to read the document, and then match with constructed coordinate database to get the character. Then system output is obtained via text to speech.

Keywords*: Assistive Device; Braille; Document Recognition*

1. Introduction

Braille system in 1821 was devised by Frenchman Louis Braille. This system can enable to read and write through touch for blind and partially sighted people. However, it is difficult to learn Braille system and inconvenient for newly blind people because users need someone to assist in learning and training stage. Araki et al [1] developed a spoken dialogue system that can allow visually impaired to learn Braille system by himself. Saad et al [2] designed a system for implementation of optical Arabic Braille Recognition (OBR) with voice and text conversion. [3-4] presented a Braille cells recognition method by image processing technique.

As time goes on, the eBook publishing industry is rapidly growing, and more and more documents are digitized and uploaded to the Internet. Because Braille book is transformed by electronic document, this assistant tool is rapidly growing, too. In fact, there are some drawbacks as burdensome and uncomfortable when users held and carried the Braille book. To solve above problems, Velazquez et al [5] developed a reading assistive device which is able to reproduce electronic books in portable electronic tactile displays and to store in a USB memory drive. However, many historical documents over the past don't exist, because this assistive device is limited to e-document. While also users can't read any specified region of document as they wished.

To overcome the above problems, we propose a reading system for visually impaired persons. We first propose to detect the area of the document by using Microsoft Kinect. The next steps consist of text detection and detection of the position of the user's fingers. The final output is obtained via text to speech or Braille display.

This paper is organized as follows. In the next section, we refer to our proposed method. Section 3 reports the experimental results. Finally, conclusion and future work is shown in section 4.

2. Proposed Method

The proposed a blind reading system comprises third steps as Fig. 1. First, we use rectangle detection method to catch the region of document under the Microsoft Kinect. Next, dilation method and projection profile method can execrate text and construct coordinate database. Finally, we combine skin detection method, BEA method and depth image of Microsoft Kinect data to get the coordinates of user's finger, and system output is obtained via text to speech.

doi:10.4156/rnis.vol14.81

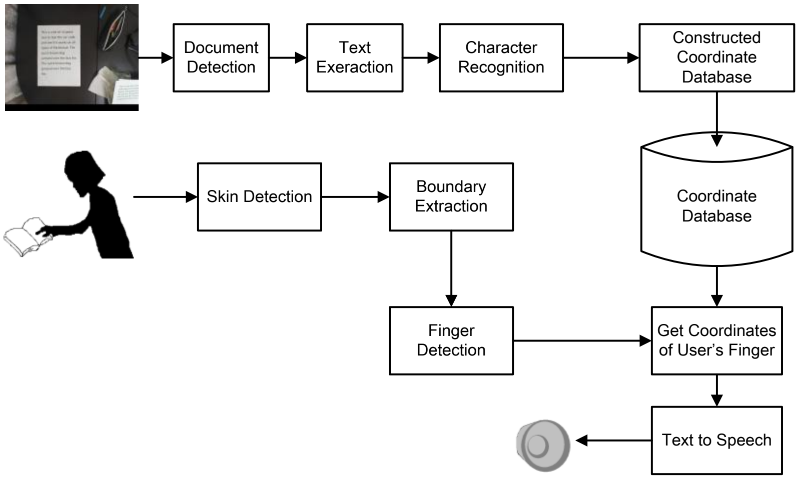


Figure 1. Block diagram for proposed blind reading system

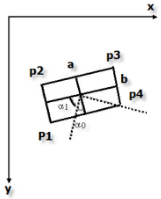


Figure 2.
A rectangle

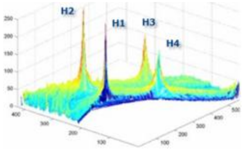
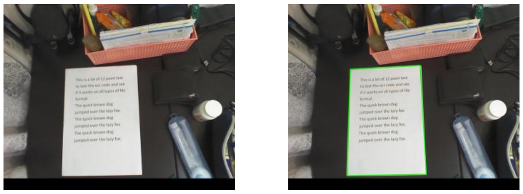
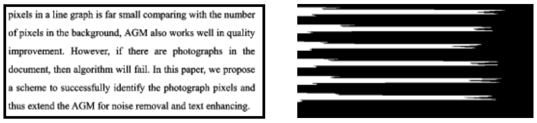


Figure 3.
Hough space transform



(a) Text image (b) Detection result

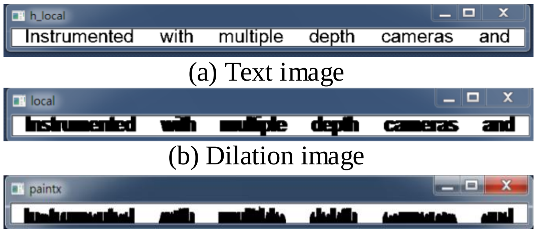
Figure 4. Document Detection



pixels in a line graph is far small comparing with the number of pixels in the background, AGM also works well in quality improvement. However, if there are photographs in the document, then algorithm will fail. In this paper, we propose a scheme to successfully identify the photograph pixels and thus extend the AGM for noise removal and text enhancing.

(a) English document *(b) Horizontal projection profile*

Figure 5. Extraction of Textlines



Instrumented with multiple depth cameras and

(a) Text image

(b) Dilation image

(c) Vertical projection profile

Figure 6. Extraction of words

2.1 Document Detection

2.1.1 The Hough Transform [6]

The Hough Transform is a powerful method for detecting linear structures in images. Any line on the xy plane can be described as Eq. (1),

$$\rho = x\ cos\theta + y\ sin\theta \tag{1}$$

whereρ is the normal distance and Θ is the normal angle of a straight line. In practical applications, the angles θ and distances ρ are quantized and obtained an array $C(\rho_k, \theta_l)$. The local maxima of $C(\rho_k, \theta_l)$ can be used to detect straight line segments passing through edge points.

Assume a rectangle with vertices $P_1=(x_1,y_1)$, $P_2=(x_2,y_2)$, $P_3=(x_3,y_3)$ and $P_4=(x_4,y_4)$ as Fig. 2. The image of this rectangle in the hough space is shown in Fig. 3. There are four peaks labeled as $H_1 = (\rho_1, \theta_1)$, $H_2 = (\rho_2, \theta_2)$, $H_3 = (\rho_3, \theta_3)$ and $H_4 = (\rho_4, \theta_4)$, that correspond to the four sides of the rectangle P_3P_4, P_1P_2, P_2P_3, P_1P_4.

These four peaks can be observed satisfy specific geometric relations:

1. They appear in pairs. The first one is formed by peaks H1 and H2 at $\theta = \alpha_1$; the second one is formed by peaks H3 and H4 at $\theta = \alpha_0$.
2. The two pairs are separated by $90^{\circ}(|\alpha_1 - \alpha_0| = 90^{\circ})$.

3. The heights of two peaks within the same pair are exactly the same because opposite sides are equal in length, i.e., $C(\rho_1,\theta_1)$= $C(\rho_2,\theta_2)$=b and $C(\rho_3,\theta_3)$= $C(\rho_4,\theta_4)$=a.
4. The same pair distance between peaks are equal in length of rectangle, i.e., P_1-$P_{2=}$a and P_3- $P_{4=}$b.

2.1.2 Detecting Rectangle Patterns [7]

Let $H_1 = (\rho_1,\theta_1)$, $H_2 = (\rho_2,\theta_2)$,…, $H_n = (\rho_n,\theta_n)$ denote the n peaks of $C(\rho\ \ ,\theta\ \ )$. The next step is to find the peaks satisfying the conditions listed in these n peaks.

First, all peaks are scanned and peaks Hi, Hj are satisfying the following conditions:

$$\Delta\Theta = \left|\Theta_i - \Theta_j\right| < T_\Theta$$
$$\left|C(\rho_i,\theta_i) - C(\rho_j,\theta_j)\right| < T_L \frac{C(\rho_i,\theta_i) + C(\rho_j,\theta_j)}{2} \qquad (2)$$

where T_θ is an angular threshold and T_L is a normalized threshold. Let $P_k(\alpha_k,\xi_k)$ denote pair of peaks Hi and Hj satisfying Eq.(3).

$$\alpha_k = \frac{1}{2}(\theta_i + \theta_j), \xi_k|\rho_i - \rho_j| \qquad (3)$$

Finally, be to compare all pairs of extended peaks P_k (k=1,2..) and are satisfying the following conditions:

$$\Delta\alpha = \left||\alpha_i - \alpha_j| - 90^\circ\right| < T_\alpha \qquad (4)$$

where T_α is an angular threshold. The four peaks are detected rectangle if these peaks are satisfying Eq. (4).

In this paper, we use Canny's operator [8] to detect edges in images and the choice of parameters D_{min} and D_{max}. The choice of D_{min} and D_{max} is made based on the sizes of rectangles to be detected. Fig. 4 is showed the result of document detection.

2.2 Text Exeraction, Character Recognition and Constructed Coordinate Database

1. Use Otsu's thresholding method [9] to separate text after document detection.
2. Extract textlines by horizontal projection profile as Fig. 5.
3. Dilation operator associated bounding box for each word and vertical projection profile extracted word as Fig. 6.
4. Character recognition used the tesseract [10], Google's OCR engine.
5. According to step 3 and 4 can record character coordinate and construct coordinate database.

2.3 Touch Sensor and Text to Speech

We want to get the coordinates of user's finger when users wish to read the region of document so we must propose a method to find user's finger and determine whether user's finger is touching. First, hand is extracted by skin detection. Next, we extract boundary of the hand and calculate the angle to detect finger. Finally, depth data is determined whether user's finger is touching.

2.3.1 Skin Detection

H. Rein-Lien [11] et al designed a skin detection algorithm for color images in presence lighting compensation technique and a nonlinear color transformation. Authors consider the impact of different Y values, and transform color format using segmental nonlinear. And regard the chroma Cb and Cr as functions of the luma Y: $C_b(Y)$ and $C_r(Y)$. Let the transformed chroma be $C_b'(Y)$ and $C_r'(Y)$. The skin color model is specified by the centers and spread of the cluster and is used for computing the transformed chroma. Then the formula of transformation from YC_bC_r coordinate space to $YC_b'C_r'$ coordinate space is:

$$\text{Cí} = \begin{cases} [C_i(Y) - \bar{C}_i(Y)]\dfrac{W_{ci}}{W_{ci}(Y)} + \bar{C}_i(Y) & \text{if } Y < K_l \text{ or } Y > K_h \\ C_i(Y) & \text{if } Y \in [K_l,\ K_h] \end{cases} \qquad (5)$$

$$\text{Where } W_{ci}(Y) = \begin{cases} WL_{ci} + \frac{(Y - Y_{min})(W_{ci} - WL_{ci})}{K_l - Y_{min}} & \text{if } Y < K_l \\ WH_{ci} + \frac{(Y_{max} - Y)(W_{ci} - WH_{ci})}{Y_{max} - K_h} & \text{if } Y > K_h \end{cases} \tag{6}$$

$$\text{Where } \overline{C_b}(Y) = \begin{cases} 108 + \frac{(K_l - Y)(118 - 108)}{K_l - Y_{min}} & \text{if } Y < K_l \\ 108 + \frac{(Y - K_h)(118 - 108)}{Y_{max} - K_h} & \text{if } Y > K_h \end{cases} \tag{7}$$

$$\text{Where } \overline{C_r}(Y) = \begin{cases} 154 - \frac{(K_l - Y)(154 - 144)}{K_l - Y_{min}} & \text{if } Y < K_l \\ 154 + \frac{(Y - K_h)(154 - 132)}{Y_{max} - K_h} & \text{if } Y > K_h \end{cases} \tag{8}$$

where C_i in (5) and (6) is either C_b or C_r, W_{cb}= 46.97, WL_{cb}=23, WH_{cb}=14, W_{cr}=38.76, WL_{cr}=20, WH_{cr}=10, K_1=125, and K_h=188 as table 1. The elliptical model for the skin tones in the transformed $C_b' C_r'$ space is described in Eq. (7) and (8), and is depicted in Table 1,

$$\frac{(x - ec_x)^2}{a^2} + \frac{(y - ec_y)^2}{b^2} = 1 \tag{9}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} Cb' - c_x \\ Cr' - c_y \end{bmatrix} \tag{10}$$

where c_x=109.38, c_y=152.02, Θ=2.53, ec_x=1.60, ec_y=2.41, a=25.39, and b=14.03 are computed from the skin cluster in the $C_b' C_r'$ space. Since this skin color method has good results, we use this method in step of extracting hand as Fig. 7.

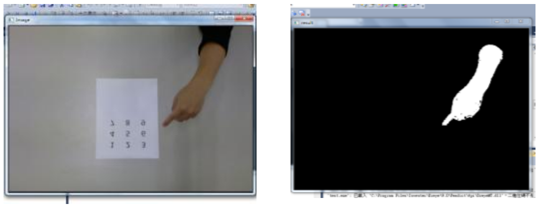

(a) Text image (b) Detection result

Figure 7. Skin detection

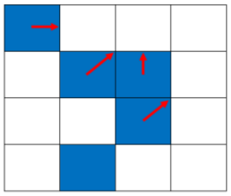


Figure 8. Illustration of boundary extraction algorithm

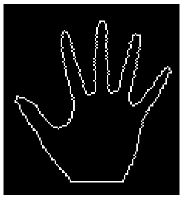


Figure 9. Result of the boundary of the hand extracted

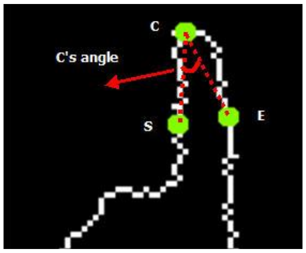


Figure 10. Example of finger's angle calculated

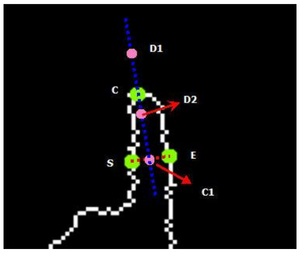


Figure 11. Example of surface point and hand point

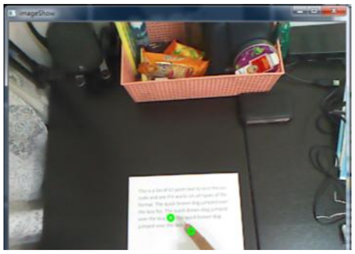


Figure 12. Example of touch sensor

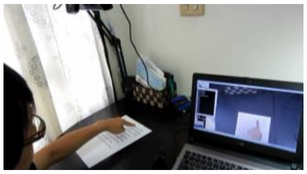


Figure 13. Reading the document experiment

Table 1. Parameter Overview in [11]

W_{cb}	WL_{cb}	WH_{cb}	W_{cr}	WL_{cr}
46.97	23	14	38.76	20
WH_{cr}	K_l	K_h	Y_{min}	Y_{max}
10	125	188	16	235

2.3.2 Contour Extraction

We extract the boundary of the hand by Liu [12] and comprise four main steps:

1. Scan the binary image from top to bottom and left to right and find the first 255 pixel as the starting point and record the position into x and y arrays, i.e., x[1]=i and y[1]=j. Let the direction number of t he starting pixel be 0.
2. We assume that i is the direction number of current pixel and j is the direction number for searching next pixel and define a decision rule for searching next pixel as: (i,j)={(0,6),(1,7),(2,0),(3,1),(4,2),(5, 3),(6,4),(7,5)}. Fig. 8 is showed illustration of boundary extraction algorithm.
3. Scan clockwise until the first pixel value which is 255 and record its position into x and y arrays.
4. Repeat steps 2 to 4 until the first pixel value which is the same as the starting pixel. Fig. 9 is showed the result of extract the boundary of the hand.

2.3.3 Finger Detection

After extract the hand boundary, we can find that the angle value is general extremely small when hand boundary is calculated by law of cosines. Therefore, we use this method to detect finger. Comprise four main steps:

1. Assume that k is length and start point S coordinate is (x[1],y[1]). Center point C coordinate is (x[k/2],y[k/2]) and end point E coordinate is (x[k],y[k]) as Fig.10.
2. Calculate the angle of C is described in Eq. (11) and record the angle into G array as Fig. 10.

$$\angle \mathrm{C} = \cos^{-1}\frac{\overline{\mathrm{SC}}^2 + \overline{\mathrm{CE}}^2 - \overline{\mathrm{SE}}^2}{2 \times \overline{\mathrm{SC}} \quad \times \overline{\mathrm{CE}}} \tag{11}$$

3. Next start point S coordinate is (x[2],y[2]) and repeat steps 1 to 2 until the start point coordinate which is the same as the starting coordinate.
4. Find the angle is the minimum of all in G array and the angle must be small than δ so this point is finger detection that we want to search.

2.3.4 Touch Event and Text to Speech

Wilson [13] presented the application of depth-sensing cameras to detect touch on a tabletop. This method offered certain interesting advantages but also has a drawback. In fact, touch signal is instable and sensitive when users touch on a tabletop surface because the depth of the surface values is recorded. In this method, the worst case error was about 6 pixels (about 15mm at the surface) and 3 pixels (7mm). To result this problem, we use depth of user's finger and the depth of tabletop at the moment to detect finger when users wish to read the document and comprise five main steps:

1. After the point C, S and E is found in above stage, we have to find the reference points as user's finger depth and tabletop depth. Thus, I can obtain center C1 point by Eq. (12) as Fig. 11.

$$\mathrm{C1(x,y)} = \left(\frac{\mathrm{S_x + E_x}}{2}, \frac{\mathrm{S_y + E_y}}{2}\right) \tag{12}$$

2. Find the linear equation by C1 and C as Fig. 11.
3. Assume the distance of surface point D1 and hand point D2 from C is β respectively, and we can define surface point D1 and hand point D2 in the above linear equation as blue dotted line of Fig. 11.
4. Let $D1_{depth}$ is the depth of surface point D1 and $D2_{depth}$ is depth hand point D2. If $D2_{depth}$- $D1_{depth} \leqq 1$, it is represented that user's finger is touching and gets the coordinate of user's finger as Fig. 12.
5. Get the character from construct coordinate database and obtain output via text to speech.

3. Experiment Results

We used Microsoft Kinect camera for experimentation and tested our system at 50cm height as Fig. 13. The application has been implemented in Visual C++ using proposed methodology and the OpenCV libraries. The application has been tested on Intel Core i5 running at 2.5 GHz. We configured the camera to report depth shift data in a 640×480 16 bit image at 30Hz and color shift data in a 1280×760 32 bit image at 15Hz. Our sizes of rectangles was chosen in document detection stage, and the choice of parameters D_{min} and D_{max} was 100x100 pixels and 1000x1000 pixels respectively. The finger maximum angleδ was 50 in finger detection stage and the distance of surface point D1 and hand point D2 from C was 10 pixels.

In evaluating the performance of our technique, we are interested in the accuracy of touch position and character recognition. In the accuracy of touch position, the worst case error was about 1 pixel (0.4 cm). In character recognition, our recognition rate was around 93.2% when font size bigger than 16pt.

4. Conclusion and Future Work

This paper presented a blind reading system without e-document and the user only need to point the document with his finger when he wished. Currently, the system performs well and in the accuracy of touch position in recognition of text when there are only chars in document. In character recognition, our recognition rate was around 93.2% when font size bigger than 16pt. This result is despaired because hardware resolution is 1280x1024.

In future studies, we would use document images dewarping technology, extraction of text form documents technology to add robust and to solve the problem of hardware resolution.

Acknowledgments: This work was supported by the National Science Council under grant number NSC 101-2410-H-032-054-

5. References

[1] Masahiro Araki, Kana Shibahara, and Yuko Mizukami, “Spoken dialogue system for learning Braille,” 35th IEEE Annual Computer Software and Applications Conference, pp.152-156, 2011.

[2] Saad D. Al-Shamma and Sami Fathi, “Arabic Braille Recognition and Transcription into Text and Voice,” 5th Cairo International Biomedical Engineering Conference Cairo, Egypt, pp.16-18, 2010.

[3] Huaxun Zhang, Jie Li and Jia Yin, “A Research on Paper-Mediated Braille Automatic Extraction Method,” International Conference on Intelligent Computation Technology and Automation (ICICTA), pp.328-331, 2010.

[4] Abdul-Malik S. Al-Salman, Ali El-Zaart, Yousef Al-Suhaibani, Khaled Al-Hokail, AbdulAziz O. Al-Qabbany, “An Efficient Braille Cells Recognition,” 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM), pp.1-4, 2010.

[5] Ramiro Velázquez, Enrique Preza and Hermes Hernández, “Making eBooks Accessible to Blind Braille Readers,” IEEE International Workshop on Haptic Audio Visual Environments and their Applications, pp.25-29, 2008.

[6] R. Duda and P. Hart. Use of the hough transform to detect lines and curves in pictures. Communications of the ACM, vol.15, no.1, pp.11–15, 1972.

[7] C.R. Jung and R. Schramm. “Rectangle Detection based on a Windowed Hough Transform”.Proceedings of the Computer Graphics and Image Processing, pp 113-120, 2004.

[8] J. Canny. A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, pp. 679–698, 1986.

[9] Otsu, N., A Threshold Selection Method from Gray-Level Histogram, IEEE Transactions on Systems, Man, and Cybernetics, vol. 9, pp. 62-66, 1976.

[10] Tesseract. /http://code.google.com/p/tesseract-ocr/S.

[11] H. Rein-Lien, M. Abdel-Mottaleb and A.K Jain, Face detection in color images, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no.5, pp.696-706, 2002.

[12] Y. H. Liu, Feature Analysis and Classifier Design and Their Applications to Pattern Recognition and Data Mining, Ph.D. Disseration, National Taiwan Univ., R.O.C., 2003.

[13] Andrew D. Wilson, “Using a Depth Camera as a Touch Sensor,”ITS’10, Saarbrücken, Germany, pp. 69-72, 2010.