IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 6, No 1, November 2013
ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784
www.IJCSI.org

Smart Braille System Recognizer

Aisha Mousa, Hazem Hiary, Raja Alomari, and Loai Alnemer

King Abdullah II School for Information Technology, The University of Jordan
Amman, 11942, Jordan

Abstract

Braille is a system that allows visually impaired people to write and read. It utilizes the finger touch on raised dots produced by specialized machine. In this paper, we propose a fully system to recognize characters for a single side Braille document. We also present an extensive review for Braille Recognition systems and related research efforts. Our Braille recognition system is entirely flexible to the size of the scanned image. We improve each step starting from the image acquisition until the Braille cell recognition final stage. Our system includes an image acquisition stage, image pre-processing for noise removal, modified image segmentation, feature extraction, and then character recognition. Our proposed system achieved over 94.39% dot recognition accuracy. Moreover, our system is applicable to any language and to both Grades one and two.

Keywords: *Optical Braille Recognition, Image Segmentation, Character Recognition.*

1. Introduction

Blind people are an integral part of all societies and they, as everyone else, can play an effective role in the development. Therefore it was necessary to provide and support those people with systems and technologies to allow communication and interaction with each other and with people without vision problems. The major senses used by visual impaired people are the hearing and the touch feeling. These two senses become more advanced and sensitive than for non-blind people. The most famous communication system for blind people is the Braille system which depends on the sense of the touch of the finger. Braille is a system that allows visually impaired people to read through touch using a series of raised dots on special papers which can only be read using the fingers. These dots are written using a specialized machine.

Each Braille "cell" or character consists of six dots coordinated in a rectangular shape. The rectangle consists of two columns with three dots each. In Braille system, a dot may be raised at any combination for the six positions; hence, 64 combinations are available ($2^6 = 64$). The Braille dot dimensions are suitable for the tactile resolution of the fingers of the reader. The inter-character spaces express a word. The interior inter-areas distances are well-defined by the Library of Congress. Approximately, the height of a dot is 0.5 mm (0.02 inches), the Horizontal Distance (HD) and Vertical Distance (VD) between the centers of the dots in a Braille cell is 2.5 mm (0.1 inches). Approximately, the distance between dots on neighboring cells is 3.75 mm (0.15 inches) horizontally (distance between horizontal (DBH)) and 5.0 mm (0.2 inches) vertically (distance between vertical (DBV)) as shown in Figure 1 [8].

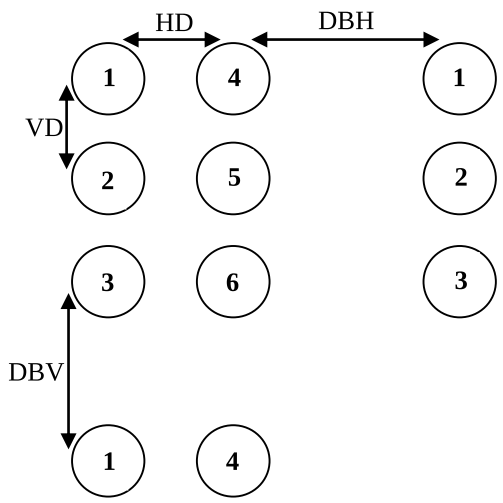

Figure 1: Braille mesh with standard distances [8].

Grade 1 Braille refers to representing only the letters in the alphabet. Then, the combination of the letters' representations makes up the words. However, this is a space and time consuming representation. Moreover, Grade 1 Braille is less popular with few books and documents. On the other hand, Grade 2 Braille system evolved for space-saving. Common words are represented with abbreviated forms such as "but", "can", and "do". These abbreviations are standardized; hence, they have been used in printing books and documents. Recently, Grade 3 Braille system has been utilized in the literature. However, it has not been standardized yet. Thus, it has not been used in official book and documents [26].

After this introduction, we present an extensive literature survey for Braille Recognition in Section 2. Section 3 presents the proposed technique. Section 4 describes our data and system requirements, experiments and results are in Section 5, followed by conclusions in Section 6.

2. Literature Review

In this section, we discuss the main concepts related to the Optical Braille Recognition (OBR) systems such as dot extraction, classification, and pre-processing. Many researchers built Braille documents recognition systems or some components within the system. We discuss the major efforts the development of similar systems.

Benjelloun et al. [2] designed a system utilizing spatial filtering, polynomial filtering, median filtering, erosion, and dilatation to extract the symbols and produce a text and called the system Lectobraille. Vidyashankar et al. [22] presented a feature extractor by collecting the number of black pixels and the area of the symbol which calculated by the fixed rectangle around each symbol. Ritchings et al. [13] proposed localization of the depressions and the protrusions in the digitized documents based on the resulting shadow of the dots. Mennens et al. [9] utilized a fixed grid on each symbol area and detected the dot based on the cross point for both horizontal and vertical lines. Blenkhom et al. [14] major contribution was to predict the letter by context of the word based on right and left context. They utilized two scans of the same document to reduce digitization noise with highly constrained digitization environment. Then with binarization, they obtain the dots to form symbols. Hermida et al. [17] utilized the luminance histogram to differentiate between verso (concave) dot and a recto (convex) dot. Similarly, Oyama et al. [19] detected both the verso (depression) and recto (Protrusions) Braille dots utilizing shadow patterns. Lau et al. [15] worked on both Chinese and English Braille text. They used a camera that actively acquires the document; apply edge enhancement and noise removal, and then segmentation utilizing edge pixels histogram. They used Laplacian of Gaussian operator to edge pixels of interest, and finally, the used an interpretation stage.

More recent efforts includes Zaghloul et al. [4] who worked on Arabic Braille with a large database of documents that include multi size and resolution digitized documents. They utilized preprocessing, cell detection, and an interpretation stages. Jiang et al. [21] worked on Mandarin Chinese. Their system consists of knowledge base for sign rules and alphabets. They perform Braille symbol segmentation and then recognize the letter as a Chinese pinyin letter. Then into characters with an error detector by the best predicted word if not found in the accompanied dictionary. Wong et al. [16] used a probabilistic neural network with simple image processing to recognize the letters. Antonacopoulos et al. [7] used an inexpensive flatbed scanner with little user interaction. They performed segmentation by an efficient two-point thresholding method to obtain: background, light, and dark. They constructed resilient grid of potential point position. Braille cells were recognized and converted into the normal printed text with a dictionary based error detection final step. Falcón et al. [6] presented the development of BrailLector to verbally speak the recognized letters. Mihara et al. [18] utilized a portable camera to design a Braille recognizer. However, this system is for small scale letters such as letters found on an elevator or an apartment number. Murray and Similarly, Murray et al. [20] designed another portable device to convert embossed Braille into normal text using a CCD camera. Al-Salman et al. [1] proposed another Braille characters recognition system using a flatbed scanner. They used a cropped grayscale image after elimination of black and white frames. For segmentation, they used a two-point threshold method to obtain: background, dark, and light. Then they applied de-skewing for the digitized document using a Binary Search Algorithm (BSA). After that the system performs a preliminary definition of Braille points before the final Braille characters recognition. Tai et al. [3] proposed a high-efficient Braille documents' cell detection approach to estimate indentations, the skewness, and distances in both vertical and horizontal directions. They estimated the obliqueness of the images by using Radon Transform. Abdelmonem et al. [10] used a flat scanner for acquisition of Arabic Braille documents. Their detection includes both a full and a partial dot.

More recent efforts include Al-Shamma et al. [8] who presented an Optical Braille Recognition (OBR) for Arabic language with text to voice conversion. Their method included a database for character prediction in case of image processing detection failure. Their method includes the document digitization, threshold the grayscale image, process skewness, edges of dots detection, then filling the openings operations and finally an image cropping for frames to produce binary dots. Yoshida et al. [5] utilized a laser fan beam projector as a sensor that helps to estimate bumps on the surface along with a CCD camera. Bhattacharya et al. [12] proposed using the Generalized Feature Vector (GFV) which detects the dots embossed on digitized Braille documents. GFV has a multimodal probability distribution in a multi-dimensional feature space. Authman et al. [11] worked on recognition and translation of colored printed Braille cells. Their system consists of several image processing stages. It includes local adaptive thresholding, morphological operations including shrinking mechanism to reshape Braille cells and make them more regular. Then they divide the image into characters and words which are then converted into ASCII and finally into Arabic letters. Chaudhary et al. [23] presented a novel method for alignment of digitized Braille document. They also concentrated on fixing corner pixels of Italic Braille patterns. They used the basic property of the triangle to calculate the angle which makes the corner dot out of alignment on a particular side. Since there is of diversity different Braille patterns, some of them begin exactly symmetrical and regular, so for asymmetrical and symmetrical Braille pattern sorting method is used.

3. Proposed System

In this work, we produce a method for single sided Braille character recognition. The Braille recognition system composed of several operations as illustrate in Figure 2. This system is highly accurate, robust, and efficient compared to the state-of-the-art systems as we show in the results section. In this section, we explain each stage within our system.

3.1. Image acquisition

We obtained images of single sided embossed Braille documents using a flat-bed scanner. The main objective for our system is robustness, functionality, and ease of use. The scanner can be used with any other application without the need to carry out complex modifications, hence, reduction in the operating cost of our system.

3.2. Image Pre-processing

We perform few image pre-processing steps to prepare the images for feature extraction. First, we convert the image from the Red, Green, Blue (RGB) color images into gray level images because color feature are irrelevant. Color information that result from digitizing the Braille documents have no meaningful usage. We convert the RGB images into gray scale using the following standard color image conversion:

$$gray = 0.2989\, R + 0.5870\, G + 0.1140\, B \qquad (1)$$

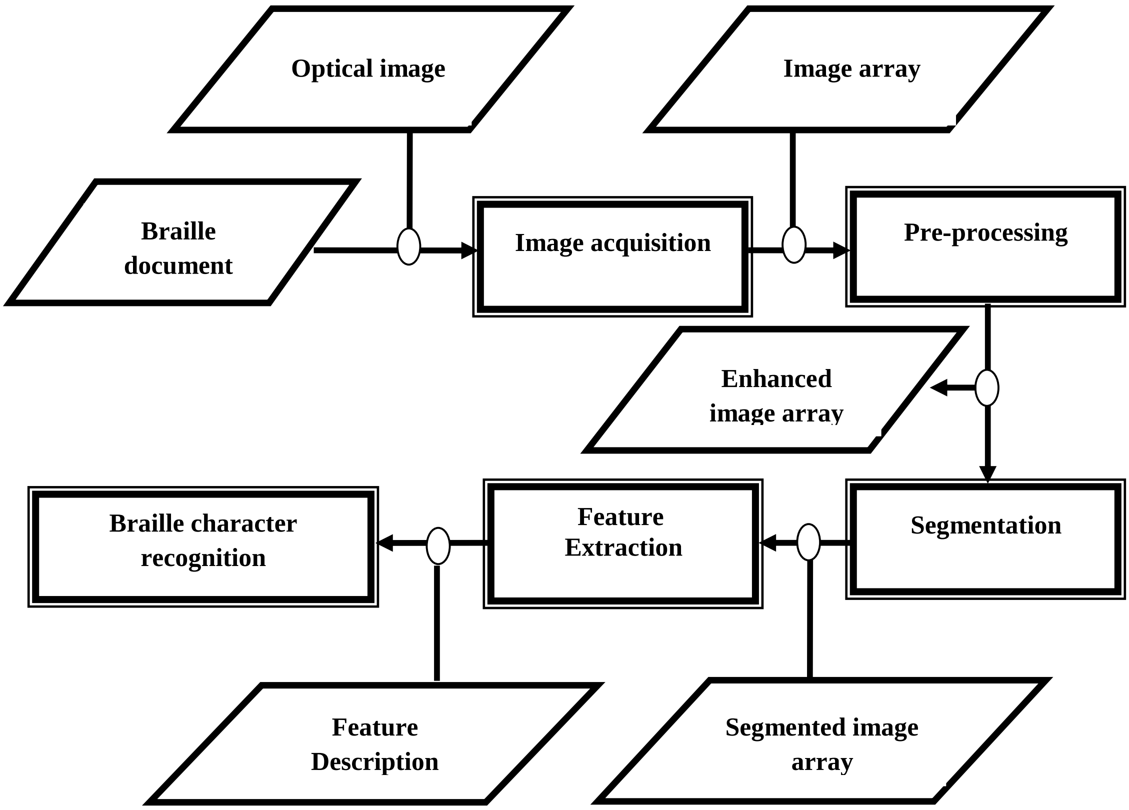

Figure 2: System overview

where R, G, and B are the Red, Green, and Blue channels of the RGB color model, respectively. Figure 3 shows a sample gray scale image.

Figure 3: Gray level image

The resulting image usually contain a darker frame around the borders that may disrupt subsequent processing steps, hence, we perform the standard image cropping as shown in Figure 4.

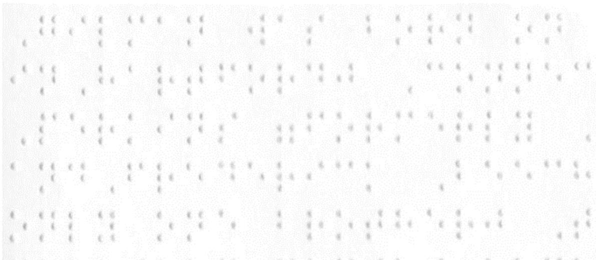

Figure 4: cropped image

3.3. Image enhancement

The main goal of image enhancement is to highlight specific image features [24]. The main image features in this paper are the dots and their relative location. We perform few image enhancement techniques to make these dots bold and easier to localize in subsequent steps. We perform the following image enhancement steps:

3.3.1. Noise Reduction

Image acquisition results in random noise that is reflected spatially all over the image. We eliminate this noise by an average filter. Figure 5 shows a sample image after noise reduction step.

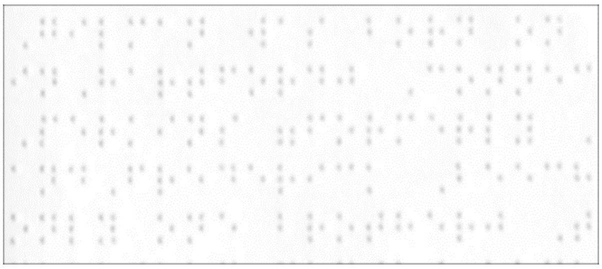

Figure 5: filtered image

3.3.2. Contrast Enhancement

Because of the uncontrolled scanning conditions where we collect our real data, image brightness vary and usually impact the quality of the dots and their discrimination. We perform contrast enhancement to concentrate on the intensity range around the dots' intensity levels. Figure 6 shows a sample image after contrast enhancement.

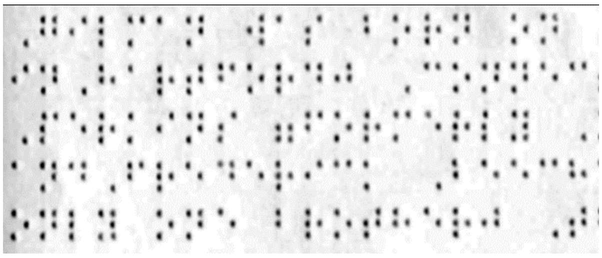

Figure 6: adjusted image

3.4. Image segmentation

We perform image segmentation to separate the desired dots from the background. To obtain well-defined dots, we perform image compliment as shown in Figure 7. Then image dilation to dilate the dots as shown in Figure 8.

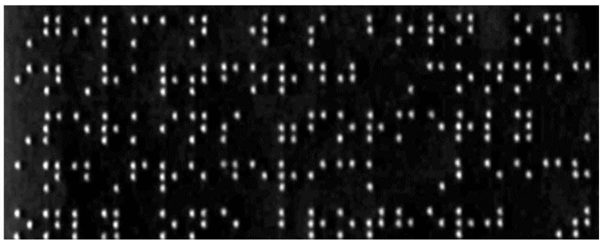

Figure 7: complemented image

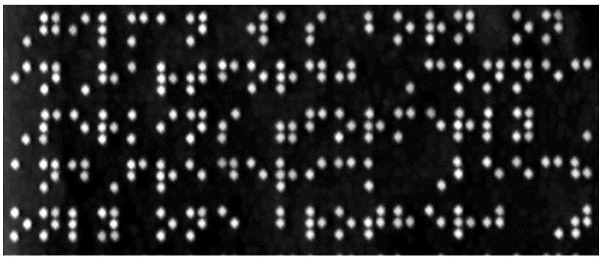

Figure 8: An image after dilation

At this stage, we perform a binarization step to separate the dots from the background as shown in Figure 9.

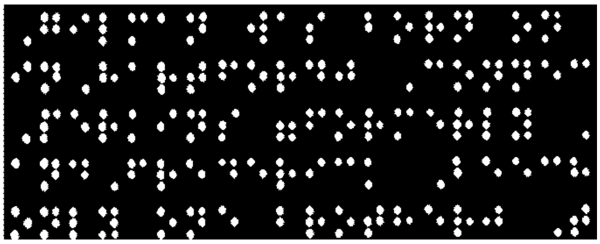

Figure 9: A Binary image

3.5. Feature Extraction

This is a major step where we extract the relevant information from the image for interpretation of the letters and words. We perform the following steps:

3.5.1. Compute Centroids of Dots

In this step we find the centroids of the dots in the image that are necessary for the alignment of the image. Computing the centroids includes the geometrical shape of the area which is a circle. Computing the centroid includes both C_x and C_y coordinates [25]:

$$C_x = \frac{1}{A}\int x\,dA \quad (2)$$

$$C_y = \frac{1}{A}\int y\,dA \quad (3)$$

$$A = \int f(x)\,dx \quad (4)$$

where C_x and C_y are the x and y coordinates of the centroid of the shape f, and A is the area of the shape function f.

For a circle shape f, a dot, the area (A) is the number of pixels within the dot (n), C_x is the average x-location of the pixels within the dot; C_y is the average y-location of the pixels within the dot. Hence, we compute the centroid x- and y-coordinate as follows:

$$C_x = \frac{1}{n}\sum_i^n X_i \quad (5)$$

$$C_y = \frac{1}{n}\sum_i^n Y_i \quad (6)$$

where the coordinates of the centroid are defined by (C_x, C_y) for each dot, and n is the number of pixels in each dot (area of the dot).

3.5.2. Braille Dot Alignment

This is an important step where we align the coordinates of the digitized page with Braille coordinate system. During the document digitization, we noticed some rotational angle in the document due to inaccurate document digitization. We apply an affinity operation by rotating the centroids so that both the vertical and horizontal axes are aligned between the page and the Braille dots.

We compute the rotation angle by computing the resulting angle between the upper edge of the image and the line formed by a series (minimum two) of the detected Braille dots at the same level. Given two points on the upper edge of the document (image) as ($x1_{edge}$, $y1_{edge}$) and ($x2_{edge}$, $y2_{edge}$), and another two centroids of Braille dots that are on the same level ($x1_{dot}$, $y1_{dot}$) and ($x2_{dot}$, $y2_{dot}$), then we obtain the two lines d_{edge} and d_{dot}:

$$d_{edge} = \sqrt{(x1_{edge} - x2_{edge})^2 + (y1_{edge} - y2_{edge})^2} \quad (7)$$

$$d_{dot} = \sqrt{(x1_{dot} - x2_{dot})^2 + (y1_{dot} - y2_{dot})^2} \quad (8)$$

Then, we compute the angle θ by the inverse tangent:

$$\theta = \tan^{-1}\frac{d_{dot}}{d_{edge}} \quad (9)$$

After the alignment we perform image filling to make sure that the dots are bold, sharp, and clear for subsequent steps as shown in Figure 10.

3.6. Braille cells recognition

This step is a major step to compose meaningful letters. It aims at grouping the dots based on the location information to obtain letters and words. The optimal methodology to perform this recognition is to rely on the

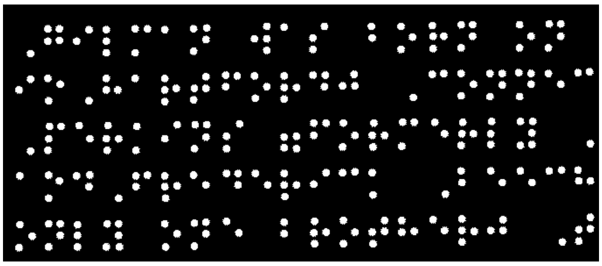

Figure 10: the image after drawing and filling circles

standard Braille measurements discussed earlier in this paper. This method ensures we recognize standard letters and words. These standard measures were shown in Figure 1. Figure 11 shows a sample image after the character recognition step which based on previously learned measurements.

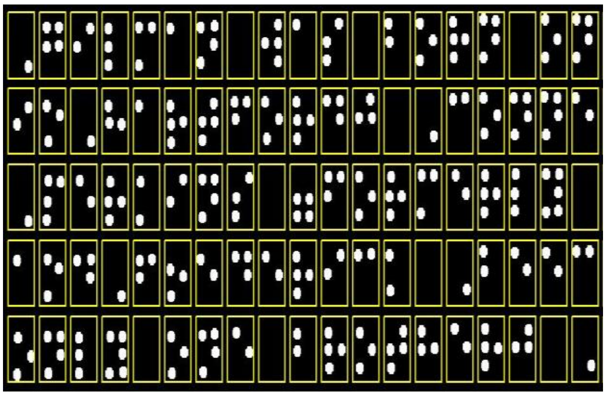

Figure 11: the image after Recognition of Braille characters.

4. Dataset and Preparation

We collect all the Braille documents and digitize them from one library. These documents are generic and not specifically designed for our system. Moreover, this data has great variability in sizes, people who printed them, and the conditions of their composition. We obtained twenty Braille documents with 355 – 434 Braille characters each. Hence, it is large and diverse enough dataset for our Braille recognition system testing and validation.

For digitizing purpose, we used an Epson perfection 1260 scanner with optical resolution 1200 x 1200 dpi. The system has been implemented and tested on a regular desktop with an Intel dual core and a 3G RAM. The hardware cost is very reasonably low for the benefit of the users of our system despite its robustness, high accuracy, and ease of use.

5. Experiments and Results

We created the ground truth manually where we detect each raised Braille point and label it. Then we apply our system on all the digitized documents. We measure the accuracy of detection of the Braille dots as follows:

$$Error = \frac{|\chi - \gamma|}{\chi} * 100\% \qquad (10)$$

where χ and γ are the number of dots in the ground truth and the number of correctly detected dots by our system, respectively. Table 1 shows the results for 11 images. Our systems' dots' detection accuracy ranges between 94.39% up to 99.76%.

On the other hand, we compute an accuracy measure based on the correctly detected words:

$$WordAccuracy = \frac{C}{N} * 100\% \qquad (11)$$

where C is the number of correctly detected words using our system and N is the total number of words in the documents.

In most available literature, they use expensive scanners while we use available cheap scanners. Yet, we achieve high accuracy and comparable error rates. Cost is crucial in these systems because of the limited resources visually impaired student and people have.

On the other hand, few techniques utilized more complicated machine learning methods to achieve the same task. However, most these techniques that have been used in the literature do not exceed the accuracy in our proposed system.

6. Conclusion

Braille system offers a way of communication for visually impaired people including writing and reading.

Table 1: Experimental results

Cases	Number of dots before processing	Number of dots after processing	Accuracy	Error
1	1034	1004	97.09	2.90
2	1070	1010	94.39	5.61
3	988	948	95.95	4.05
4	992	981	98.89	1.11
5	930	902	96.98	3.01
6	814	809	99.39	0.61
7	1065	1038	97.46	2.54
8	992	980	98.79	1.21
9	836	834	99.76	0.24
10	1048	1027	97.99	2.003
11	987	973	98.58	1.42

However, few research efforts have been conducted to help this important community for the advancement of cultures and societies. There are few techniques in the literature that transform the normal text to Braille script. However, little to not major efforts exist in the literature that perform the reverse operation which converts a Braille document back to a written text to quickly allow visually impaired people to communicate with non-visually impaired people via emails or similar communication technology.

In this paper, we present an extensive review of the literature with emphasis on the more recent research in development of Braille Recognition systems. We found a high similarity in most research efforts. However, there is a great variability in used data and the expenses required for the development.

In our proposed system, we used a set of coordinated methods to build an accurate, robust method to convert Braille documents to written text. Yet, our system is affordable and does not have to buy the expensive specialized scanners to utilize our system. We aim at positively impacting the society by increasing the communication with this important and effective community from each society.

In our system, we performed a set of sequentially designed engineering steps to produce a readable text out of the Braille document. It includes document digitization, image preprocessing, image enhancement, image segmentation, image alignment and image recognition. Our system's recognition ability ranges between 94% and 99%. We strongly believe that our proposed system has a great potential for daily use to enhance communication with this important and effective population in each society.

References

[1] AbdulMalik Al-Salman, Yosef AlOhali, Mohammed AlKanhal, and Abdullah AlRajih (2007), "An Arabic Optical Braille Recognition System", Proceedings of the International Conference in Information and Communication Technology & Accessibility (ICTA'07), Tunisia, pp. 81-86.

[2] J. Dubos, M. Benjelloun (1988), "Image Processing Techniques to Perform An Autonomous System to Translate Relief Braille into Black-Ink, Called: Lectobraille", Proceedings of the International Conference on IEEE Engineering in Medicine & Biology Society, LA, USA, Vol. 4, pp. 1584-1585

[3] Zhenfei Tai, Samuel Cheng, and Pramode Verma (2008), "Braille Document Parameters Estimation for Optical Character Recognition", Lecture Notes in Computer Science, Vol. 5359, pp. 905-914.

[4] Rawan Ismail Zaghloul and Tomader Jameel Bani-Ata (2011), "Braille Recognition System – With a Case Study Arabic Braille Documents", European Journal of Scientific Research, Vol. 62 pp. 116-122.

[5] T. Yoshida, A. Ohya and S. Yuta (2000), "Braille Block Detection for Autonomous Mobile Robot Navigation", Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol. 1, pp. 633-638.

[6] Néstor Falcón, Carlos M. Travieso, Jesús B. Alonso, and Miguel A. Ferrer (2005), "Image Processing Techniques for Braille Writing Recognition", Lecture Notes in Computer Science, Vol. 3643, pp. 379-385.

[7] A. Antonacopoulos and D. Bridson (2004), "A Robust Braille Recognition System", Lecture Notes in Computer Science, Vol. 3163, pp. 533-545.

[8] Saad D. Al-Shamma, and Sami Fathi (2010), "Arabic Braille Recognition and Transcription into Text and Voice", Proceedings of Biomedical Engineering Conference (CIBEC), Cairo, Egypt, pp. 227-231.

[9] Jan Mennens, Luc Van Tichelen, Guido Francois, and Jan J. Engelen (1994), "optical recognition of Braille writing using standard equipment", IEEE Transactions on Neural Systems and Rehabilitation Engineering, Vol. 2, No. 4, pp. 207-212.

[10] Marwa Abdelmonem, M. El-Hoseiny, Asmaa Ali, Karim Emara, Habiba Abdel Hafez, and Asmaa Gamal (2009), "Dynamic Optical Braille Recognition (OBR) System", Proceedings of the International Conference on Image Processing, Computer Vision, & Pattern Recognition (IPCV), Las Vegas, Nevada, USA, pp. 779-786.

[11] Zainab Authman, and Zamen Jebr (2012), "Arabic Braille scripts recognition and translation using image processing techniques", Journal of College of Education, Vol. 2, pp. 18-26.

[12] J. Bhattacharya, and S. Majumder (2011), "Braille Character Recognition Using Generalized Feature Vector Approach", Proceedings of the International Conference on Information Processing (ICIP), Bangalore, India, Vol. 157, pp. 171-180.

[13] R.T. Ritchings, A. Antonacopoulos and D. Drakopoulos (1995), "Analysis of Scanned Braille Documents", In A. Dengel and S. Marinai (Eds.), Document Analysis Systems, pp. 413-421.

[14] P. Blenkhorn (1997), "A system for converting braille into print", IEEE Transactions on Rehabilitation Engineering, Vol. 3, No. 2, pp. 215-221.

[15] C. Ng, V. Ng, and Y. Lau (1999), "Regular Feature Extraction for Recognition of Braille", Proceedings of the International Conference on Computational Intelligence and Multimedia Applications, New Delhi, India, pp. 302-306.

[16] Lisa Wong, Waleed Abdulla, and Stephan Hussmann (2004), "A Software Algorithm Prototype for Optical Recognition of Embossed Braille", Proceedings of the International Conference on Pattern Recognition (ICPR), Cambridge, UK, Vol. 2, pp. 586-589.

[17] X. Hermida, A. Corbacho Rodriguez, and F. Martin Rodriguez (1996), "A Braille O.C.R. for Blind People", Proceedings of the International Conference on Signal Processing Applications & Technology (ICSPAT 96), Vol. 1,. Boston, USA.

[18] Y. Mihara, A. Sugimoto, E. Shibayama, and S., Takahashi (2005), "An Interactive Braille-Recognition System for the Visually Impaired Based on a Portable Camera", Proceedings of CHI Extended Abstracts on Human Factors in Computing Systems, pp. 1653-1656.

[19] Y. Oyama, T. Tajima, and H. Koga (1997), "Character Recognition of Mixed Convex-Concave Braille Points and Legibility of Deteriorated Braille Points", System and Computer in Japan, Vol. 28, No. 2, pp. 44-53.

[20] Iain Murray, and Andrew Pasquale (2006), "A Portable Device for the Translation of Braille to Literary Text", Proceedings of ACM Conference on Assistive Technologies (ASSETS), pp. 231-232.

[21] Minghu Jiang, Xiaoyan Zhu, Georges Gielen, Elliott Drábek, Ying Xia, Gang Tan, and Ta Bao (2002), "Braille to print translations for Chinese", Information and Software Technology, Vol. 44, No. 2, pp. 91-100.

[22] Vidyashankar, G. Hemantha Kumar, and P. Shivakumara (2004), "Rotation Invariant Histogram Features For Recognition Of Braille Symbols", Proceedings of the International Conference on Cognitive Systems (ICCS '04), New Delhi, India, pp. 32.

[23] Avinash Chaudhary, Pardeep Garg, Arjun Agarwal (2012), "Using Rotation Method for Removal of Misalignment of Scanned Braille Pattern", International Journal of Advancements in Electronics and Electrical Engineering (IJAEEE), Vol. 1, No. 2, pp. 145-149.

[24] R. C. Gonzalez, R. E. Woods, and S. L. Eddins (2004), "Digital Image Processing using MATLAB", Pearson Education.

[25] Nathan Altshiller-Court (2007), "College Geometry", Dover Publications, (orig. Barnes & Noble 1952), pp.70-71.

[26] Braille Authority of North America (2006), "Capitalization Style for the Word "braille"", www.brailleauthority.org