Trends And Technologies In Optical Braille Recognition

AbdulMalik S. Al-Salman, Yosef AlOhali, and Layla O. Al-Abdulkarim
Computer Science Dept., King Saud University, Saudi Arabia
{salman, yousef}@ccis.ksu.edu.sa

ABSTRACT

Due to the rising importance of Optical Braille Recognition (OBR) around the world, a number of such systems have emerged. Their significance lies in the high demand for such systems in a number of facilities including printing houses, or even for the use by visually impaired individuals. This field has witnessed a number of efforts that have contributed to its advancements during the past few years. In this paper, we discuss the main concepts related to OBR systems; list the work of different researchers with respect to the main areas of an OBR system, such as pre-processing, dot extraction, and classification. Finally, we conclude with a comparison among results of the different systems technologies.

Key Words: Braille, optical recognition, blind, visually impaired, image processing

1. Introduction

Braille is a writing system that enables visually people to read and write through touch using a series of raised dots to be read with their fingers. Braille contractions representing groups of letters or whole words that appear frequently in a language. This is usually referred to as Grade 2 Braille. The use of contractions permits faster Braille reading and helps reduce the size of Braille books, making them somewhat less cumbersome. Each Braille character or "cell" is made of 6 dots arranged in a rectangle comprising 2 columns of 3 dots each as it can be seen in Figure 1. A dot may be raised at any of the 6 positions, or any combination. Counting the space, in which no dot is raised, there

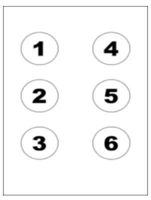

Figure 1. Braille Cell.

are 64 such combinations (that is $2^6 = 64$).
The dimensions of a Braille dot have been set according to the tactile resolution of the fingertips of person. The horizontal and vertical distance between dots in a character, the distance between cells representing a word and the inter-line distance are also specified by the Library of Congress. Dot height is approximately 0.02 inches (0.5 mm); the horizontal and vertical spacing between dot centers within a Braille cell is approximately 0.1 inches (2.5 mm); the blank space between dots on adjacent cells is approximately 0.15 inches (3.75 mm) horizontally and 0.2 inches (5.0 mm) vertically. A standard Braille page is 11 inches by 11.5 inches and typically has a maximum of 40 to 43 Braille cells per line and 25 lines. Braille has been adapted to write many different languages including Arabic and is also used for musical and mathematical notation. Note that both Arabic and English Braille are read from left-to-right.
OBR is a computer-based system that automats the process of acquiring and processing images of Braille documents. It converts a Braille embossed symbols into natural language characters. OBR system consists of several basic modules including: image processing, dot localization & segmentation, and dot recognition & conversion. Usage of OBR system offers many advantages especially when it comes to reproduction of Braille documents and when people who lack Braille reading skills need to know the contents of such documents. To ensure its usefulness, an OBR system must satisfy a number of

specifications, for example the average processing time should be reasonable. Furthermore, the system should be resistant to any kind of disturbance that may occur in images. Among issues that must be taken into consideration when implementing an OBR system are factors that negatively influence the identification process, such as lighting conditions, page placement, and page movement [6]. In this paper, we discuss different acquisition techniques of images of Braille in Section 2. Section 3 presents a number of obstacles and problematic issues facing developers. In Section 4 several design constrains are given, whereas Section 5 discusses image-preprocessing techniques. Section 6 introduces dot detection and extraction, while Section 7 discusses cell recognition and classification. Optimal resolution of input images is discussed in Section 8. In Section 9, test results and performance of the different systems are compared. Finally, Section 10 states the conclusion.

2. Image acquisition Techniques

The first step of any pattern recognition system is acquiring pattern data, basically to be classified into its respective pattern classes. In OBR systems data is provided to the system in the form of images of Braille embossed pages. The process of acquiring these images digitally can be achieved using a number of different techniques. Table 1 provides a comparison of image acquisition techniques employed by different OBR systems through out the literature.

3. Problematic Issues

A number of problems and issues must be taken into consideration to overcome challenges and avoid their negative influences on the recognition process. Deformation is one of the prominent problems that are usually introduced during the embossing process. Such deformation may be related to the shape of a single dot such as: degradation of the dots, variation in the space between dots/cells, and fading shade patterns in double-sided Braille document. In addition, deformation can be related to the adjustment of cells across the page, such as: page movement while typing text with a Braille typewriter.

More difficulties arise when attempting to recognize and convert text that falls within a certain context, such as:

Unit abbreviations, e.g., 20 m, 44 yds.

Initials in names, e.g., Mr. K Smith.

Ambiguous Braille signs.

Other obstacles and challenges that may also face designers of OBR systems also include: variation in paper colors and textures including blemishes, variation in background reflectance between documents, unknown size of the pages being read …etc.

Table 1: Various Image Acquisition Techniques.

Ref	Technique	Description
[4]	Scanner	AGFA HORIZON Laser scanner –A3 paper size
	Vidicon	---
	CCD Camera	---
[3]	Twain Standard	Used to support different scanners such as HP, Logitech, Caere and Aldus
[2]	Digital Camera	JVC Charged Couple Device mounted on a movable arm gave fast and simple capture of high resolution images.
[7]	Scanner	CanonScan FB320P
[9]	Scanner	---
[10]	Hand held scanner	Hand held scanner to capture real time Braille images via a 128-pixel CCD (Charge Coupled Device) array
[11]	Scanner	A special scanner to digitize Braille text. An air-coupled focused piezoelectric transducers, produced and characterized for the Braille digitization
[12]	CCD Camera	CCD camera and a laser fan beam projector are employed as sensors.

4. Design Constraints

Upon the design of an OBR system, there exist a number of constraints and considerations that must be taken into account. Such considerations are related to the quality of the system functionality that in turn affects the overall system performance. Among them, illumination is very crucial matter; it must be uniform over the entire page. In reality this problem was tackled by Neovision in their OBR system by using a yellow correcting filter which is placed beneath the scanned page on a flatbed scanner The filter's color was selected so the effect is balanced for both white and natural color (brown, yellow) paper. According to Neovision this effect cannot be done via software solutions for many scanners [8].

5. Image Preprocessing

The algorithms used for image processing differ from one system to another depending on the classification approach. An early effort was presented by Mennens et al. in 1993 [4]. They addressed the problem of false shadows in the image caused by imperfectly flat Braille pages. This frequently happens due to tension in the paper's surface. Authors suggest subtracting a locally averaged image from the original image. Next, authors addressed the problem of cell alignment across the page. The image is divided into a group of sub-images, each possibly holding a Braille cell. Finally, the authors apply skew correction using vertical projection. Reducing resolution in the horizontal direction speeds up the de-skewing process significantly.

Hermida's et al. [3] system, published later in 1996, employed thresholding as a pre-processing method. Analysis of scanned images of Braille dots revealed that a recto (convex) dot consists of a couple of spots, a bright spot above a dark one, while the reverse pattern is true for verso (concave) dots (Figure 2). Based on this concept their algorithm converts a digital image of a scanned Braille page into one consisting mainly of black and white spots denoting the dots. The thresholds used were adaptively calculated from the luminance histogram of the input image. They have initially set the threshold to 5% above/below the total area of the histogram.

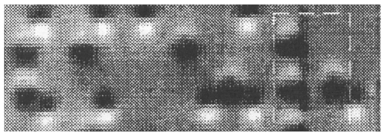

Figure 2: Recto and Verso Braille dots [4].

In 1995, Hentzschel and Blenkhorn [2] presented an OBR system based on twin shadows approach. They subtract two images of the same Braille page, where each image is taken under different illumination conditions. This helps eliminate blemish and noise in images caused by the texture of the paper. Input images are captured under carefully set conditions. Furthermore, a matt black background was used to eliminate secular reflections in the image and to help define the edge of the page. The image-processing module encompassed a variety of routines, each serving a different and crucial purpose. First, images had to be enhanced using random-noise reduction filter to avoid undesired emphasis of noise that would result after applying morphology on the image. The main reason for using morphology in the form of adaptive dilation is to accentuate and enlarge Braille dots for easier recognition. To avoid the gradual shading problem caused by single side illuminated page, the image is binarized using variance threshold. An iterative process is applied to find the optimal threshold. A threshold is said to be optimal if it maximizes the separation between gray level modals. The next image preprocessing module is image differencing, where the two input images are subtracted from one another resulting in a third image which supposedly consists of two shadows per dot, each in a different direction; that is left and right.

In [9], preprocessing consists of two operations: noise filtering and edge enhancement. Noise filtering is achieved via a low-pass Gaussian filter. Edge detection is achieved using convolution Sobel kernels according to the following equation:

Output=|convolute (input, X)|+|convolute (input, Y)|

To deal with different paper qualities, authors of [10] obtain average reference level from areas that contain only reflection from blank paper. Automatic contrast with respect to the backing medium is then computed.

In [13], authors described a number of pre-processing operations including spatial filtering, median filtering, erosion and dilatation that permit extracting contrasted relief from the bottom. They used the following polynomial filter:

$$\mathbf{Z = f(X,Y)=b_1+b_2*X+b_3*X^2}$$

Where b_i is estimated based on the luminance of the median pixel on a 1x5 neighborhood.

6. Dot Detection and Extraction

The approach adopted in [4] for extracting Braille dots from an image is based on several assumptions. Dots are assumed to be located on an intersection of orthogonal grid. A single dot is supposedly represented by two gray level intensities: a light area right above a dark one (as shown in Figure 1). Furthermore, they assume that the distance between the center of the dark and light areas is five pixels. Their system is designed to recognize a double-sided Braille pages. The algorithm starts by applying a grid over the image and then producing quantized image consisting of three gray levels based on Gaussian assumption. Next, a vertical mask is applied to locate center of the dots. Naturally, the vertical size of the mask corresponds to the initial assumption about the distance between bright and dark areas centers. Applying the mask results in an image which consists of five different levels (values): +2, -2, 1, -1, and 0 denoting recto core regions, verso core regions, recto side-lobe, verso side-lobe, and background, respectively. This approach has a disadvantage of producing false core regions if two dots are vertically aligned. Grids must then be searched to locate Braille dots by generating a vertical histogram of rows having 5 bins (-2, -1,0, 1, 2). Next, a horizontal histogram is generated in a similar fashion considering rows where Braille dots have been previously identified. This process is followed by searching horizontal lines again to locate dots. This approach has the advantage of eliminating false core regions caused by vertically neighboring dots. Since the gridlines were calculated separately for each sub-image, they must be regrouped in strips before starting to search for all atoms (atom is a group of lines that belong to a single Braille cell). This is achieved using an algorithm developed in [4], which builds atoms by restoring the vertical and horizontal gridlines. The algorithm estimates lines' locations based on distances calculated on local bases, as the algorithm learns from local conditions. When all atoms are found, minimal local correlation routine is run to check whether a Braille dot is found on the intersection of two lines.

The localization and extraction algorithm developed by Hermida et al. [3] takes as input a quantized image consisting of couples of white and black spots, where each couple denotes a single Braille dot. The algorithm starts by computing the centers of black and white spots that are referred to and dealt with as black and white points. Next, the routine starts looking for white points above black ones denoting recto dots and marks them as mountain-point. Similar process is applied to denote verso dots. Though this method is both easy and quick, it suffers from a couple of problems as some legitimate points are lost, and false ones are produced. This happens when a mountain and a valley are very near to each other, and are joined after quantization. Mesh detection is applied to solve these two problems. A mesh is the group of lines and columns present in the entire input text rather than a single cell. The number of points in a mesh depends entirely on the number of cells in the input text. If the text consisted of N characters, then the mesh has Nx6 dots. The algorithm begins by selecting a starting point, from which the algorithm moves a standard distance to construct the mesh. Selection of start point is done by using maximal count methods which tests ten points, all of which are located in the top left corner. Each point is tested as the algorithm tries to move vertically a standard distance downward, and counts the number of detected points. Based on the maximum of counts for each point, the appropriate starting point is chosen. The relative position can be any one of the six possible positions within a Braille cell. The entire mesh is constructed in a similar fashion. The algorithm scans one line after another, when a single cell mesh is detected it starts to look for the rest of characters on the same line by trying to locate points within a certain distance from detected dots.

The dot localization and extraction technique used by Hentzschel et al. in [2] consists of two steps; the first is image registration, which results in setting the page position & orientation, followed by line detection using row-counting. Though useful, applying row-counting on Braille page introduces a few complications that require more computation than it normally would for ordinary character recognition. This is most evident when taking a closer look at a line of ordinary text that consists of a single group of peaks, where the same is not true for a line of Braille text that can have 1, 2, or 3 groups of such peaks depending on the characters residing on the line. Therefore, they built their system on the assumption that Braille's cell height to width ratio is 2:1 rather than assuming fixed-height lines. The second step of dot localization and extraction is character segmentation, which requires generation of a profile for each line of text using column-counting method. The profile reveals the types of frequency of peaks that reflect occurrences of characters within the line. A character is said to be present if there is at least one peak or at most four (two sets of shadows per column). Inter-character and Intra-character spacing are two important properties that must be taken into consideration. Despite the fact that inter-character spaces are always greater than intra-character spaces, sizes of both kinds vary depending on the characters present in the line. The final step of dot extraction is normalization, where each Braille character is represented by a 2x3 matrix. Each bit in the matrix denotes a dot in Braille cell (character). Deciding whether a bit value is either 0 or 1 is based on a threshold used to test 1/6 of the cell area. The threshold is adjusted to reflect the number of active dot in a Braille cell.

In 2004 Wong et al. [7] proposed an OBR system that is capable of recognizing a single-sided Braille page. The system preserves the format of the original document in the produced text file. Their algorithm is built around the concept of detecting half characters. The idea is to locate characters by detecting possible dot positions. The algorithm processes the image one row at a time producing a thresholded image consisting of white area denoting characters on a black background. When a row is found to contain characters, it is buffered for further processing which aims to determine positions of half-characters.

The dot detection module in the OBR system proposed by Oyama et al. [6] in 1997 is designed to detect both recto and verso Braille dots; that is detecting dots on both sides of the page. This was possible due to the difference in light reflectance between concave and convex dots. To determine the strength of the reflected light L_r (off the surface of Braille embossed page) they used the equation:

$$L_r = a\, L_i \, Sin\,(\theta_i + \theta_r)\, Cos\, \theta_r$$

Where L_i is the strength of the incident light, θ_i is angle of the incident light, θ_r is angle of the reflected light, and a is constant.

The above technique is designed in a way to detect dots on both sides of Braille page; that is convex and concave dots. They defined a three-state function of the strength of the reflected light L_r as follows:

$$F(x) = \begin{cases} +1 & if\ L_r(x) > t_h \\ 0 & if\ -t_l \le L_r(x) \le t_h \\ -1 & if\ L_r(x) > t_l \end{cases}$$

Where t_l and t_h are thresholds. g(x) denotes the correlation between f(x) at position x and f(x+δx) at position x.δ+x:

$$g(x) = f(x)\,f(x+dx) = \begin{cases} +1 & \text{if } f(x) = -1 \text{ and } f(x+dx) = +1 \\ -1 & \text{if } f(x) = +1 \text{ and } f(x+dx) = -1 \\ 0 & \text{otherwise} \end{cases}$$

The sign of g(x) determines whether a detected dot is a convex or concave. Dot extraction is then achieved by using a 2x3 convolution mask of the following form:

M_{11}	M_{12}
M_{21}	M_{22}
M_{31}	M_{32}

Unfortunately, the success of this method solely depends on whether or not a dot is found in M_{11}. Therefore, as a solution to this problem, the mask was enhanced taking the form:

W_{11}	W_{12}	W_{13}
W_{21}	W_{22}	W_{23}
W_{31}	W_{32}	W_{33}

M_a M_b

Where the sub-mask M_a={w_{11}, s_{12}, w_{21}, w_{22}, w_{31}, w_{32}}, and the sub-mask M_b={w_{12}, w_{13}, w_{22}, w_{23}, w_{32}, w_{33}}. Image is scanned from top left corner to the bottom right corner to locate Braille cells. When a dot is detected, it is decided to belong to either M_a or M_b [6]. Other points are suspended temporarily to be confirmed later based on the previously confirmed cells. Determining whether a dot belongs to M_a or M_b is based on the distance between the undetermined window and a previously determined one.

7. Recognition and Classification

The recognition module proposed in [1] is based on finite state approach. It performs both left and right context checking using matching algorithms. One of the greatest advantages of this system is the flexibility to convert Braille to any natural languages, depending on the provided conversion rules. The system consists of a number of finite states, a number of input classes, decision tables and conversion rules table. Though the system proved to be functional for U.K. and North American Braille codes, it suffers from a number of shortcomings, mainly with regard to the mishandling of certain signs and punctuation marks.

Authors of [2] and [4] did not elaborate on the techniques used for converting an extracted Braille cell into natural language characters. In [2], the classifier is based on "American Computer Braille" which is used by several printing houses and printers. [4] adopted binary Braille character sets as basis for their classifier, that is grouping the dots and representing each dot by a bit position.

The classifier presented by Hermida et al [3] is quite simple; it takes as input the image produced from the dot extraction module. Each dot is represented by a single bit (0 or 1) and then the Braille-to-ASCII conversion is accomplished.

The recognition module in Wong's et al. OBR system [7] is designed to work with quantized image resulting from the half-chars detection module, where white regions in the image indicate probable characters. The half-chars recognition module classifies half-chars to one of seven possible arrangements of dots within a single column excluding the possibility of an empty column (empty half character). The seven arrangements are shown in Figure 3.

Figure 3: The Seven dot arrangements

The classification process is carried out using a probabilistic neural network with a hidden layer consisting of seven radial basis function neurons. The output layer is a competitive network. After classifying half-chars, the respective positions are processed to locate the entire character. Determining the position and location of characters and half- characters is governed by standard spacing (inter/intra-character spacing).

For interpretation purposes, authors in [9] determine centriod distances between each dot and its four possible neighbors. Dots are then grouped in cells. Two standard templates are constructed to represent the front-face and back-face dots based on the boundary coordinates information and illumination characteristics.

In [10], authors applied a linear motion detector to capture any movement. Image is then processed in vertical slices. Each position is labeled using 3 pairs of 2 bits representing whether that position is bright, shadow or reference. After each slice capture, the system attempts to recognize the current potion of the image using a bit mask comparison with ideal mask.

In [13], authors decomposed recognition into four steps. First, they calculate mean value of inter/intra-character distances. Second, they mark by left or right the image mass axis after consulting a previously built knowledge base. Third, they mark the inter-axis in two opposite directions. Finally, they interpolate between confirmed mass axes to generate fictive axis. Similar process is applied along vertical axis.

8. Optimal Resolutions

Resolution of images affects the efficiency of different OBR systems in terms of detection and recognition. Throughout the literature, researchers usually design their systems around a certain level of resolution to ensure the best performance. In Table 2 gives a comparison of different resolutions adopted by some of the authors.

Table 2: Comparison of Different Image Resolutions.

Ref	**Resolution**
[4]	Used 120 dpi, Recommended range: 80 – 200 dpi, Over 200 dpi gives too much detail, which confuses the algorithm.
[2]	760 * 575 pixels
[7]	300 dpi
[9]	512x512 pixels with 8-bit gray scale (11x11 inch / page)

9. Test Results and Performance

Table 3 gives a comparison among results of different tests conducted by researchers throughout the literature. The table indicates effectiveness of each method. Table 4 provides similar comparison in terms of execution speed.

Table 3: Effectiveness of different methods.

Ref	Description	Results	
[4]	Tests were carried out on sets of different textures and colors of single/double sided Braille paper. Each set contained equal number of samples.	Group 1: (4 sets) Converted with no errors.	
		Group 2: (7 sets) Average error rate: 0.25% per character. Error caused by defects or stains on the paper's surface.	
		Group 3: (1 set) Error rate not specified (presumed high) Caused by particular deformation of the character matrix that the program could not cope with.	
[2]	Light axis angle	Detection	Recognition
	0	100%	92%
	45	60%	13%
[7]	Half-character method	100%	99.5%
[9]	Test is conducted on single-sided and double-sided specimen	100% (on single-sided specimen)	97% (on double-sided specimen)

Table 4: Execution speed of different methods

Paper	**Exe. Rate Sec/char**	**Page Sides**	**Platform**
[4]	60	Single	Macintosh IIfx
	80	Double	
[3]	40	Double	486DX2 PC
[7]	32.6	Single	AMD 2000+ CPU 256 MB RAM PC

10. Conclusion

In this paper, stages of a typical OBR systems and efforts of researchers in previous years were explored in a logical order. First the paper described preprocessing techniques used in OBR systems, such as thresholding that was exploited by most researchers. Next, the paper discussed Braille dot extraction modules that mainly consisted of localization segmentation sub-modules. Furthermore, it states problems and obstacles that face designers of OBR systems in relation to the nature of Braille. In addition, the paper illustrates the constraints that govern the implementation of different stages. It also presents test results and performance speed of different systems. From this comprehensive survey, we found no research conducted for the Arabic OBR; the authors are developing an Arabic OBR. Continuation of efforts and research in the area of OBR is inevitable since such systems have become an essential part of many Braille documents reproduction facilities and visually impaired people.

References

[1] Blenkhorn, P., "A System for Converting Braille into Print", IEEE transactions on rehabilitation engineering, Vol. 3, No. 2, June 1995.

[2] Hentzschel, T. W., and P. Blenkhorn, "An Optical Reading Systems for Embossed Braille Characters using a Twin Shadows Approach", Journal of Microcomputer Applications, pp. 341-345. 1995.

[3] Hermida, X. F., A. Corbacho Rodríguez, and F. Martín Rodríguez, "A Braille O.C.R. for Blind People", Proceedings of ICSPAT-96. Boston (U.S.A.). October, 1996.

[4] Mennens, J., L. Van Tichelen, G. Francois, and J. Engelen, "Optical Recognition of Braille Writing", IEEE, 1993.

[5] Mennens, J., L. Van Tichelen, G. Francois, and J. Engelen, "Optical Recognition of Braille Writing Using Standard Equipment", IEEE transactions of rehabilitation engineering, Vol. 2, No. 4, Dec. 1994.

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

[7] Wong, L., W. Abdulla, and S. Hussmann, "A Software Algorithm Prototype for Optical Recognition of Embossed Braille", the 17th conference of the International Conference in Pattern Recognition, Cambridge, UK, 23-26 August 2004.

[8] Private Communication: Radek Svoboda, Neovision, Prague, svoboda@neovision.cz, http://www.neovision.cz

[9] C. Ng and V. Lau, "Regular feature extraction for recognition of Braille", Third International Conference on Computational Intelligence and Multimedia Applications (ICCIMA'99), 1999.

[10] I. Dias, "A portable device for optically recognizing Braille-Part II: Software development", 7th Australian & Neazlan Intelligent Information Systems Conf., pp. 18-21, Perth, W. Australia, Nov 2001.

[11] T. Gomez, F. Montero and A. Pozas, "AIR-Coupled Ultrasonic Scanner for Braille", IEEE ultrasonic Symposium, pp. 591-594, 2001.

[12] T. Yoshida, A. Ohya and S. Yuta, "Braille Block Detection for Autonomous Mobile Robot Navigation", Proc. Of the 2000 IEEE/RSJ intl. Conf. on Intelligent Robots and Systems, pp. 633-638, 2000.

[13] J. Dubus, M. Benjelloun … etal, "Image Processing Techniques to Perform an Autonomous System to Translate Relief Braille into Black-Ink, Called: Lectobraille", IEEE Engineering in Medicine & Biology Society 10th annual Inter. Conf., 1988.

[14] Antonacopoulos, A., and D. Bridson, "A Robust Braille Recognition System", Proceedings of the IAPR International Workshop on Document Analysis Systems (DAS 2004), Florence, Italy, 8-10 September, 2004, Springer, pp. 533-545.

[15] Mihara, Y. , A., Sugimoto, E., Shibayama, S., Takahashi "An Interactive Braille-Recognition System for the Visually Impaired Based on a Portable Camera", CHI Extended Abstracts, pp. 1653-1656, 2005.

[16] Murray, I., and A. Pasquale, "A Portable Device for the Translation of Braille to Text", Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility Assets, October 2006.