Image Processing Techniques for Braille Writing Recognition

Conference Paper *in* Lecture Notes in Computer Science · February 2005

DOI: 10.1007/11556985_49 · Source: DBLP

4 authors, including:

Carlos M. Travieso
Universidad de Las Palmas de Gran Canaria
396 PUBLICATIONS **3,653** CITATIONS
SEE PROFILE

Jesús B. Alonso
Universidad de Las Palmas de Gran Canaria
286 PUBLICATIONS **3,332** CITATIONS
SEE PROFILE

Miguel A. Ferrer
Universidad de Las Palmas de Gran Canaria
300 PUBLICATIONS **3,891** CITATIONS
SEE PROFILE

Image Processing Techniques for Braille Writing Recognition

Néstor Falcón, Carlos M. Travieso, Jesús B. Alonso, and Miguel A. Ferrer

Dpto. de Señales y Comunicaciones, Universidad de Las Palmas de Gran Canaria
Campus de Tafira s/n, E-35017, Las Palmas de Gran Canaria, Spain
nesmofalcon@gmail.com, ctravieso@dsc.ulpgc.es

Abstract. In this paper we present the development of *BrailLector,* a system able to speak from Braille writing. By means of dynamic thresholding, adaptive Braille grid, recovery dots techniques and TTS software (Text-To-Speech), *BrailLector* translates Braille scanned images into normal text, and not only that, it speaks the translated text. *BrailLector* is a robust application with innovative thersholding and Braille grid creation algorithms which detects and read Braille characters with 99.9% of correct symbols and an error variance below 0.012. The conversion time is only 26 secs for double-sided documents by MATLAB programming language.

1 Introduction

Nowadays, lack or problem of vision has been an important obstacle to access to printed contents and to the information society. For this reason, some people have tried to achieve that blind people are able to access to the printed culture, for example Valentin Haüy and Luis Braille who have understood the importance of a communication code. Globally, an estimated 40 to 45 million people are blind and 135 million have low vision according to the World Health Organization (WHO) [1] and this number grows every year.

A Braille Optical Character Recognizer is interesting due to the following reasons;

- It is an excellent communication tool for sighted people (who do not know Braille) with the blind writing.
- It is a cheap alternative Braille to Braille copy machine instead of the current complex devices which use a combination of heat and vacuum to form Braille impressions.
- Braille writing is read using the finger so is necessary touch the document, for this reason the book after many readings is possible has been deteriorated.
- It is interesting to store a lot of document of blind authors which were written in Braille and were never converted to digital information.
- *Braillector* offers a better integration of blind people to the "information society".

Since the most part of Braille books are written with two kinds of points, we will have to distinguish between each one. One kind is like "mountains" and the other is as little "valleys", hence the finger of the blind person who reads the text only detects

R. Moreno Díaz et al. (Eds.): EUROCAST 2005, LNCS 3643, pp. 379 – 385, 2005.

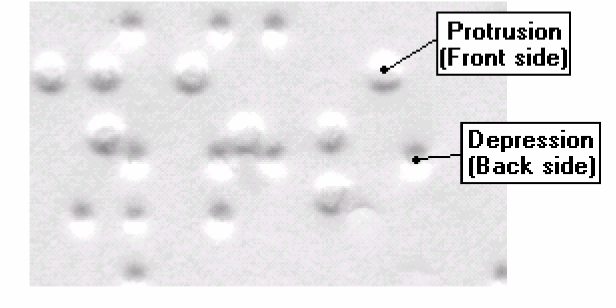

Fig. 1. Protrusions and Depressions on a Braille sheet

the protrusions, while depressions are points written to be read from the other side of the sheet. If we are able to distinguish between each point we will have a big advantage, we will be able to recognize the two sides with only one scan.

The structure of this paper is the next: In the next text we will describe the characteristics of the database created. After, image processing techniques used for dots detection and recovering will be described. Then, we will explain the conversion from Braille text to standard text. Conclusions close this paper.

2 Database

A big database has been created in order to check the global system with as many characters as we could. This database provides single and double-sided documents, which have dots in one or both sides of the sheet respectively. The number of characters in this database ensures the correct testing of the developed system and a good analysis of the error variance. The next table gives a full explanation of this database.

Table 1. Database created

Braille sheets	26
Total number of characters	30862
Mean number of character per sheet	1235
Digital format	Gray scale
Resolution	100 dpi (horizontal and vertical)
Image size	1360 Kbytes
Image format	Bitmap (‘bmp’)
Braille type	Double sided – grade 1
Document size	29.5 cm. (horizontal) x 30.5 cm. (vertical)

The mechanism used for image acquisition has been a flat-bed scanner instead of a digital camera because it is a cheap alternative which can be used for so many other applications and it is easy and quick to use. The system is able to work with images of different resolution than 100 dpi since it uses interpolation methods to resize the input image.

3 Image Processing Techniques

The next image represents the global blocks of image processing techniques;

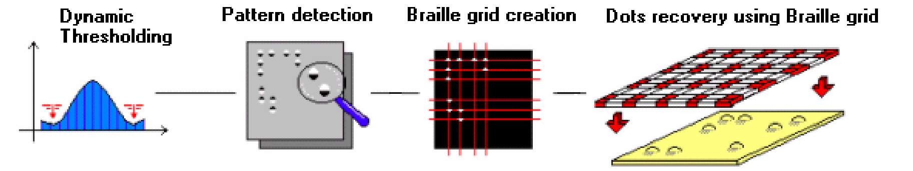

Fig. 2. Braille Image Processing

The different steps of the scanned image for its translation are;

1. An innovative thresholding method has been used to extract the useful information of Braille images instead of traditional methods [2]. Once we know the optimum area of Braille spots (it reduces the number of wrong symbols detected), an iterative algorithm looks for the best threshold according to these areas of Braille dots. This area criterion for thresholds selection offers an accuracy way to get the optimum levels to separate black, white and grey.

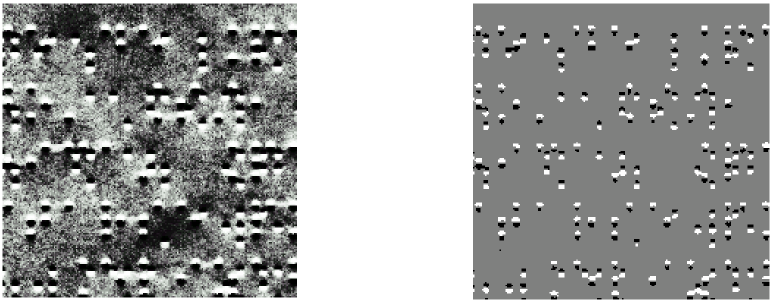

Fig. 3. Automatic threshold selection process

After this first step, no useful information has been rejected and the image is now ready to be processed in the pattern detection block.

2. Once this primary process is done, we take advantage of the shadows which make dot patterns. As we have seen, these protrusions on a Braille sheet have a brilliant zone above and a dark zone below. The depressions have exactly the opposite pattern (these shadows are created by the skew angle of the light beam

in reflection scanners) so this difference will facilitate the separation between front and back side dots;

- o Moving white "islands" 4 pixels downwards and doing a logical "and" with black spots we will extract front side dots.
- o Moving white "islands" 4 pixels upwards and doing a logical "and" with black spots we will extract back side dots.

The goal of this secondary process is to separate each side of the document in different images. This algorithm consists of a "shift and overlap" process since it only moves the spots downwards or upwards and carries out a logical and. It is the fastest method we have tried because it avoids the sequential reading of the image matrix and it is very simple and efficient. In this stage, skew angle of the scanned document is detected by means of horizontal histogram and mass centers calculation and corrected rotating the original image.

3. Not all the dots are detected with overlapping process, some of them are missed, either because they are very small or their shadows do not overlap with only 4 pixels. For this reason, a new stage was added to the system: The Braille grid.

   An adaptive algorithm has been developed in order to make this mesh from the detected dots. The algorithm builds columns in a first stage: since distances between points are normalized [3], the process begins searching for groups of dots in the same vertical plane that respect these distances. Then it builds the columns according to the pattern of Braille columns adapting itself to the layout of the document. Thus, we get a flexible mesh that tolerates small differences between columns. The process for the rows is quite similar but in this case the pattern to search is a Braille row. Detected columns and rows will be arranged together to create the final structure. This grid is flexible and respects the layout of the original document which makes it suitable for copying Braille sheets without losing the format.

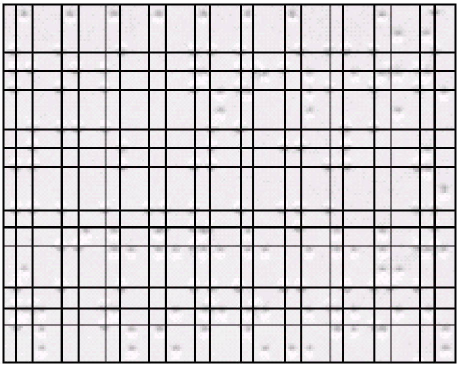

Fig. 4. Braille grid layout for frontal side

4. After mesh building, all valid Braille positions are known. Those intersections between rows and columns will define a valid position for a Braille dot. First, dilation techniques [4] are used to expand the search zone. Then, we fit this

image on the dots image after thresholding in order to check in detail those positions where dots were not found. In this point lays the intelligence of the global system, only potential positions of dots are checked; this means time saving and efficient search. Original dots will be recovered (they belong to correct Braille positions) and false dots will be discriminated as they are out of the valid places for a Braille dot. In the following figure, we show this effect on the recuperation of Braille dots.

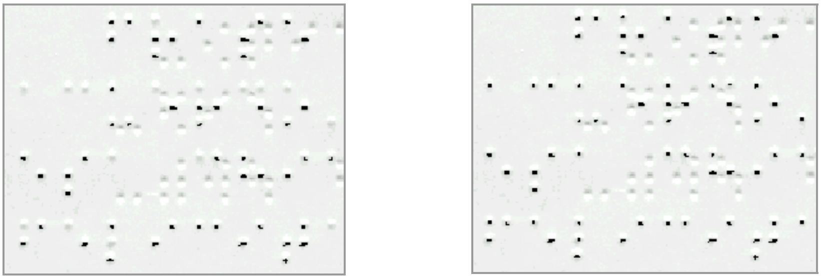

Fig. 5. Front side detected dots before and after the use of the recovered algorithm

4 From Braille to Normal Text

At this point all valid Braille dots have been detected in both sides of the document. This final image contains the Braille dots represented by spots, so now it is analyzed and text is segmented in rows and characters. For this segmentation process we will take advantage of the Braille mesh one more time since it marks all the positions of Braille dots. Every character will be converted into a binary number according to the active dots. The process consists of reading character by character and each one of the six positions that make the basic cell [5]. Hence we will have six possible values for each character (either raised or flat). This way of coding is simple and fast. It can be explained better by means of the next figure.

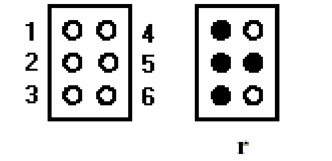

Fig. 6. Standard Braille Character (left) and ‘r’ symbol

In this way and looking at the previous example, ‘r’ symbol can be coded like ‘111010’ where each black dot is an active dot or a “mountain” on the paper and they become ‘1’ in the binary number. This way of Braille text binarization makes the global system independent of the language of the document and easily configurable for adding different alphabets. The output of this step will be a file with each character coded like a binary number; we will only have to translate each number for its equivalent letter in normal text to get the final output like a text file.

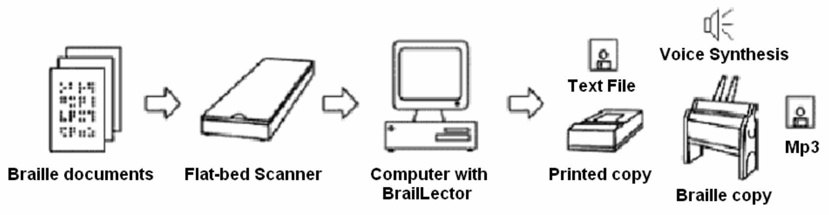

Fig. 7. Global translation process of Braille documents

This final output can be presented in different formats such as a text file, a new Braille printed copy, voice (by means of TTS software [6]) or even mp3 audio format.

5 Conclusions

In this paper we have explained the development of an automatic system for translating Braille text to normal text or voice. The global algorithm is very fast and robust. It has been divided in different modules for each part of the image processing. For achieving this system, dynamic thresholding and adaptive Braille grid has been used, adding some intelligence to the global process and making it able to detect dots in both sides of the document with only one scan. This process has an efficiency of 99.9% and it takes only 26 secs to translate a double-sided document improving all the main references found in the bibliography [7], [8], [9].

Acknowledgments

As with most works, many have provided help in the development of this research. Many thanks to Dr. Fatih Kurugollu, Lecturer of School of Computer Science at Queen´s University, Belfast, United Kingdom where a big part of this work was done. He provided invaluable help and clear ideas to solve many problems about image processing. He has my admiration for his patience and my deep appreciation for his assistance.

This work was supported by the research Project “Pi 2003/005” from Canary Autonomous Government (Spain).

References

1. http://www.who.int Active on April 1st 2005
2. N. Otsu. “A threshold selection method from gray-level histograms” In IEEE Transactions on Systems, Man, and Cybernetics, vol 9, no.1, pp 62-66. Jannuary 1979
3. Dubus J., Benjelloun M., Devlaminck V., Wauquier F., Altmayer P.: Image processing techniques to perform an autonomous system to translate relief Braille into black-ink, called: Lectobraille. In Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, (1988) 1584–1585.

4. Rafael C. González, Richard E. Woods "Digital Image Processing" Prentice Hall (2002)
5. http://www.fbraille.com.uy/alfabeto Active on April 1st 2005
6. http://www.textreader.net Active on April 1st 2005
7. Wong L., Abdulla W., Hussman S.: A Software Algorithm Prototype for Optical Recognition of Embossed Braille. The University of Auckland, New Zealand, (Technical Report) (2004).
8. Mennens J.,. Tichelen L. V, Francois G., Engelen J.: Optical recognition of braille writing using standard equipment. IEEE Transactions on Rehabilitation Engineering, 2(4) (1994) 207– 212.
9. Ng C., Ng V., Lau Y.: Regular feature extraction for recognition of Braille. In Proceedings of Third International Conference on Computational Intelligence and Multimedia Applications, (1999) 302-306.