ROBUSTIC ARABIC OPTICAL BRAILLE RECOGNITION SYSTEM

Marwa Abdelmoen, Asmaa Ali, Habiba Abdel Hafez, Asmaa Gmal, Mohammad Hamdy,
Karim Emara, Haythem El-Messiry, and Abdalbadeeh Salem
Ain Shams University, Faculty of computer and information science, Cairo, Egypt

ABSTRACT

Optical Braille Recognition (OBR) systems allow blind people to read volumes of typewritten documents with the help of ordinary flatbed scanners and OBR software. The method implemented looks at developing a dynamic system to recognize an image of Arabic Braille and then convert it into readable text. The system has been tested with a wide variety of scanned Braille documents, both single and double sided, produced with different mechanical methods and scanned with different scanners.

1. INTRODUCTION

Communication in the written form plays a very important role in the daily life for many people like in the educational purposes or taking notes etc.
But a problem is faced by the visually impaired or the blind People dealing with these kinds of written means of communication.
The most commonly used writing convention among visually impaired people is Braille.
Using the new technology, producing Braille documents is somehow easy; the remaining problem is how to computerize these Braille documents[6,7]. This is an important problem to be solved because of two reasons; first, there is a wealth of books and documents that only exist in Braille that, as with other rare/old documents, are deteriorating and must be reserved (digitized). Second, there is an everyday need for duplicating (the equivalent of photocopying) Braille documents and for translating Braille documents for use by non-Braille users.
Braille characters (cells) consist of six dots arranged in the shape of rectangle with two columns and three rows; this will generate 64 combinations that can contain all the alphabets and the punctuation marks.

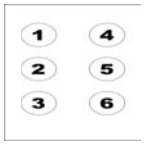

Fig. 1. Braille cell

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.
Braille documents [1] can be written in two different ways (grade 1, grade 2). In grade one there is a one to one correspondence to the ordinary characters, but in grade two (also called contracted, or literary Braille) there are conventions for representing whole strings of printed characters by a single Braille one.
The system based on a simple scanning process using an ordinary flatbed scanner, this also to be easy to be used by the blind or the visually impaired people or even people who may deal with Braille users. The paper is organized as follows: section1, introduction, section2, proposed methodology, section3, results and conclusion, and finally, future works.

2. PROPOSED METHDOLOGY

In this section a brief explanation of the proposed method. Figure 2, is system description to simplify all the steps of the algorithm.

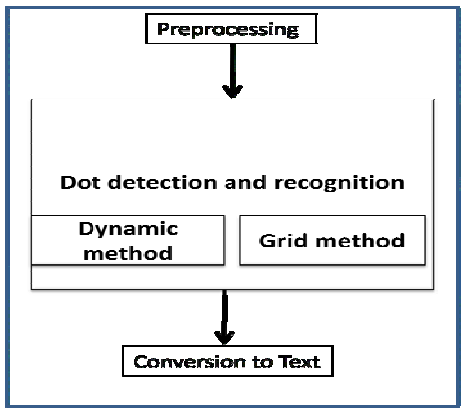

Fig. 2. System description

There are some steps must be done on the scanned image as preprocessing to be prepared for the basic steps
The Algorithm runs as follows:

2.1 Preprocessing

1- Converting the image to gray level [1]

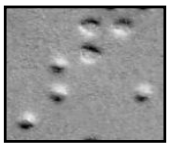

Fig. 3. image converted to gray

2- Image threshold [1]

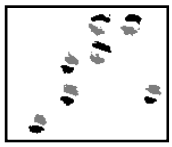

Fig. 4. After threshold

3- Applying flood fill algorithm with 4-neighbour method, this step not for enhance the output image but is done to be used in next phases to make some steps in the algorithm dynamic depend on the incoming image which differ in dot height and width flood fill calculate the average dot height and width to make the algorithm dynamic and not dependant on fixed image with fixed dot heights

2.2 Dot parts detection

After dividing the image into three colors (dark, bright, gray) as in figure 3, the next step is to determine the recto and verso dots.

In paper [2] this is done depending on static number of pixels to detect the dots in the image.
As most of Braille images - in common resolution- the dot height is 8 pixels; each dot is composed of a bright and a dark region with a small space between them. The implication is that if the bright region comes at top where the dark one comes at bottom then this is a recto dot, the contrary situation results in a verso dot.

Fig. 5. Recto (Gary dots) and Verso (White dots)

The introduced method uses dynamic technique to overcome being static one -as mentioned.

According to average height is calculated in the flood fill we can calculate average dot height using this equation:

AvgDotH= average dot height*(8/12)

And we use this calculated value as follows:
A vertical search for dot parts is performed on the array starting from top to bottom such that:

If pixel (1) + pixel (2) > 0 AND last tow pixels in the array of average dot height <0 then this column is part of a recto dot.

If pixel (1) + pixel (2) < 0 AND last two pixels in the array of average dot height >0 then this column is part of a verso dot.

In case of having any of the two conditions true we register that in the array and then go down with AvgDotH since this is the vertical space between two dots. At the end, we will have recto and verso dots as in Figure 10. Now we can separate them in two arrays.

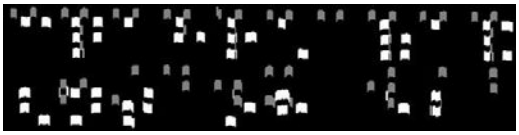

Figure 6: Output of dot part detection

After doing this the result will be recto and verso dots with some noise and overlapping areas so the next step which is the whole dot detection is considered to remove noise and overlapping and separate the recto and verso dots into two different buffers.

2.3 Whole parts detection

This step is for removing the noise and overlapping resulted from dot parts detection phase and for separating recto and verso dots into two different buffers to easily recognize the cells [1].

Fig. 7. Buffer contains recto dots only

Using the average dot height calculated from the flood fill we go over each column and row pixel by pixel as follows:

- A vertical search is performed on the array starting from up to down, then check if the value resulted from this (5 * AvgDotH/8) is the same color then it is considered one dot
- Doing this over all the image result in identifying the dots then,
- Counting the number of dots in each column in the image with width 1 pixel result in array contains number of dots in each column in the image which in sense will increase and decrease as illustrated in figure 11.
- This numbering under the curve indicate the number of dots in each column in the image.
- Taking the number of dots in the resulted array with number before and after it is less than or equal it is the highest one and indicate the true number of dots in the given column and then store the position of column and number of dots in it.
- Point is registered at the corresponding position in the array as 3x2 point.

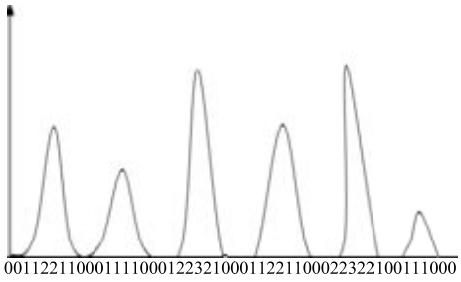

Figure 8: Increasing decreasing curve

- After this step the resulted image is containing 2*3 points which represent the recto dots in the correct position then we need to recognize the cells to identify the characters.

Figure 9: Whole dot detection output

2.4 Braille cells recognition

This step does not depend on fixed lengths for cells heights and distances between them.
The main idea in this step is to determine the number of rows and columns [1]

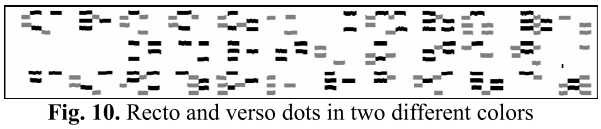

Fig. 10. Recto and verso dots in two different colors

2.5 Conversion to Binary

Each Braille character is coded into a six bit binary number based on the presence or absence of dots. The presence of a dot is indicated by a 1 and its absence by a 0.

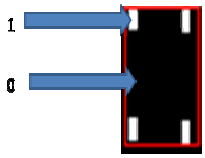

Fig. 11. Conversion to Binary

2.6 Conversion to text

Simply this step is a simple one to one mapping from a database containing the binary code and the letter corresponding to this representation
So, once the code is obtained corresponding to each Braille character, determined by the presence or absence of dots, it can be replaced by its corresponding character. Thus the Braille to text conversion takes place.

Fig. 12. Conversion to text

3. RESULTS & CONCLUSION

The system has been tested with a wide variety of scanned Braille documents, both single and double sided, produced with different mechanical methods and scanned with different scanners.

Due to the sheer variety of documents, it has not been possible to obtain textual ground truth. For our tests, ground truth was created manually by counting the number of Braille dots and characters in each document (both sides of a page if characters are present on the verso). As such, the general recognition results given here refer to the detection of dots and characters.
Finally, one of the factors which affect the performance of character recognition is the position of the dots of a character in the image. Currently, if a valid dot lies outside the expected bounds of the Braille character it will

not be assumed as part of that character. Hence, the character will not be correctly recognized.

Overall, the majority of the errors can be attributed to the quality of the image of the Braille document. Also, it should be pointed out that the quality of the Braille document itself is important. Very old documents with some of the protrusions flattened due to heavy use will give rise to more incorrectly recognized characters.

Work is currently being carried out to improve the performance of the system; experiments include scanning at different resolutions and greater grey level range Also considered is scanning of both sides of an inter-point Braille document and correlation of results. The implication of documents with different colors in the analysis is also under study. It should be noted, however, that this paper concentrates on the application of the method to inter-point Braille documents, which pose.

Far more challenges than the single sided ones that previous approaches have dealt with. As a whole, the approach described here shows the feasibility of a cost-effective, fast and easy to use Braille reading system. It does not require expensive custom made and complicated hardware [4]. It uses a flat-bed scanner which can be shared with other applications, inter-point Braille documents are handled in addition to single sided ones.

4. FUTURE WORK

In the future, the system aims to implement more features that couldn't be implemented at the present time.
The system will be able to interpret more than one language. Also, the system will be able to translate any Braille document regardless the grade in which the document was written with, like grade 2.The resolution will represent no problem in the scanning process as the system will be able to work properly at any gives resolution.

5. REFERENCES

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

2. AbdulMalik Al-Salman, Y. A. (2007). Arabic Optical Braille Recognition System. ICTA’07, April 12-14, Hammamet, Tunisia .

3. Bridson, A. A. (2004). A Robust BrailleRecognition System. Document Analysis Systems VI, A.

4. R.T. Ritchings, A. A. (1995). ANALYSIS OF SCANNED BRAILLE DOCUMENTS. In Document Analysis Systems, (pp. pp. 413-421). Manchester: World Scientific Publishing Co,.

5. 2001-CE-355, K. S. (2001). BRAILLE SPEAKER. Sir Syed University of Engineering and Technology , ALL.

6. André, P. (2005). Intelligent Flood Fill or: The Use of Edge Detection. A project report submitted for the award of MEng Computer Science , ALL.

7. Bridson, A. A. (2004). A Robust Braille Recognition System. Document Analysis Systems VI, A.