Designing Braille Copier Based on Image Processing Techniques

Abdul Malik S. Al-Salman, Ali El-Zaart, Yousef Al-Suhaibani, Khaled Al-Hokail, Abdu Gumaei

Abstract*— *Braille is a very important communication code for low vision and blind people. Recently, there has been an increasing trend to use computers for entering, editing and printing Braille documents using special purpose software and printers. Also, there is a large number of old Braille documents that have started to wear out and they need to be reproduced so that they can be preserved and copies made available to many people. Hence, the motivation of this research is the need to duplicate many of Braille documents automatically in very easy manner (like traditional photocopying machine) to be preserved and copies made available to many people. Implications of this research include building a Braille Copier machine that produces copies of Braille documents in exact format regardless of the language used. In addition, this machine is able to work as two-in-one (Coping and Printing). The method used requires optical recognition and image processing techniques so that Braille papers can be copied in similar way to copying ordinary printed text. The results obtained were excellent as we were able to copy Braille documents successfully for both single and double sided papers.

Index Terms*— *Braille Image Segmentation, Braille Cells, Verso dots, Recto dots, Grid formation and Cell Detection

I. INTRODUCTION

Braille is a reading and writing system which enables blind and partially sighted persons to read and write through touch. Braille system was invented by Louis Braille in 1824. It generally consists of cells of six raised dots which are arranged in three rows and two columns. These six positions of dots are arranged to give just 64 different Braille codes “characters”. Braille documents are essential communication means allowing blind and partially sighted persons to read and write through touch. There are established standards for the production of Braille that determine the height and diameter of a dot, the spacing between dots and between characters. Since the past two decades, there has been an increasing trend to use computers for entering, editing and printing Braille documents using special purpose software and printers. Computerized systems can now produce Braille documents from ASCII text. But these Braille documents are possibly weak or exposed to damage. Therefore, they must be reproduced so that they can be preserved and copies made available for many people.

Manuscript Received on November 2014.

Prof. Abdul Malik S. Al-Salman, Department of Computer Science, King Saud University, Riyadh, KSA, Saudi Arabia.

Dr. Ali El-Zaart, Department of Mathematics and Computer Science, Beirut Arab University, Beirut, Lebanon, Saudi Arabia.

Mr. Yousef Al-Suhaibani, Department of Computer Science, King Saud University, Riyadh, KSA, Saudi Arabia.

Mr. Khaled Al-Hokail, Department of Computer Science, King Saud University, Riyadh, KSA, Saudi Arabia.

Mr. Abdu Gumaei, Department of Computer Science, King Saud University, Riyadh, KSA, Saudi Arabia.

Manual transcription reproduction method is laborious, error-prone and costly; hence there is an important need for a system to automatically reproduce Braille documents. There are many existing methods for reading and translating Braille documents. One of such methods is the use of an *Optical Braille Recognition system* (OBR) to scan the Braille document and convert it into normal ASCII text, and then print it using a Braille embosser. In this regard, there have been many attempts to perform optical Braille page recognition using different methods. Dubus et al. [1] proposed an algorithm called a *Lectobraille* to translate relief Braille into Black-ink and convert it to printed text using an image processing technique. Mennens et al. [2] developed an optical recognition system based on a commercially available scanner to recognize Braille writing documents. However, this system cannot manipulate deformation in the dot grid arrangement. A flatbed scanner is used by Ritchings et al. [3] to scan both single and double-sided Braille documents at 100dpi and at 16 grey-levels with few image-based operations. These operations are used to skew Braille characters based on character-region search and to handle the variation in positions of characters based on a fixed grid. But the fixed grid caused some problems in Braille recognition. Blenkhorn [4] presented a method for converting Braille dots into print by using a finite state system to hold the current context with right context and then check what was achieved by using matching algorithms. This system was designed to be configurable for a wide range of languages and character sets. Also, Hentzschel and Blenkhorn [5] introduced a system for optical Braille recognition based on twin shadows technique, which subtracts two images of the same Braille page, where each image was taken under different illumination conditions. This system can locate and extract the Braille dots in Braille images as pairs of white and black spots, where each pair of white and black spots represents a single Braille dot, and if some pairs are lost, false ones are formed. However, this system did not detect the distances between Braille dots automatically. Oyama et al. [6] proposed a dot detection module incorporated in the OBR system to detect both recto and verso Braille dots on both single and double sided pages. The problem in this work was due to the difference in light reflectance between recto and verso dots. Ng et al., [7] presented an automatic Braille recognition system to translate Braille documents into English or Chinese text using edge enhancement, noise filtering and boundary detection techniques. The recognition rates were good; but, there is no explanation about grid deformed input, nor its effectiveness. A portable device has been designed by Murray and Dais [8] for optically scanning embossed Braille and conversion of the scanned text to binary Braille representation. Since the user of

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

this device is in charge of managing the orientation of scanning, only a small part is scanned at a time, and grid deformation is not a main concern. An easier algorithm was used to give efficient and immediate translation of Braille codes. Wong et al. [9] proposed an OBR system that is capable of recognizing a single sided Braille page in addition to preserving the format of the original document in the produced text file. The algorithm processes the image one row at a time reducing the computation time significantly. This recognition module is designed to work with threshold images resulting from the half-character detection module. The classification process is carried out using a probabilistic neural network. Antonacopoulos and Bridson [10] proposed Braille recognition system to identify Braille dots on both single and double-sided documents of average quality where many improvements were added to increase the cost-effectiveness and usability of the system. This approach is similar to the approach proposed by [3] but solves the problem caused by the fixed grid by using a flexible grid. The results reported for this system were over 99%, while Braille characters were also correctly recognized in over 99% of documents of average quality (in both single and double-sided documents). Falcón et al. [11] presented the development of *Braille-Lector* system that translates Braille scanned images into normal text, as well as speaking the translated text. *Braille-Lector* is a robust application with innovative thersholding and Braille grid creation algorithms which detects and reads Braille characters. The results reported were 99.9% of correct symbols and an error variance below 0.012. The conversion time reported is only 26 seconds for double-sided documents by using MATLAB programming language. Namba and Zhang [12] proposed the Braille image recognition system by CNN (*Cellular Neural Network*) for associative memory. Their system consists of three stages: pre-processing, feature extraction, and recognition. Pre-processing stage is achieved by gray-scale inversion, binarization, noise removal, dilation, and normalization. Feature extraction stage is achieved by downsizing, and adjustment on the pre-processed image. In recognition stage, representative binary patterns (±1) are stored in CNN; and based on an input pattern obtained by image processing, CNN self-recalls a pattern, which is considered as the final recognition result. The authors have obtained a good recognition rate (87.9%). Al-Salman et al. [13] developed a system to recognize an image of embossed Arabic Braille, both single and double sided and then convert it to text. This work helped to build a fully functional Optical Arabic Braille Recognition system. It has two main tasks, first is to recognize raised Braille cells, and second is to convert them to regular text. The results obtained were 99% of the dots are correctly recognized on both single and double sided Braille documents. There is a practical study on the development of a wearable sensor system for reading Braille is introduced by Tanaka et al. [14]. This study is intended to develop a compact tactile sensor system which uses a *polyvinylidene fluoride* (PVDF) film for the sensory receptor. Many of novel approaches are proposed by AL-Saleh et al. [15] and Al-Salman et al. [16] to detect and recognize Braille characters embossed on Braille document based on between-class variance with gamma distribution and a mixture of beta distributions in an attempt to increase the accuracy of OBR. Recently, there are many attempts Al-Shamma and Fathi [17] and Padmavathi et al. [18] for Braille recognition and conversion into text and voice. In addition, Shreekanth and Udayashankara [19] introduced a review traces the earlier works carried out by the researchers on the development of OBR and highlighted the existing OBR solutions with special emphasis on dot recognition of the embossed Braille image characters. However, the results of all previous methods were obtained when there was no rotation in acquired Braille documents. They focused merely on converting a Braille document into a Braille image and translating it into a natural language. The reproduction (copying) of the Braille document is very important when these documents need to be preserved and made available for many people. None of these methods mentioned here were able to copy the exact Braille document. Tetsuya and Susumu [20] introduced one of the current methods in this direction is to use a Braille Thermoform Machine using a piece of thermal-sensitive material, and melt the material on the document to be reproduced. While this method produces an accurate replica of the original document, it is very primitive, generates a single copy at a time, produces a bad smell because of heating, cannot be used either for double-side Braille paper, for plastic Braille documents, or make a copy on normal Braille paper; and the quality and resolution of the Braille dots (of the original document) are degraded as a result of the process. Another method is to use an *Optical Braille Recognition* system (OBR) to scan the Braille document, convert it to normal ASCII text and then print it using a Braille embosser. The problems with this method are: first, the new Braille document loses the format of the original document. Second, the OBR has to recognize the original language in order to convert it to text, so if it works for specific language it may not work for other languages. Moreover, there is a time overhead for Braille-to-Text translation. The rest of this paper presents the achieved work of building the *Braille Copier* (BC) with some testing results.

II. PROPOSED BRAILLE COPIER SYSTEM

Braille Copier system is composed of a computer connected to a flatbed scanner, a touch screen and a Braille embosser. All of these components are assembled to configure our Braille Copier as an integrated machine. A yellow transparent plastic can be placed above the scanner glass plate. This is needed when the user tries to scan the Braille document which is embossed on the white paper; it will be difficult to differentiate between the light regions of the dot and the paper background. As a result, we cannot identify and detect Braille dots. Fig. 1 shows the Braille Copier system and the flow of the copying process. The process starts with an original Braille document which will be scanned by flatbed scanner to produce an image. The image is processed in the subsequent stage by using image processing techniques to detect Braille dots and cells. The ASCII codes that correspond to the detected cells will be sent to the Braille embosser which will reproduce one or more copies of the Braille document.

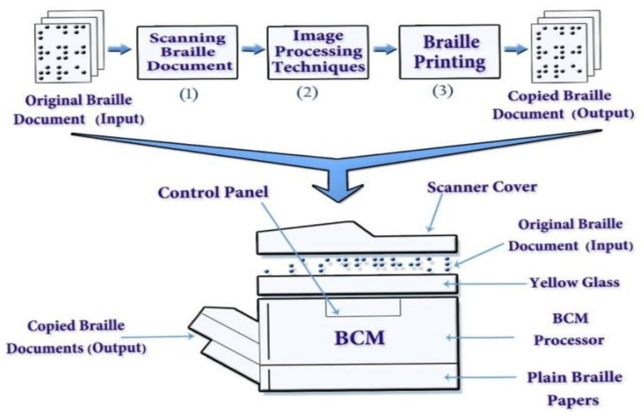

Scanning Braille Document
Image Processing Techniques
Braille Printing
Original Braille Document (Input)
(1)
(2)
(3)
Copied Braille Document (Output)
Control Panel
Scanner Cover
Original Braille Document (Input)
Yellow Glass
BCM
BCM Processor
Copied Braille Documents (Output)
Plain Braille Papers

Fig. 1. Braille Copier with its Copying Processes

Our application interface is very simple as shown in Fig. 2. This simplicity will help the blinds to easily use the BC. In the next subsections, we will describe each stage in the copying process.

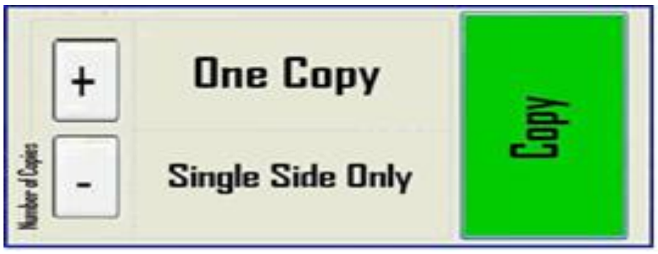


Fig. 2. System Interface of Braille Copier

A. *Review Stage Scanning Braille document*

Scanning Braille document is the first step to capture. the original Braille document that needs to be copied. Many devices, such as flatbed scanner and digital camera scan can be used to perform this step. The use of a commercially available flatbed scanner for image acquisition is a cost-effective solution. The original Braille page is scanned using flatbed scanner at 150 dpi resolution and a grey level Braille image is acquired. This resolution and Braille image format were selected experimentally. The only crucial observation is that the illumination of the Braille document is non-uniform, i.e. there is only one light source in the scanner and at a slight distance from the image sensor (CCD). Fig. 3 shows an image of a scanned single-sided Braille document.

Fig. 3.Scanned Single-Sided Braille Document

The scanned Braille image serves as an input to our dot and cell recognition software on the computer. The dot detection algorithm is based on the implemented algorithm. The recognition software will apply stability thresholding using Beta distribution in order to initiate the process of multi-mode estimator that calculates threshold values. Segmentation is then performed using those values. To ensure correct detection and extraction of dots composing of Braille characters, a grid is formed to contain the Braille dots. We identified a recto dot by a light region that exists above a dark region as shown in Fig. 4. In the same way, we identify a verso dot in double sided document by a light region that exists below a dark region as shown in Fig. 5.

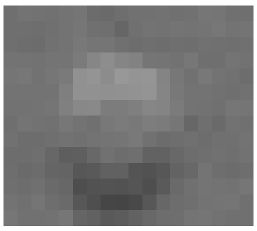


Fig. 4.Recto Dot

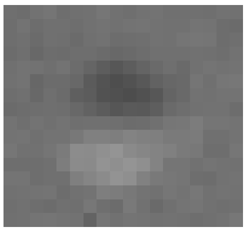


Fig. 5.Verso Dot

B. *Applying Image Processing Techniques*

A number of image processing techniques, in addition to scanning Braille document stage, can be applied to process Braille dots in Braille images. These techniques include image segmentation, image rotation, and grid formation for dot detection and cell detection. In this section, we will explain these techniques.

1) *Braille Image Segmentation*

The three modes of a histogram of a Braille image represent the following three classes of pixels: (i) Mode 1: represents the dark region of a recto and verso dot. (ii) Mode 2: represents the background. (iii) Mode 3: represents the light region of a recto and verso dot (See Fig. 4 and Fig. 5). The problem of segmentation is the estimation of thresholds T1 and T2 for separating the three classes. We assume that a histogram of a Braille image is a combination of three Beta distributions. The Beta distribution is a continuous probability distribution with the *probability density function* (pdf) defined on the interval [0, 1] [21]:

$$f(x,\alpha,\beta)=\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\ \Gamma(\beta)}\ x^{\alpha-1}\ (1-x)^{\beta-1};$$

Where α and β are the shape parameters of the distribution and must be greater than zero, x is a random variable, and it must be between 0 and 1. The Beta distribution can take different shapes depending on the values of its two parameters α and β. The histogram h(x) of a Braille image can be written as follows:

$$h(x)=p_1 f(x,\alpha_1,\beta_1)+p_2 f(x,\alpha_2,\beta_2)+p_3 f(x,\alpha_3,\beta_3)$$

The estimated threshold T_i^{new} of a Braille image can be calculated using the following formula [21]:

$$T_i^{new}=1-e^{\frac{-A-B\log(T_i^0)}{C}};$$

Where $A=\log((p_iK_i)/(p_{i+1}K_{i+1}))$; $B=\alpha_i-\alpha_{i+1}$; $C=\beta_i-\beta_{i+1}$ and

$K_r=\Gamma(\alpha_r-\beta_r)/\Gamma(\alpha_{r+1}-\beta_{r+1}), r=i,i+1$

The statistical parameters of the histogram (p_i,α_i,β_i); i=1, 2, 3 are estimated using the stability of thresholding algorithm [21].

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

2) *Braille Image Rotation*

The cells in Braille document are arranged in horizontal and vertical directions. Hence, the dots in Braille document are also arranged in horizontal and vertical directions. Due to some reasons of copying Braille document, there are many Braille documents where cells are not arranged in horizontal and vertical directions (See Fig. 6a). This has made the processing of dots and cells detection more difficult. To solve this problem, we rotated the Braille image by arranging the cells in right directions using the information in segmented image [5]. In Fig. 6b, the original Braille image is rotated where all dots are arranged in horizontal and vertical directions. Braille image rotation is achieved by the using binary search algorithm to arrange cells in horizontal and vertical directions. The maximum degree of recognizing a rotated image is 4 degrees from either the left or the right side. To calculate the rotating degree, we have used the following algorithm:

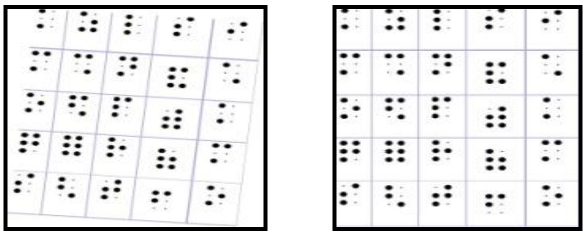


Fig. 6. (a) Original Braille Image Where Cells are Rotated. (b) Original Braille Image Where All Dots are Arranged in Horizontal and Vertical Redirections

Algorithm:

A. After Braille image segmentation, we select either the dark or the bright part that resulted from segmentation and delete the other part.

B. Horizontal projection is performed to count the number of pixels in each row for the selected part in the previous step. Then we calculate those rows having more than 10 points.

C. The image is rotated 4 degrees one time to the left and one time to the right; and in each time, step (2) is repeated.

D. Calculate the number of rows in the image in its three cases (right/ left/ middle) we have either of the two possibilities:
   - If the number of rows for the right side and the left side are the same, then the image is not rotated and we stop here.
   - If the number of the rows for the right side and the left side are different, then the image is rotated and we proceed to step E.

E. At this step we have two possibilities as well:
   - If the number of rows in the right side is less than that for the left side then we set the middle point calculated in the fourth step to be the new starting point for the left side. We then calculate the number of rows for this new middle point for both, the right and left sides.
   - If the number of rows in the left side is less than that for the right side then we set the middle point calculated in the fourth step to be the new starting point for the right side. We then calculate the number of rows for this new middle point for both the right and left sides.

F. Repeat steps D and E as long as half the difference between the right and left deviation is more than 1/16. After determining the deviation degree, we rotate the original gray image and then repeat steps A and B for the rotated image.

3) *Grid Formation for Cell Detection*

The procedure presented in this section is used for detecting recto dots from double-sided Braille documents. In order to accomplish this task, a grid is first formed using the segmented image and then the detection of dots follows. The grid is formed by, first selecting a starting dot and then generating horizontal lines with regular intervals and vertical lines. Horizontal lines generation is straightforward being compared with vertical lines generation. This is due to the regular horizontal arrangement of Braille cells opposed to the irregular vertical arrangements of cells. Therefore, we cannot predict the spaces between cells in the same line and draw vertical lines accordingly. After a gird with each of the recto dots contained within a box has been formed, recto dots detection should then take place. Starting from the first horizontal block in the resulting grid; only boxes are checked with certain size range. For each box in the grid, a test is carried out to decide whether it holds a recto dot or not. If a recto dot is found then it will be drawn on the output image in the same location. The output of this step is an object file that contains the coordinates of the top-left corner of each found recto dot. Verso dots are detected in a similar way. Having identified all possible valid dots, the system defines the region containing all the dots such that no dots exist outside this region. We know that there are standard distances between dots inside a cell (See Fig. 3). Also there are standard distances between cells in Braille image. Based on these distances, the Braille cells can easily be recognized.

4) *Cell Detection*

The purpose of this process is to detect the whole Braille cell and store it along with its location. This information is going to be used later at the printing stage. We have used arrays for storing Braille cells. For each side of Braille document, we used one dimension array to store all Braille cells in that side. Each element of the array points to another array that has six elements representing the six dots in a cell. The authors of the accepted manuscripts will be given a copyright form and the form should accompany your final submission.

C. *Braille Printing*

The detected cells from the previous step will be an input to the printing stage. The Braille embosser is controlled through programming interface and using standard Windows APIs. The printing command expects the ASCII code for each Braille cell in the document along with its coordinates and Braille font. In case of double sided documents, we send two pages to the printer and it will emboss them in one double sided paper. The cells are composed of zero or more dots; those dots will be used to identify the ASCII code of the whole cell. In order to make the Braille Copier works for any Braille document regardless of the original language, we will not convert the detected cells to its corresponding letter in any language. Instead, we created a lookup table (Table 1) of all the possible combinations of Braille dots in each cell starting from zero dots (no raised dots) to six raised dots. An alternative approach is to send each single dot with its coordinates to the embosser, yet this approach made the copying process very slow. The Braille embosser is controlled through our *Braille Copier's* interface using these two steps:

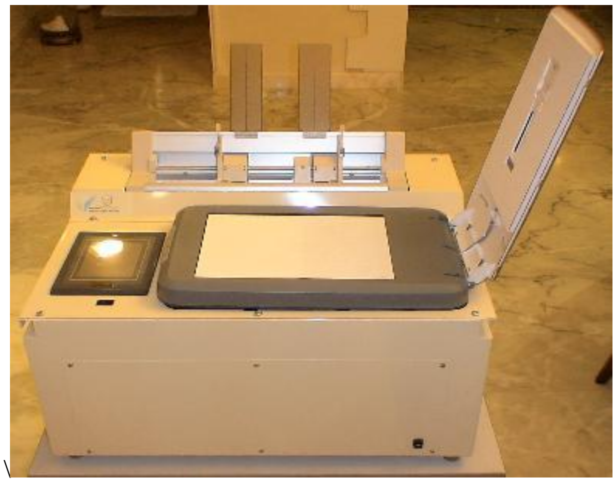
1. It allocates the ASCII code for each cell using the lookup table. We do this for all the cells in each line of the Braille document.
2. Then, it draws the processed cells in line with their coordinates (x, y).

It repeats the two steps for each line of the Braille document line by line and sends the Braille page to the printer (embosser).

Table 1.ACSII Codes Mapping

Combination	Table Index	ASCII Code	Cell	Combination	Table Index	ASCII Code	Cell
0	0	64	⠀	1	32	65	⠁
6	1	96	⠠	16	33	97	⠡
5	2	68	⠐	15	34	69	⠑
56	3	100	⠰	156	35	101	⠱
4	4	80	⠈	14	36	81	⠉
46	5	112	⠨	146	37	113	⠩
45	6	84	⠘	145	38	85	⠙
456	7	116	⠸	1456	39	117	⠹
3	8	66	⠄	13	40	67	⠅
36	9	98	⠤	136	41	99	⠥
35	10	70	⠔	135	42	71	⠕
356	11	102	⠴	1356	43	103	⠵
34	12	82	⠌	134	44	83	⠍
346	13	114	⠬	1346	45	115	⠭
345	14	86	⠜	1345	46	87	⠝
3456	15	118	⠼	13456	47	119	⠽
2	16	72	⠂	12	48	73	⠃
26	17	104	⠢	126	49	105	⠣
25	18	76	⠒	125	50	77	⠓
256	19	108	⠲	1256	51	109	⠳
24	20	88	⠊	124	52	89	⠋
246	21	120	⠪	1246	53	121	⠫
245	22	92	⠚	1245	54	93	⠛
2456	23	124	⠺	12456	55	125	⠻
23	24	74	⠆	123	56	75	⠇
236	25	106	⠦	1236	57	107	⠧
235	26	78	⠖	1235	58	79	⠗
2356	27	110	⠶	12356	59	111	⠷
234	28	90	⠎	1234	60	91	⠏
2346	29	122	⠮	12346	61	123	⠯
2345	30	94	⠞	12345	62	95	⠟
23456	31	126	⠾	123456	63	127	⠿

Fig. 7 shows our Braille Copier after assembling all the components


\
Fig. 7. Developed Braille Copier Device

We tested this machine on several Braille images (more than 200 images), the results were excellent. In the following, we will show the result of applying our technique to Braille documents. Fig. 8 shows an original image of double-sided Braille document.

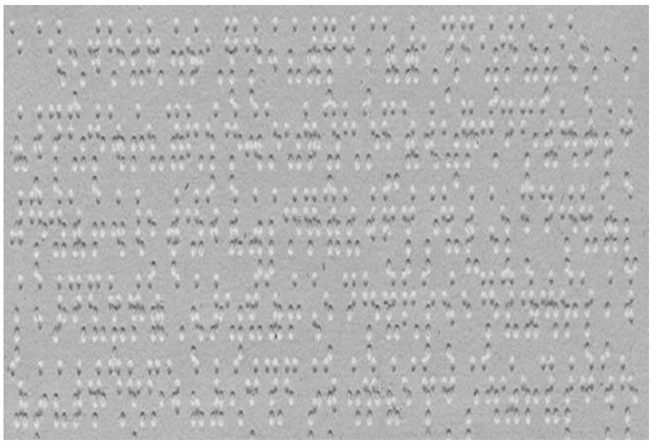


Fig. 8. An Original Double-Sided Braille Image

Fig. 9 shows the segmented image where there are three classes of pixels. Black pixels represents the background of Braille image, white pixels represents the light regions of recto and verso dots, and gray pixels represents the dark regions of recto and verso dots. Fig. 10 shows the detected recto dots from the double-sided Braille image. Fig. 11 shows the detected recto cells from the double-sided Braille image. We remark that the percentage of recognized cells is almost 100% (more testing is needed, especially with old Braille documents). The same results for verso dots and Braille cells recognition are shown in Fig. 12 and Fig. 13. Finally, Fig. 14 shows an image of the copied double-sided Braille document.

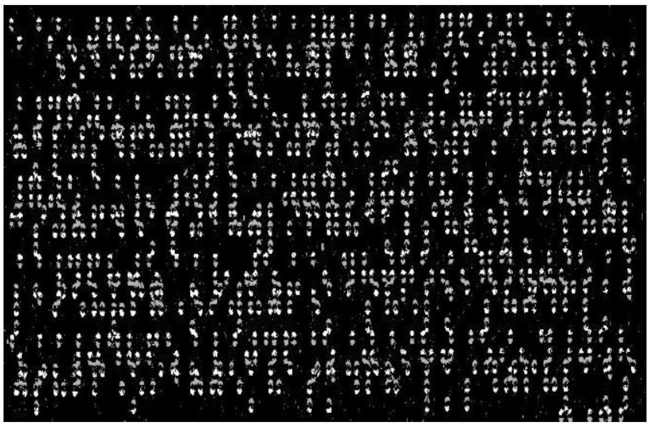


Fig. 9.Double-sided segmented image

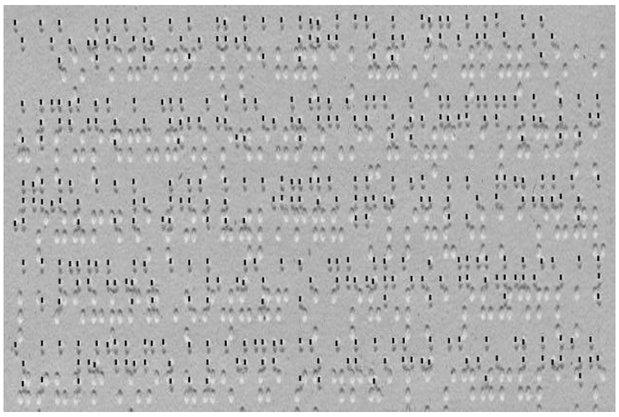


Fig. 10.Double–sided recto dots detection

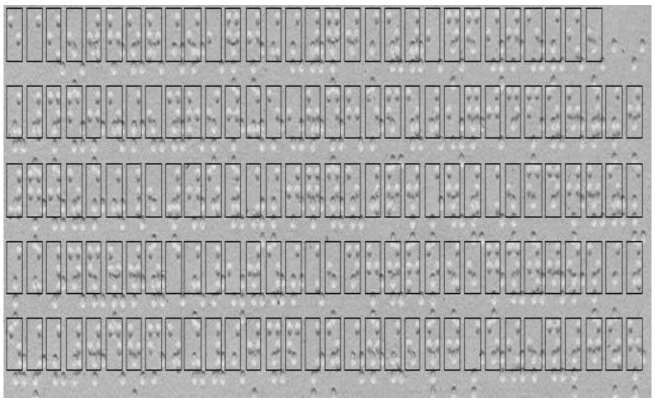


Fig. 11. Double-Sided Recto Cells Detection

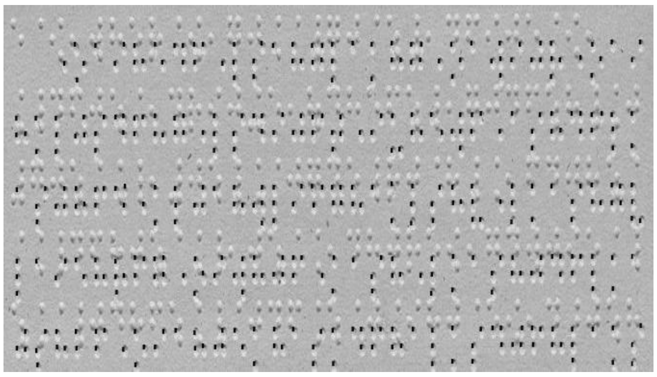


Fig. 12. Double-Sided Verso Dots Detection

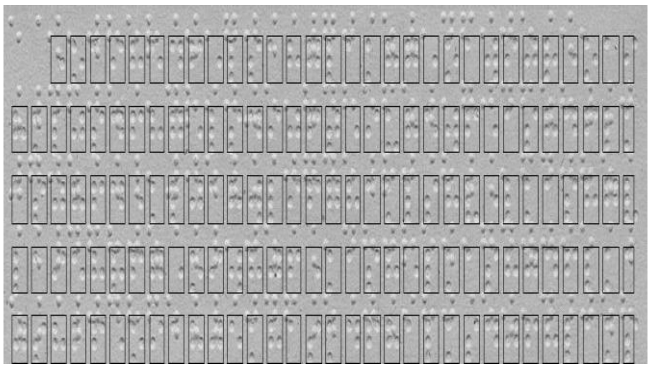


Fig. 13. Double-Sided Verso Cells Detection

Fig. 14. Copied Double-Sided Braille document

Also, we tested our machine on single sided Braille documents and the results are shown as follows; Fig. 15 shows an original image of single-sided Braille document. Fig. 16 shows a segmented image where there are three classes of pixels. Fig. 17 shows the detected dots from the image. Fig. 18 shows recognized cells for dots. We remark that the percentage of recognized cells is 100%. Finally, Fig. 19 shows a Braille image of the copied single-sided Braille document.

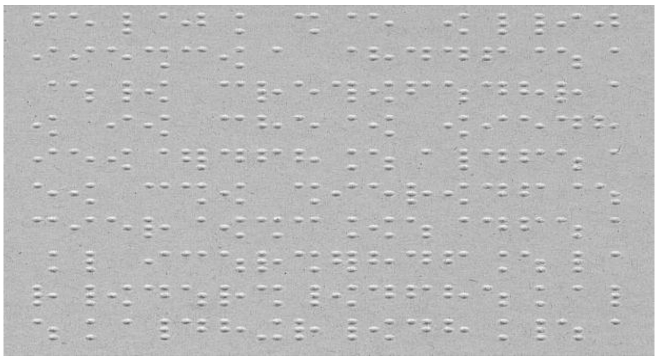


Fig. 15. An Original Single-Sided Braille Image

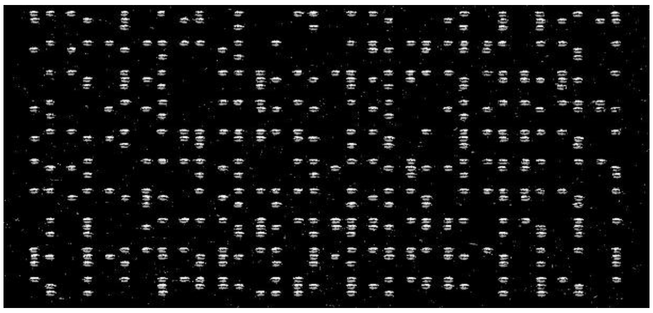


Fig. 16. Single-Sided Segmented Image

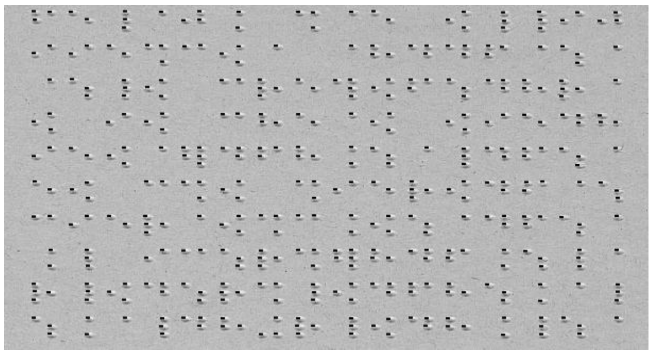


Fig. 17. Single-Sided Dots Detection

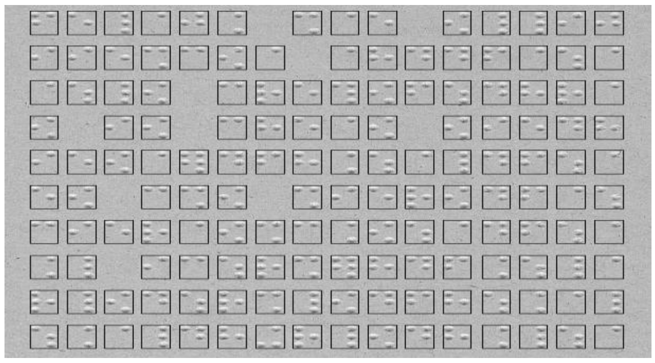


Fig. 18. Single-Sided Cells Detection

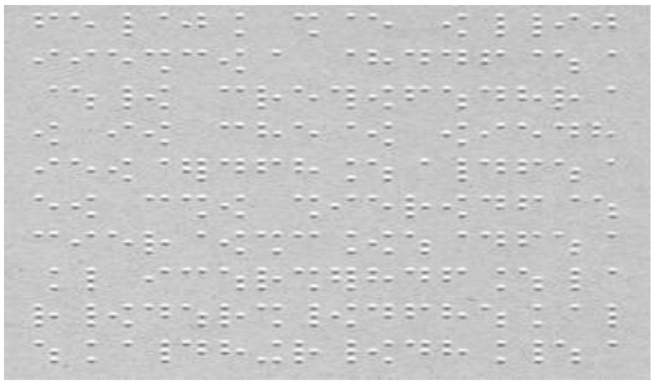


Fig. 19. Copied Single-Sided Braille document

IV. DISCUSSION

The need to reproduce a copy of Braille documents is significant and the current methods of reproducing Braille documents have many drawbacks. We have completed successfully a proof of the concept of building a *Braille Copier* (BC) that produces a copy of the original document on plain Braille paper with the exact format regardless of the language. Our method involves using optical recognition and image processing techniques. Braille papers are copied in similar way to copying ordinary printed text. The BC is composed of a computer (a Programmed Chip can be used instead) connected to a scanner, a touch screen and a Braille embosser. The machine task can be summarized into three steps as the following:

- Utilize an ordinary (flatbed) scanner to scan the original Braille document producing an image.
- Process the image: After fixing the skew, detect Braille dots & cells and maintain the parameters of the original document such as paper margins and format.
- Send the ASCII code corresponding to the detected Braille cells to a Braille embosser to produce one or more copies similar to the original Braille document.

The results were very promising. We were able to copy single and double sided Braille documents successfully with the exact format and margins. The BC will detect automatically whether the original Braille document is single or double sided and embosses the same. However, the user can copy just one single side from a double sided Braille document by choosing this option.

V. CONCLUSION AND FUTURE WORK

In this paper, an invented *Braille Copier* based on image processing techniques is presented. *Braille Copier* system consists of two main parts, are the system hardware (a computer connected to a scanner for scanning Braille document, a touch screen for dealing with the system interface and controlling the device and a Braille embosser for printing) and the system software (system interface and system algorithms). The system software is very simple and coded in C#. There are three main steps to do the copying process. First step is scanning Braille document. Second step is applying image processing on the scanned Braille image. Finally, third step is printing the ASCII code corresponding to the detected Braille cells to a Braille embosser to produce one or more copies similar to the original Braille document. Experimental results of *Braille Copier* on several Braille documents indicate that we are able to copy single and double sided Braille documents successfully with the exact format and margins. Future work will be focused on four points. First point is enhancing the recognition component in order to increase the detection ratio even if there is a high degree of skew. Second point is expanding the application to make it copy any raised dots on the paper so that users can copy Braille pages that have simple graphics such as circles and squares. Third point is adding automatic sheet feeder to the machine, which at present is difficult because raised dots make Braille documents stick together thus making the scanner/copier to pull more than one document at a time. Finally, fourth point is checking document size compatibility between the original Braille document and the machine plain paper. Another improvement is to produce another version of the machine with a bigger scanner (A3 size) with lighter components and will be capable of accepting all sizes of Braille papers. Although this research led to build a new machine to duplicate Braille documents successfully and the results obtained were excellent but there are some limitations. The first limitation is that the maximum degree of recognizing a rotated Braille image is four degrees from either the left or the right side. The second limitation is that when a Braille image is upside down. In this case, we need to identify if the document is upside down. For future work, we plan to extend our research to use Hough Transform method to detect any rotation angle of Braille document and context analysis to recognize if Braille document was upside down.

VI. ACKNOWLEDGEMENTS

The authors would like to thank Dr. Jacob Adeniyi, and Mr. Abdulaziz Al-Qabbany for their assistance, valuable comments and contributions. Also, the authors would like to thank the National Plan for

Science and Technology for its funding and support.

REFERENCES

[1] Dubus, J.P., 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", Proceedings of the Annual International Conference of the IEEE (Engineering in Medicine and Biology Society), Nov. 4-7, New Orleans, LA, USA, pp.1584–1585. DOI:10.1109/IEMBS.1988.94726, 1988.

[2] Mennens, J., Tichelen L. V., Francois G., and Engelen J., "Optical Recognition of Braille Writing Using Standard Equipment", IEEE Transactions on Rehabilitation Engineering, 2(4):207–212, DOI:10.1109/ 86.340878, 1994.

[3] Ritchings, R. T., Antonacopoulos, A., Drakopoulos, D., "Analysis of Scanned Braille Documents", In: Dengel, A., Spitz, A.L. (eds.), Document Analysis Systems, World Scientific Publishing Company, pp:413–421, 1995.

[4] Blenkhorn, P., "A System for Converting Braille into Print. IEEE Transactions on Rehabilitation Engineering", 3(2): 215–221, 1995.

[5] Hentzschel, T. W., and Blenkhorn, P., "An Optical Reading Systems for Embossed Braille Characters using a Twin Shadows Approach", Journal of MicroComputer Applications, 18(4):341-354. DOI: 10.1016/S0745-7138(05)80034-X, 1995.

[6] Oyama, Y., Tajima, T., and Koga, H., "Character Recognition of Mixed Convex-Concave Braille Points and Legibility of Deteriorated Braille Points", System and Computer in Japan, 28(2): 44–53. DOI: 10.1002/(SICI)1520-684X(199702)28:2<44::AID-SCJ5>3.0.CO;2-R, 1997.

[7] Ng, C., Ng, V., and Lau, Y., "Regular Feature Extraction for Recognition of Braille", Proceedings in Third International Conference on Computational Intelligence and Multimedia Applications, ICCIMA 99, pp.302–306. DOI:10.1109/ICCIMA.1999.798547, 1999.

[8] Murray, I. and Dias, T., "A portable device for optically recognizing Braille", PART I: Hardware Development, Proceedings in the Seventh Australian and New Zealand Intelligent Information Systems Conference, pp.129–134. DOI: 10.1109/ ANZIIS.2001.974063, 2001.

[9] Wong, L., Abdulla, W., and Hussmann, S., "A Software Algorithm Prototype for Optical Recognition of Embossed Braille", Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004, August 23-26, IEEE Xplore Press, pp:586-589.DOI:10.1109/ICPR.2004.1334316, 2004.

[10] Antonacopoulos, A. and Bridson D., 2004. A robust Braille recognition system, In: Document Analysis Systems VI, Lecture Notes in Computer Science (3163), Springer Berlin/Heidelberg, pp.533-545. DOI: 10.1007/978-3-540-28640-0_50. ISSN 0302-9743.

[11] Falcón, N., Travieso, C. M., Alonso, J. B., and Ferrer M. A., "Image Processing Techniques for Braille Writing Recognition", Lectures Notes in Computer Science, Ed. Springer, pp.379-385. ISSN: 0302-9743, 2005.

[12] Namba, M., and Zhang, Z., "Cellular Neural Network for Associative Memory and Its Application to Braille Image Recognition", Proc. of IJCNN’06, pp. 4716-4721. DOI:10.1109/IJCNN.2006.247066, 2006.

[13] Al-Salman, A., Al-Ohali, Y., Al-Kanhal, M., and Al-Rajih, A., "An Arabic Optical Braille Recognition System", ICTA'07, Hammamet, Tunisia, pp.81-87, 2007.

[14] Tanaka, M., Miyata, K., and Chonan, S., "A Wearable Braille Sensor System With a Post Processing", IEEE/ASME Transactions on Mechatronics, 12(4):430 438. DOI:10.1109/TMECH.2007.901923, 2007.

[15] AL-Saleh, A., El-Zaart, A., and Al-Salman, A., "Dot Detection of Braille Images Using A Mixture of Beta Distributions", Journal of Computer Science (Science Publications) 7 (11): 1749–1759, ISSN 1549-3636, 2011.

[16] Al-Salman, A., El-Zaart, A., Al-Salman, S., Gumaei, A., "A Novel Approach for Braille Images Segmentation, International Conference on Multimedia Computing and Systems (ICMCS 2012), Tangier, May 10-12, IEEE Xplore Press, pp.190-195. DOI:. DOI:10.1109/ICMCS.2012.6320146, 2012.

[17] Al-Shamma, S. D. and Fathi, S., "Arabic Braille Recognition and Transcription into Text and Voice", 5th Cairo International Biomedical Engineering Conference 2010, Cairo, December 16-18, pp.227-231. DOI: 10.1109/CIBEC.2010.5716095, 2010.

[18] Padmavathi, S., Manojna, K. S. S., Reddy, S. S., Meenakshy, D., "Conversion of Braille to Text in English, Hindi and Tamil Languages", International Journal of Computer Science, Engineering and Applications, 3(3):19:32. DOI:10.5121/ijcsea.2013.3303, 2013.

[19] Shreekanth, T., and Udayashankara, V., "A Review on Software Algorithms for Optical Recognition of Embossed Braille Characters", International Journal of Computer Applications, Published by Foundation of Computer Science, New York, USA. 81(3):25-35. DOI: 10.5120/13993-2015, 2013.

[20] Tetsuya, W., and Susumu, O., "A study on legible braille patterns on capsule paper: Diameters of braille dots and their interspaces on the original ink-printed paper", The Bulletin of the National Institute of Special Education, 30:1-8, 2003.

[21] El-Zaart, A., and DjemelZiou, "Statistical Modeling of multimodal SAR Images", International Journal of Remote Sensing, 28(10):2277–2294. DOI: 10.1080/ 01431160600933997, 2007.