ISSN:0975-9646

Knowledge based Inter-point Braille document Analysis and Recognition

Srinath.S[1] , C.N. Ravi kumar[2]

[1,2]*Department of Computer Science and Engineering*
JSS Research Foundation
JSS TI's Campus, Manasagangothri, Mysore-570006, Karnataka, India

***Abstract*—** Braille is a writing system used by visually impaired community. The Braille writing is embossing dots arranged in different combinations of three rows and two columns. Braille can be embossed only one side using Braille slate. Braille printer or Braille embosser can print simultaneously on both sides of a paper. Inter-point Braille document looks like having little mountains (front side embossing-recto dots) and valleys (back side embossing-verso dots). The reading finger of visually impaired will read only the bulged mountains of the Braille dots and hence only one side of the document can be read by human reading. The valleys are to be read from other side of the document. When you turn the page to read these valleys they look like mountains on that side. For human reading by visually impaired they are read separately. But when the inter-point Braille is digitized, we can recognize both sides in one scan using image processing technique. This paper presents an algorithm which analysis and recognizes both front and back embossed dots of single scanned image of an Inter-point Braille document. Algorithm uses the knowledge of Braille dot size, and dimension of the Braille cell**.**

***Keywords*—**Braille document Image processing, Inter-point Braille document, dimension of the Braille cell

I. INTRODUCTION

A. Braille Document:

Reading and writing by visually impaired persons is through an embossed language called Braille language. It is uses 6 dots notation for every character of the language. It is not the language by itself. Instead the normal language characters are mapped on to Braille language. Six dots used in Braille language are arranged in 64 possible combinations are mapped to different characters of normal language. Every language has its own Braille representation. Out of six dots used for a character, we can have 64 different combinations and a language which has less than 64 characters can be easily mapped on to Braille. For other languages more than one Braille box is used to represent a character [1-5]. The six dots of a character together called Braille cell. Braille slate and stylus are popularly used for embossing the characters on a sheet. It is manual embossing. Embossing can also be done in larger scale using Braille embosser. It is called machine embossing.

B. Inter-point Braille Document:

Embossing Braille using machine allows using both sides of a sheet. It embosses par allay both from front and back side. It reduces the cost of paper by 50%. Double sided printout used inter-point system of embossing. We have only 22 lines per sheet in normal embossing. In inter-point embossing with both side printing 44 lines can be embossed per sheet in sheet of size 8x10 inch. To emboss on the back side Braille, the space between the dots of the front side is used. Figure-1 shows the template of inter-point Braille

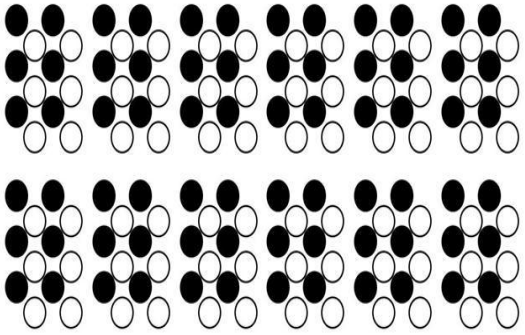


Figure-1
Inter-point Braille Template

The Braille document which is punched from back side of the sheet will have embossed dots created on the front side of a sheet. In our discussion this is called as front embossed dots or little mountain. Space between the front embossed dots are effectively used and punched from the front side and it creates embossing on the back side of the sheet, and it is called back embossed dots or little valley. The reading finger of visually impaired will read only the bulged mountains of the Braille dots and hence only one side of the document can be read by human reading. The back embossed dots are to be read from other side of the sheet. Figure-2 shows inter-point Braille document which is digitized using a scanner. When the inter-point Braille is digitized, we can recognize both sides in one scan using image processing techniques.

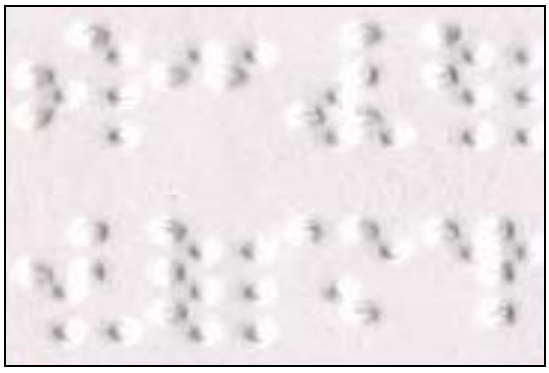
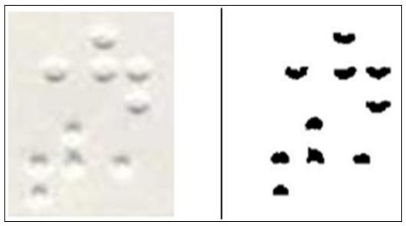


Figure-2
Digitized Inter-point Braille document

C. Analysing inter-point Braille document:

Digitized Braille document is binarized and pre-processed to eliminate all the noise to retain only the true Braille dots. After this the true Braille dots creates dark regions in different shapes depending on the light source and also for mountains and valleys.

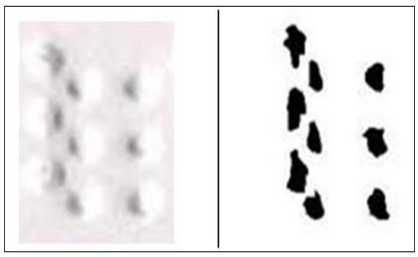
If the scanner lights fall from the top edge of the scanner, then dark region will be created as: Semi-circular fashion for the mountains at the bottom half of the Braille dot and Semi-circular fashion for the valleys at the top half of the Braille dot. It is as shown in Figure-3 (a) and its pre-processed version is shown in Figure-3(b).



(a) (b)

Figure-3
Scanner light falling from top edge

If the scanner light is focused from the left side of the Braille document, the dark regions will be created as: Semi-circular fashion for the mountains at the right half of the Braille dot and Semi-circular fashion for the valleys for the left half of the Braille dot. It is as shown in Figure-4(a) and its pre-processed version is shown in Figure-4(b).



(a) (b)

Figure-4
Scanner light falling from left edge

II. LITERATURE SURVEY

Before processing the Inter-point Braille document for character recognition, the front and back side of the embossing are to be separated. Very few researchers have worked on separating inter-point Braille document and they have worked based on the dark pattern created. Efficiency depends on the dark pattern created and it also requires the knowledge the position of the light source in the scanner while digitizing. The conventional methods which use the shape of the dots may fail because the shape may vary depending on the lighting condition. Smudging of verso and recto dots also leads to creation different shaped dots. In our work we are proposing a novel algorithm which overcomes the problems faced by the contemporary researchers.

III. PROPOSED METHOD

Given an inter-point Braille document is pre-processed to eliminate all noise and correct the skew [9] if any. Use this document for further processing. Inter-point Braille document shown in Figure-2 is pre-processed and it is as shown in Figure-5. The main aim to recognize the front and back embossed Braille characters.

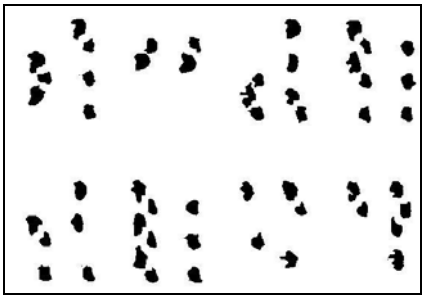


Figure-5
Pre-Processed Inter-point Braille document
(Scanner light focused from left edge)

Proposed method uses dot position in place of the dot shape. Average Braille dot size is computed using successive experiment. Dot distance within character, between the characters and between the lines are standard in nature and this knowledge base [10,14-15]is used in processing the inter-point Braille document.

Scan through the document pixel by pixel from top left edge row wise till we get upper edge of the first dot row. Similarly mark the left edge of the left Braille dot column. It is as shown in Figure -6

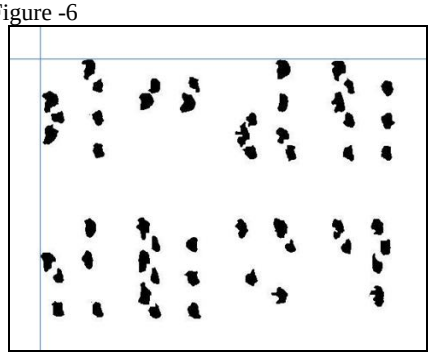


Figure -6
Marking upper and left edge of the Braille document

Now process the Braille document from bottom row to identify the bottom edge of the last Braille dot row. Similarly identify the right edge of the rightmost Braille dot column. Now the four edges of the Braille dot column are identified and it is as shown in Figure-7.

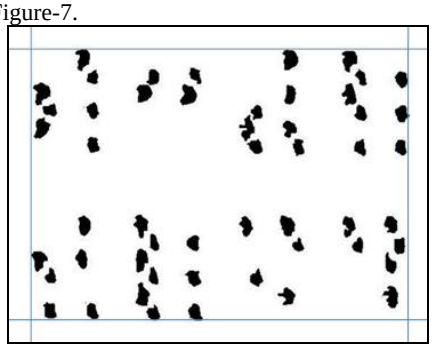


Figure-7
Marking upper, bottom left and right edge of the Braille document

Remove all the edges of the Braille document which are of no significance in processing. Figure-8 shows the pre-processed Braille document after removing the edges of the Braille document. This process is done to remove part of the document which is of no use in computation.

As stated earlier the average size of the Braille dot is computed. As the space vertical space between the dots within the character and space between the lines are known, moving from the top edge keep drawing the horizontal lines are marked and the till the end of the document. Note the co-ordinates. Process is repeated vertically from left edge moving towards right edge. It is as shown in Figure -9

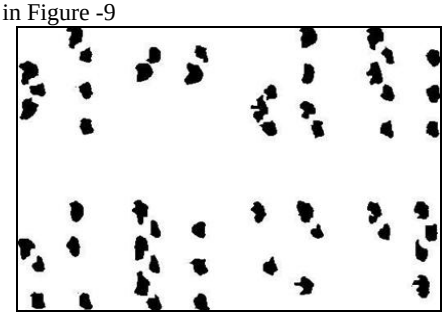


Figure-8
Pre-processed Braille after removing the edges

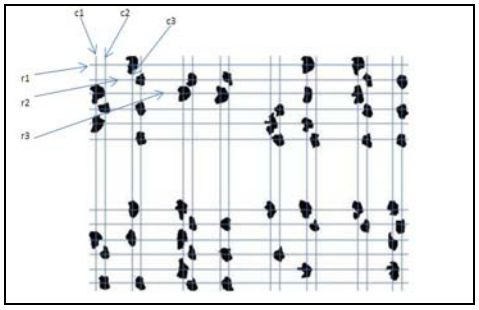


Figure-9
Braille document with Horizontal and vertical dot positions marked

After marking the lines, the vertical odd lines (say c1, c3, c5, c7 etc.) represents the lines having front faced Braille dots. Similarly the even vertical lines (say c2, c4, c6, c8 etc.) represents the lines of the back embossed Braille dots.

The first character in the first line can have dots in positions where c1,c3 meets with rows r1,r3,r5. To find the presence of dots in the six possible position of the first front embossed character, a mask of size 15X15 ones (White) is multiplied element by element (dot multiplication) with the possible dot having center at the row and column meeting position. The resultant matrix is all 1 if there is no dot present else indicates the presence of dot. This is done with histogram computation. Repeat the process for all the six dot positions and get the data about presence or absene of dots. The mask used is as shown in Figure-10.

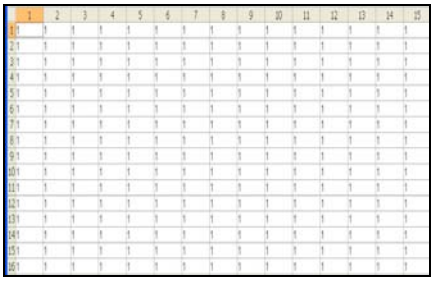


Figure-10
Mask used for detection of dots

The process is repeated till the end of the line to get the information about the front faced Braille characters. Continue the process till the end of the document.

In the next iteration the entire process is repeated to find the back embossed character dots. Column numbers cn,c(n-2) (alternate columns) and row number r2,r4,r6 are associated with back embossed first line first character dots. The dot positions are also exchanged in columns. This is because the last character of back embossed is the first character when it is turned back.

D. Algorithm:

1. Start
2. Read the two Braille document image
3. Pre-process the document to convert the document into binary image and to eliminate the noise and extra dot which are not the part of Braille character.
4. Eliminate the extra edge parts of the document which does not carry any data.
5. Use the knowledge base about the size of the Braille dot, distance between the dots within the character and between the characters, between the Braille character lines and also between the inter-point Braille dots.
6. Top edge of the document represents the upper edge of the first Braille dot row. Based on the Knowledge base of step 5, draw the marking line horizontally in the middle of the dot. Repeat the process from top to bottom of the document. Mark them as c1, c2, c3, …cn
7. Left edge of the document will contain the left most Braille dot column.
8. Step 6 is repeated vertically from left edge to right edge, marking vertical lines in the center of the Braille dots based on the knowledge base.
9. All the odd number columns (c1, c3, c5, c7…) and odd number rows (r1, r3, r5, r7, r9 …) are associated with the front faced character. The Even number columns (c2, c4, c6, c8…) and even number rows (r2, r4, r6, r8 …) are associated with the back faced characters.
10. First two odd number columns and three odd number rows intersect points can have Braille dots corresponds to first line front faced Braille character.
11. A mask of size 15x15 ones is used to detect the presence of dot in the intersect position.
12. Keeping the intersect point as centre, 15x15 pixels are extracted and it is multiplied with the mask element by element.
13. If the resultant matrix contains all ones it indicates the absence of dot, else the intersect point contains the Braille dot.
14. Step 12 and 13 are repeated for all the six dot positions and the dot presence is computed.
15. Based on the dot pattern the normal language character is represented.
16. All the characters in the first line is analysed and recognized by repeating the Steps 10 to 15 with subsequent columns keeping the same rows.
17. The next three odd rows represent the next Braille line. Steps 10 to 16 are repeated in this line.
18. Steps 17 are repeated the entire front faced Braille character lines.
19. First three even number rows (r2,r4,r6) and last two even number columns (cn, c(n-2)) are associated with first character of the back embossed character.
20. Braille dots in cnth column contains the Braille dots corresponding to first column of the first Braille character for the back embossed character, and the last but second column i.e c(n-2)th Represents the second column of the first Braille character.
21. Step 11 to 18 is followed in similar way for the back embossed dots.
22. End.

IV. CONCLUSION

The proposed algorithm presented analyses and recognizes the inter-point Braille successfully. Irrespective of the shape create by the front and back embossed Braille dots, character recognition is done with the knowledge base about the dimension of the Braille cell is used effectively in the algorithm

ACKNOWLEDGMENTS:

Our sincere thanks to Visvesvaraya Technological University for funding the Equipments required for this work under VTU Research grant Scheme.

REFERENCES

[1] Paul Blenkhorn, "A System for Converting Braille into Print", IEEE transactions on rehabilitation engineering, Vol. 3, No. 2, June 1995. Pp 215-221, 1995

[2] R.T. Ritchings, A.Antonacopoulos and D.Drakopoulos, " Analysis of Scanned Braille Documents", Document Analysis Systems, World Scientific Publishing Company, pp413-421, 1995

[3] C. Ng, V.Ng and Y.Lau, "Regular feature extraction for recognition of Braille". Third International conference on computational Intelligence and Multimedia Applications, 1999. ICCIMA '99 Proceedings, pp 302—306, 1999.

[4] Minghu Jiang,Xiaoyan Zhu, Georges Gielen, Elliott Drabek, Ying Xia, Gang Tan, Ta Bao, " Braille to print translations for Chinese", Journal of Information and Software Technology, pp 91-100, 2002

[5] Lisa Wong, Waleed Abdulla and Stephan Hussmann. "A software Algorithm prototype for Optical Recognition of Embossed Braille", Proceeding of 17th International Conference on Pattern Recognition (ICPR 04), Cambridge, UK, Vol. 2, pp 586-589, 2004

[6] Vidyashankar, G.Hemantha Kumar, P.Shivakumara, " Rotation Invariant Histogram Features for Recognition of Braille Symbols", Proceeding of the International Conference on Cognitive Systems (ICCS'04), New Delhi, India, pp 32-41, 2004

[7] Abdul Malik S, Al-Salman, yosef Alohali and Laylo O.Al-Abdulkarim, "Trends and Technologies in Optical Braille Recognition", 3rd International Conference on Information Technology (ICIT07), Paper ID 337, 2007

[8] Babadi,M.Y, Nasihatkon,B., Azimifar, Z., Fieguth,P, "Probabilistic Estimation of Braille Document Parameters", IEEE International Conference on Image Processing, Cairo, pp 2001-2004, November 2009

[9] Abdul Malik S, Ali El-Zaart and Abdu Gomai, "A New Approach for Adjusting Braille Image Skewness in Optical Braille Recognition", ICSECS 2011, Part I, Springer-Verlag Berlin Heidelberg 2011, CCIS 179 pp 735-746, 2011

[10] Srinath.S and C.N. Ravikumar, "An Insight into Optical Braille Character Recognition since its conceptualization", International Journal of Computer Applications, Vol 33, No.6 November 2011, pp 1-4, 2011

[11] Bhattachary, J; Majumder,S., " Braille Character Recognition using Generalized Feature Vector Approach", International Conference on Information Processing (ICIP), Bangalore, India, Vol.157, pp 171-180, 2011

[12] Srinath.S and C.N. Ravikumar, "Recognition of Hand Punched Kannada Braille Characters using knowledge based Multi Decision Concept: Basic Symbols", International Conference on Advances in Computing ,MSRIT, Bangalore (Published in "Advances in Intelligent and Soft Computing" series, Springer), pp 847-857, 2012

[13] S.Padmavathi, Manojna K.SS, Sphoorthy Reddy.S and Meenakshy.D, "Conversion of Braille To Text in English, Hindi and Tamil Languages", International Journal of Computer Science Engineering and Applications (IJCSEA), Vol-3, No-3, pp 19-32, June 2013.

[14] Information about Braille Script: http://scriptsource.org/cms/scripts/page.php?item_id=script_detail&key=Brai

[15] National Association for Blind at: http://www.nabindia.org/