

# AN ADAPTIVE APPROACH FOR OPTICAL RECOGNITION OF INTER-POINT HINDI DEVANAGARI BRAILLE AND ITS TRANSCRIPTION TO TEXT

SHREEKANTH.T<sup>1</sup>, UDAYASHANKARA.V<sup>2</sup>

<sup>1</sup> Research Scholar JSSRF, <sup>2</sup> Department of Instrumentation Technology  
Sri Jayachamarajendra College of Engineering, Mysore-570006, India.

E-mail: [speak2shree@gmail.com](mailto:speak2shree@gmail.com), [v\\_udayashankara@yahoo.co.in](mailto:v_udayashankara@yahoo.co.in)

**Abstract**—Optical Braille Character Recognition (OBR) system is in great need in order to restore the old Braille documents to make them available for the large section of visually impaired people. The recognition of the double sided Braille dots is a challenging task due to the overlapping of the front side dots with back side dots in an inter-point Braille. The fact that the Braille documents can be acquired with varied resolution makes the usual method of template matching fail. To address this issue, this paper introduces a new system that is capable of differentiating the recto and verso dots from an Inter-point Braille document even with varied resolutions. The primary reason for selecting the Hindi Devanagari Braille is that, Hindi is the national language of India and OBR for Hindi Braille is not available. This effort of OBR development for Hindi Devanagari Braille has been done for the first time. The proposed system is more efficient than the other state of the art techniques in terms of computation time and accuracy.

**Keywords**—Bharati-Braille; Connected Component labelling; Projection profile; Recto; Verso.

## 1. INTRODUCTION

Braille system, the most extensively used method by the visually impaired people to read and write was devised by Louis Braille in 1824. Any Braille document comprises of a line of characters and each character is made up of raised dot patterns. Each character is a 3x2 matrix of points and sixty four combinations are conceivable using any of these 6 dots [1]. Based on the existence of protrusions and depressions the Braille document can be categorized as single sided and inter point Braille. If the document contains only the protrusions on single side, it is a single sided Braille document as shown in Fig.1 (a). If the document contains both protrusions and depressions on single side then it is a double sided Braille document and is shown in Fig.1 (b). The printed Braille documents are very bulky and to mitigate this problem, most Braille documents are printed in inter-point with the embossing done on both sides of every page with a slight diagonal offset to prevent the dots on the either sides from interfering with one another [2]. Since in inter-point Braille the dots on both sides of the page are

visible from one side, both sides of the sheet can be recognized in a single scan thus making the translation process simpler.

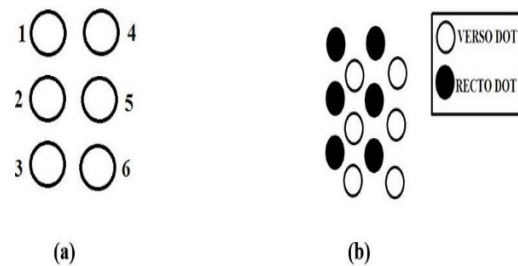


Fig 1: (a) The Braille Cell (b) Inter-Point Braille

It is essential to transliterate the Braille documents to the text document in the corresponding language in order to establish a bi-directional communication amid the visually impaired and the sighted community practicable. Also, persons who wish to work with blind people and do not know Braille can successfully communicate with the blind through the OBR.

Various authors have developed OBR systems for English and Arabic Inter-point Braille document [1]-[9]. All these approaches were based on the common principle of exploiting the differences in the gray levels in the image caused by the reflected light and the shadows created by the protrusions and the depressions on the document surface during the scanning process. Two standard templates were then constructed to represent the recto and verso dots. The templates were then applied to every position of the image and evaluated the correlation at each pixel position. Depending on the correlation values obtained, the recto and verso dots were extracted. These techniques differ only in the process of grid construction.

The significant limitations of the earlier works are:

- i) The augmented average processing time introduced by the template matching procedure for differentiating the recto dots from verso dots
- ii) The call for redesigning of templates depending on the height of the Braille character as the spatial resolution of the

Braille image show a discrepancy. Thus these techniques do not direct to an adaptive approach.

An extensive literature survey indicates that lesser number of works have been done on recognition of Inter-point Braille document as the recognition of single sided Braille documents is easier when paralleled to the double sided Braille document. Also the OBR systems have been effectively developed for recognition of printed braille characters of non-Indian languages. Most of the works have been done for English and Arabic Braille documents. More efforts have to be put in for the development of an efficient OBR for Indian languages, especially for Hindi, the national language.

Motivated by the above facts, an attempt has been made in the present work to develop an adaptive algorithm using the concept of connected component labelling based on eight connectivity and projection profile for Hindi Devanagari Braille. The experimental results show that the proposed method can provide improved performance when compared to the other techniques that are debated. This paper is divided into four sections: In section II an overview of Bharati Braille is presented. Section III discusses in detail the proposed work. Section IV presents the results and discussion. Section V, provides conclusions and directions for the future work.

## 2. AN OVERVIEW OF BHARATI BRAILLE

Bharati Braille is a largely unified Braille script for writing the languages of India which was developed and urbanized in April 1951. The Bharati Braille Code is the collective appellation given to all the dialects of India. The Bharati Braille uses phonetic equivalents from the English Braille to represent the Hindi texts and its Braille notations are shown in Fig.2 [11]. Bharati Braille confirms to the syllabic writing system for all the Indian languages and hence developing the OBR is not a straight forward task for Indian vernaculars as compared to English language, as the basic character set of all the Indian Languages is more than 64 and to denote a character two Braille cells are required.

## 3. PROPOSED METHOD

The objective here is to develop an optical Braille character recognition system which takes the different resolution of the scanned Braille document into considerations. In this direction an adaptive algorithm has been designed for differentiating the recto and verso dots and an algorithm for adaptive grid construction has also been proposed. Further, an algorithm for converting the Braille text to Hindi text has been proposed.

### Algorithm1: Adaptive recognition of Recto/Verso dots

Step1: Convert RGB image to gray image.

Step2: Use thresholding to retain only the shadow pattern

अ a (1)	आ ā (345)	इ i (24)	ई ī (35)	उ u (136)	ऊ ū (1256)	ऐ e (26)	ए ē (15)	ऐ ai (34)	ओ o (1346)	ओ ō (135)	औ au (246)	ऋ ṛ (5, 1235)	ॠ ṝ (6, 1235)	ऌ ḷ (5, 123)	ॡ ḹ (6, 123)	क k (13)	ख kh (46)	ग g (1245)	घ gh (126)	ङ ṅ (346)	च c (14)	छ ch (16)	ज j (245)	झ jh (356)	ञ ñ (25)	ट ṭ (23456)	ठ ṭh (2456)	ड ḍ (1246)	ढ ḍh (123456)	ण ñ (3456)	त t (2345)	थ th (1456)	द d (145)	ध dh (2346)	न n (1345)	प p (1234)	फ ph (235)	ब b (12)	भ bh (45)	म m (134)	य y (13456)	र r (1235)	ल l (123)	ळ ḷ (456)	व v (1236)	श ś (146)	ष ṣ (12346)	स s (234)	ह h (125)	क्ष kṣ (12345)	ज्ञ jñ (156)	ड r/ṛ (12456)	ढ ṛh (5, 12456)	फ f (124)	ज़ z (1356)
---------	-----------	----------	----------	-----------	------------	----------	----------	-----------	------------	-----------	------------	---------------	----------------	--------------	---------------	----------	-----------	------------	------------	-----------	----------	-----------	-----------	------------	----------	-------------	-------------	------------	---------------	------------	------------	-------------	-----------	-------------	------------	------------	------------	----------	-----------	-----------	-------------	------------	-----------	-----------	------------	-----------	-------------	-----------	-----------	----------------	--------------	---------------	-----------------	-----------	-------------

Fig 2: Hindi Braille Notations

$$T_h = 255 - [(255 - \max(I(i, j))) * \alpha + \beta] \quad (1)$$

$$\forall 1 \leq i \leq M, 1 \leq j \leq N$$

where,  $T_h$  = Threshold value

M = Number of rows, N = Number of columns

Through empirical observation  $\alpha$  is set as 10 and  $\beta$  as 5.

Step3: Preprocessing: Use Median filtering to remove the left out noise after thresholding.

Step4: Binarize the image to find the centroid.

Step5: Apply connected component labelling using the eight connectivity property to account for number of dots present in the document [10].

Step6: Find the centroid of all the labelled components.

Step7: Find the dot with maximum number of eight connected pixels. Let M denote the number of eight connected pixels.

Step8: Crop the image to set only the dot with maximum number of eight connected pixels using the equations given below

$$X1 = \text{Centroid } x1 - M/10, Y1 = \text{Centroid } y1 - M/10 \quad (2)$$

$$X2 = \text{Centroid } x1 + M/10, Y2 = \text{Centroid } y1 + M/10 \quad (3)$$

where,  $x1$  and  $y1$  are the centroid co-ordinates of the dot with maximum eight connected pixels, and  $X1$ ,  $X2$ ,  $Y1$  and  $Y2$  defines the cropping region.

Step9: Plot the horizontal and vertical projection profiles for the cropped dot.

Step10: Find the dot height from horizontal projection profile and dot width from vertical projection profile. Horizontal profile is the sum of all the pixels in the row direction and Vertical profile is sum of all the pixels in the column direction.

Step11: Then the size of the adaptive mask is defined as

$$X_{top}(i) = \text{Centroid } x(i) - (\text{Width}/2) \quad (4)$$

$$X_{bottom}(i) = \text{Centroid } x(i) + (\text{Width}/2) \quad (5)$$

$$Y_{left}(i) = \text{Centroid } y(i) - (\text{Height} * 5/3) \quad (6)$$

$$Y_{right}(i) = \text{Centroid } y(i) + (\text{Height} * 5/3) \quad (7)$$

where,  $0 < i < \text{number of objects}$  (i.e. number of dots),  $X_{top}$ ,  $X_{bottom}$ ,  $Y_{left}$  and  $Y_{right}$  defines the x and y co-ordinates of the mask region.

Step12: The designed mask from the above step is placed over all the dots such that the center of the mask coincides with the centroid of the dot and within the mask region each and every pixel is checked to see whether it is less than the threshold value ( $t_h$ ).

$$t_h = \max(I(i,j)) - \max(I(i,j))/5, \forall 1 \leq i \leq M, 1 \leq j \leq N \quad (8)$$

Step13: If the count of any mask is less than  $2/5 * \text{height} * \text{width}$  then such a dot is recognized as recto dot else it is a verso dot.

Step14: Using the centroid co-ordinates information separate images are constructed for recto and verso dots. The separated image contains only the centroid pixel of each dot. Fig.3 shows the sample inter-point Braille document. Fig.4 and Fig.6 depict the recto and verso dots extracted images respectively.

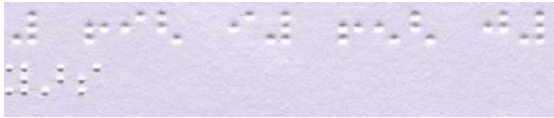


Fig 3: Sample Inter-point Braille image



Fig 4: Output of the dot recognition phase



Fig 5: Recto dots extracted image



Fig 6: Verso dots extracted image

### Algorithm2: Adaptive Grid construction

Grid construction plays a crucial role in conversion of the recognized Recto/Verso dot to its decimal equivalent. To

achieve this an algorithm has been designed which is independent of the resolution of the Braille document being used.

Step1: Read the Recto dots extracted image.

Step2: Identify the location of absent dots as all the Braille cells may not contain 3x2 information. An adaptive grid construction is used for this purpose. This uses the logic that, if any one of the centroid pixel is ON in a particular row, then the column pixels of that particular row are also made ON provided the centroid pixels in the columns of the rows below it are also ON.

Step3: Dilate the centroid pixels of Recto dots using the disc shaped structural element.

Step4: Super impose the dilated image of Recto dots over the grid constructed image.

Step5: At this stage apply horizontal and vertical projection profiles on the super imposed image to find the number of lines and cells in the Braille document respectively. Step6: Divide by 3, the total number of peaks in the horizontal profile to find the number of lines and divide by 2, the vertical profile to find the number of cells as each Braille cell is represented as a 3x2 matrix.

Step7: Perform the AND logic between the superimposed image and the image containing only the centroid pixel of Recto dot. If the product is true it indicates the presence of a dot and represent its value as 1 and if the product is false it indicates the absence of dot and represent its value as 0.

Fig.7 and Fig.8 show the adaptive grid constructed images for recto and verso dots respectively



Fig 7: Adaptive grid construction for Recto dots.



Fig 8: Adaptive grid construction for Verso dots.

Step7: Represent each cell by 6 bit string depending on the presence or absence of the dot in a cell. Convert this string to its equivalent decimal value using the expression:

$$d_6 + d_5 * 2 + d_4 * 4 + d_3 * 8 + d_2 * 16 + d_1 * 32 \quad (9)$$

The dots  $d_1$  to  $d_6$  are numbered as shown in Fig.1 (a). For example in the Braille cell shown in Fig.9, the presence of dots are represented by black pixel and absence of dots by white pixel. The binary equivalent for this cell is 101010 and its equivalent decimal code is 42.

Step8: Flip the verso dots extracted image as it corresponds to the back side text of the Braille document and apply the similar procedure as applied to recto dots to find its decimal equivalent.

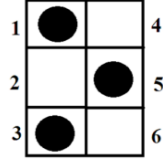


Fig 9: Braille Cell after recognition

### Algorithm3: Braille text to Hindi text transcription

Hindi Braille follows the expanded format for representing the vowel-consonant combined character. It means, the combined character दे is written separately as द and ए in Braille notation. The reader should not read them as separate characters; rather consider them as single character written in two boxes. So the complexity for conversion of Braille to normal Hindi text increases. The recognizer has to identify each character as a standalone vowel or consonant or consonant-vowel combined character.

Fig.10 shows the Braille equivalent of the Hindi word देवनागरी. Fig.11 depicts the text equivalent of the Braille cells.



Fig 10: Hindi Braille version

द ए व ना आ ग र ई

Fig 11: Expanded form of Hindi word

Step1: Check whether the read decimal code corresponds to vowel or not. If a vowel, map it to its equivalent character.

Step2: If it corresponds to a consonant, then check for the following decimal code.

Step3: If the following decimal code corresponds to a consonant then map the previous consonant to its equivalent character.

Step4: Else if, the following decimal code corresponds to a vowel, then add the previous consonant with the present vowel and map to its equivalent consonant-vowel character.

Step5: Repeat the process till the end of the decimal equivalent of the Braille document is reached.

The conversion from Braille text to natural text is carried out using the .net tool.

## 4. EVALUATION OF THE ADAPTIVE OBR

In this section, the materials used for evaluating the algorithm, the performance measures used to evaluate the adaptive algorithm and the results obtained are presented. The proposed algorithm have been applied on Braille documents with varied resolution. We have generated the Braille documents with spatial resolutions 2121x1501, 1380x985 and 685x503. For evaluating the performance of the proposed algorithm we have considered the three events. False Negative Rate (FNR) is the fraction of recto dots erroneously detected as verso dots. False Positive Rate

(FPR) is the fraction of verso dots erroneously detected as recto dots. True Negative Rate (TNR) is a fraction of recto/verso dots which are classified as non dot in the dot extracted image. The accuracy of the proposed method is calculated using the expression:

$$\text{Accuracy} = [1 - (\text{FNR} + \text{FPR} + \text{TNR})] * 100 \quad (10)$$

Table.1 gives the average FNR, FPR, TNR and average accuracy for Braille documents with varied resolutions. The proposed system possesses the excellent detection rate with average computation time of 12 seconds per document. The errors crept in are attributed to the deteriorated Braille dots in the document.

Table 1: Performance evaluation

Resolution	Data sets	FNR	FPR	TNR	Accu-racy
2121x1501	10	0.003	0.002	0.001	99.4
1380x985	10	0.004	0.004	0.002	99
685x503	10	0.006	0.004	0.003	98.7

## 5. CONCLUSION

In this paper we have recommended the adaptive segmentation technique using the concept of connected component labelling, and projection profile for segmenting the inter-point braille dots into recto and verso dots. This has the ability to extract the recto and verso dots of Braille documents with varied resolution. The proposed method is faster than any conventional approaches, and it can be extended with no further modifications to any Braille document irrespective of the language and the grade. In the existing work only the fine quality of the printed Braille document are considered without any overlapping of dots. The proposed method can be protracted to include the documents with deteriorated and overlapping dots.

## 6. ACKNOWLEDGEMENT

We would like to thank Council of Scientific and Industrial Research (CSIR), New Delhi, India for providing the financial backing for this work under the research scheme No. 22(0613)/12/EMR-II.

## 7. REFERENCES

- [1] J. Mennens, L. Van Tichelen, G. Francosis and J. Engelen, "Optical Recognition of Braille Writing Using Standard Equipment", *IEEE transactions of rehabilitation engineering*, Vol. 2, No. 4, December 1994.
- [2] R .T. Ritchings, A. Antonacopoulos and D .Drakopoulos, "Analysis of Scanned Braille Documents", In: Dengel, A., Spitz,

A.L. (eds.): Document Analysis Systems, *World Scientific Publishing Company*, pp. 413–421, 1995.

[3] Y. Oyama, 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.

[4] C.M. Ng, V.Ng and Y.Lau, “Regular feature extraction for recognition of Braille”, *Third International conference on computational Intelligence and Multimedia Applications, IEEE*, New Delhi, India, pp. 302—306, Sep 23-26, 1999.

[5] A. Antonacopoulos and D. Bridson “A Robust Braille Recognition System”. Document Analysis Systems VI, A. Dengel and S. Marinai (Eds.), *Springer Lecture Notes in Computer Science*, LNCS 3163, pp. 533-545, 2004.

[6] N. Falcon, C. M. Travieso, J. B. Alonso and M. A. Ferrer, "Image Processing Techniques for Braille writing Recogniton", *EUROCAST*, LNCS 3643, pp. 379-385, 2005.

[7] A. Malik Al-Salman, Y. ALOHAI, M. Alkanhal and A. Airajith. “An Arabic Optical Braille Recognition System”, *ICTA*, Hammamet, Tunisia, Apr 12-14, 2007.

[8] A. Malik S. Al-Salman, A. El-Zaart, Y. Al-Suhaibani, K. Al-Hokail and A. O. Al-Qabbany. “An Efficient Braille Cells Recognition”, 6<sup>th</sup> International conference on Wireless communication, Networking and Mobile computing, *IEEE*, Chengdu, China, Sep 23-25, 2010.

[9] A. Al-Saleh, A. El-Zaart and A. Malik Al-Salman, “Dot Detection of Braille Images Using A Mixture of Beta Distributions” , *Journal of Computer Science*, pp-1749-1759, 2011.

[10] L. Di Stefano and A. Bulgarelli, "A Simple and Efficient Connected Components Labeling Algorithm". *Proceedings ICIAP*, *IEEE*, Venice, Italy, pp. 322-327, 1999.

[11] WorldBrailleUsage, UNESCO. National Library Service for the Blind and Physically Handicapped. Library of Congress. Washington, D.C, USA, 1990.