

# Recognition of Double Sided Amharic Braille Documents

**Hassen Seid Ali and Yaregal Assabie**

Department of Computer Science, Addis Ababa University, Addis Ababa, Ethiopia  
Email: {hssnseid@gmail.com, yaregal.assabie@aau.edu.et}

**Abstract**—Amharic Braille image recognition into a print text is not an easy task because Amharic language has large number of characters requiring corresponding representations in the Braille system. In this paper, we propose a system for recognition of double sided Amharic Braille documents which needs identification of recto, verso and overlapping dots. We use direction field tensor for preprocessing and segmentation of dots from the background. Gradient field is used to identify a dot as recto or verso dots. Overlapping dots are identified using Braille dot attributes (centroid and area). After identification, the dots are grouped into recto and verso pages. Then, we design Braille cell encoding and Braille code translation algorithms to encode dots into a Braille code and Braille codes into a print text, respectively. With the purpose of using the same Braille cell encoding and Braille code translation algorithm, recto page is mirrored about a vertical symmetric line. Moreover, we use the concept of reflection to reverse wrongly scanned Braille documents automatically. The performance of the system is evaluated and we achieve an average dot identification accuracy of 99.3% and translation accuracy of 95.6%.

**Index Terms**—Amharic Braille Recognition, Direction Field Tensor, Double Sided Braille, Recto Dot and Verso Dot Identification, Braille Code Translation

## I. INTRODUCTION

Braille is a tactile format of written communication for people with low vision and blindness worldwide since its inception in 1829 by Louis Braille [1]. It is a system of writing that uses patterns of raised dots to inscribe characters on paper [2]. A Braille cell consists of 6 dots, 2 across and 3 down, which is considered as the basic unit for all Braille symbols as shown in Fig. 1(a). The 6 dots totally give  $2^6 = 64$  different possible combinations of Braille character. This represents a single character, for example in English and Arabic languages [2, 3, 4]. Although most countries adopt and define their Braille code so as to fit into their local language characters, Braille systems used in the world currently are categorized into two levels as Grade 1 Braille and Grade 2 Braille [3]. Grade 1 Braille is a form of braille in which print characters are represented by one Braille cell using a mode indicator character [3]. The mode indicator determines how the character is to be read. For instance,

in the English Braille, the lower case letters ‘a-z’ and the major punctuation symbols are represented by a single braille character or Braille cell, but others such as upper case letters, digits, and italics are represented with ‘shift’ character as an indicator [5]. Grade 2 Braille is introduced as a result of the rigorous attempt to minimize the volume of Braille documents by contracting words so as to minimize the time required to read a Braille document as compared to Grade 1 Braille document [3]. In Grade 2 Braille, context sensitive rules which are apparently language dependent and frequently used letter groups are used for the contraction of words. These rules determine the correspondence between one or more Braille cells and the print characters. For example, in Standard English Braille, a Braille symbol may stand for ‘dis’ referring the word distance when it comes at the beginning of a word and ‘dd’ referring the word ladder when it comes at the middle of a word [5]. A Braille document can be embossed not only on a single side of the Braille document but also on both sides to overcome space consumption by single sided Braille documents. On double sided Braille, the embossing process is done with slight diagonal offset to prevent recto and verso dots interference [6, 7] as shown in Fig. 1(b).

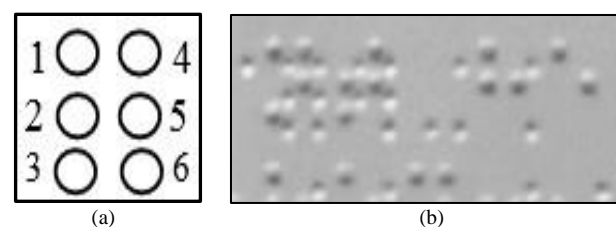


Fig.1. (a) Braille cell, (b) double sided Braille document.

Although the invention of Braille writing system is a big stride for visually impaired people, it still limits the communication to be only among them. As a result, an optical Braille recognition (OBR) system is introduced to convert Braille text into print text. Accordingly, several OBR systems have been developed over the years for various languages [1, 2, 3, 8]. In general, Braille image recognition systems have four major processes: *image acquisition and preprocessing*, *segmentation*, *feature extraction* and *recognition* [9]. Braille document images are acquired by digital devices like scanner and this is usually followed by preprocessing with the purpose of enhancing the quality of the Braille image [2, 9]. Segmentation involves identification and separation of

embossed Braille cells from the background [10]. During feature extraction phase, salient features of the Braille cells that would be used for recognition are computed [11, 12]. Finally, recognition is done by matching a set of features of Braille cells against each record in a translation table. In general, recognition of double sided Braille documents is more difficult than that of single sided as the identification of recto and verso dots in double sided Braille documents brings in additional tasks in each phase of the recognition system [7].

Braille writing system was introduced to Ethiopia in 1923 and a number of Amharic Braille documents have been produced since then [13]. With an effort to overcome space consumption, many of the Amharic Braille documents are embossed in both sides of the document. Thus, this paper presents the development of recognition system for double sided Amharic Braille documents. The remaining part of the paper is organized as follows. Section II presents related works in the area of optical Braille recognition. In Section III, we present the characteristics of Amharic Braille system. The proposed recognition system for double sided Amharic Braille documents is presented in Section IV. Section V presents experimental results, and conclusion and future works are highlighted in Section VI.

## II. RELATED WORK

Research and development on recognition of Braille documents has been reported since the 1980s [14]. Although Braille dots have similar embossing across different scripts, the way they are encoded to form equivalent print characters vary from one script to another. For example, a single cell containing six dots represents a single Latin character whereas two or more cells may be required to represent a single character in other scripts depending on the number of characters used by the scripts. This leads to the necessity of designing specific methods for recognition of Braille documents embossed for each script. As a result, various Braille document recognition methods have been proposed over the years for different languages.

One of the earliest works on recognition of Braille documents was conducted in 1985 by François and Calders [14]. The system was designed to convert Braille characters into their equivalent Latin characters with relatively complex constraints on the setup of the camera and lighting. Other attempts were made during the 1990s with the aim of reducing the constraints [2, 5, 15]. Ritchings *et al.* [7] made significant improvement by designing a system that uses a commercially available flatbed scanner to acquire a grayscale image of a Braille document. Furthermore, they considered the recognition of double sided Braille documents. More recent works on recognition of Latin Braille characters focus on improving the overall performance of the system and it has now reached an accuracy of over 99% [6, 16].

Attempts were also made to recognize Braille documents into non-Latin texts. Al-Salman *et al.* [3] conducted a research on recognition of double sided

Arabic Braille document. In this work, Braille images scanned using flatbed scanner were converted to grayscale and image thresholding algorithm was developed to examine the value of each pixel so as to classify a pixel into one of the three classes (dark, light and background) based on variation of intensity level. Using two threshold values (*low* and *high*, calculated from grayscale values of the image), dot detection algorithm was developed to classify a dot as recto (contains bright region at the top and dark at the bottom) or verso (contains dark region at the top and bright at the bottom). Finally, recognition was made by translating each Braille cell into its corresponding Arabic character. Furthermore, Al-Shamma and Fathi [17] developed a system that transcribes the results of Arabic Braille recognition into text and voice.

Jiang *et al.* [8] developed a system that segments Mandarin Braille words and transforms to Chinese characters. Word segmentation consists of rule, sign and knowledge bases for disambiguation and mistake correction using adjacent constraints and bi-directional maximal matching, and segmentation precision is reported to be better than 99%. The overall translation accuracy of Mandarin Braille codes to Chinese characters for common documents was 94.38%. On the other hand, several languages of India use a largely unified braille script for writing known as Bharati Braille [18]. Bharati Braille recognition has recently received attention from researchers and developers. Accordingly, a database of Bharati Braille document image was developed to assist the effort in the development of Indian Braille recognition system [18].

Hassen and Assabie [1] conducted a research on recognition of Amharic Braille characters in single sided documents using direction field tensor [19] for noise removal and isolation of braille dots from the background. A half character detection method, which differentiates braille dots from noise, was applied and braille cells were formulated by way of examining horizontal distances between half characters. The system was also designed to determine braille dot sizes automatically, which would enable the recognition system to be resilient to differences in the size of braille dots. It was reported that the system achieved an average accuracy of 98.5% for single sided documents.

In general, preprocessing and segmentation tasks in optical Braille recognition systems may not depend on the type of scripts (or languages) as the embossing technology is the same across various scripts. However, feature extraction and recognition steps rely on the types of scripts as the number of Braille cells representing a character in natural languages varies from one script to another. Accordingly, research and development conducted on recognition of other Braille documents cannot be directly applied for recognition of Amharic Braille documents because the techniques and algorithms used to encode Braille cell into Braille code and translation of the code into print text are different. Furthermore, to our best knowledge, there is no published work on the recognition of double sided Amharic Braille

documents so far. Thus, this work focuses on the recognition of double sided Amharic Braille documents where we have dealt with the unique characteristics of Amharic Braille cell representation embossed on both sides of documents.

### III. AMHARIC BRAILLE SYSTEM

Amharic is the working language of the federal government and several states of Ethiopia which is currently estimated to have a population of about 100 million. Even though many languages are spoken in Ethiopia, Amharic is the *lingua franca* since it is spoken as a mother tongue by a large segment of the population and it is the most commonly learned second language throughout the country [20]. The present writing system of the language is derived from Geez. The alphabet consists of 34 base characters and six other orders making it a total of 238 characters, where a character represents syllable combination of a consonant and a vowel. For example, the base character ሀ (m ä) has the following six other orders: ሁ (mu), ሂ (mi), ሃ (ma), ሄ (me), ህ (mə) and ሆ (mo). In addition, Amharic language contains 44 labialization characters and punctuation marks. The language uses both Geez and Hindu-Arabic numerals to represent numbers.

The Amharic Braille system is organized into four fundamental groups: *consonants*, *vowels*, *numbers* and *punctuation marks*. Braille codes for consonants and vowels are represented each by a single Braille cell. Each character with syllable combination is derived from consonant and vowel, requiring two Braille cells. For instance, the character ‘U’ (h ä) is a syllable combination of consonant ‘U’ (h) and vowel ‘ኧ’ (ä). Thus, its Braille code is derived from ‘1:2:5’ and ‘2:6’ which are Braille codes of ‘U’ and ‘ኧ’, respectively as shown in Fig. 2 [21].

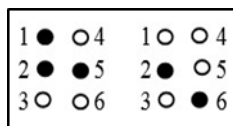


Fig.2. Braille code for character ‘U’ (hä).

In Amharic Braille system, the same Braille code may also be used to represent three print characters (Geez numbers, Hindu-Arabic numbers or Amharic characters) as shown in Table 1. In such a case, mode indicators are used before numbers whereas, in the absence of mode indicators, Braille cells are normally interpreted as Amharic characters. The mode indicator is a Braille cell that tells Braille readers to recognize if the next Braille cell is Geez or Hindu-Arabic number. The mode indicators are represented by Braille codes of ‘1:2:3:4:5:6’ and ‘3:4:5:6’ for Geez and Hindu-Arabic numeral systems, respectively [21]. Punctuation marks in Amharic Braille system are represented by a combination of up to three Braille cells.

Table 1. Braille codes representing multiple print characters.

Braille Code	Symbols Represented
1	1, ከ and አ
2:4	9, ከ and አ.
1:2	2, ከ and ብ
1:2:4	6, ከ and ቍ
1:2:4:5	7, ከ and ግ

### IV. THE PROPOSED SYSTEM

The proposed system architecture has six basic components: *Braille image acquisition and preprocessing*, *Braille dot segmentation*, *Braille dot identification*, *page identification*, *page transformation* and *recognition*. The general system architecture is shown in Fig. 3.

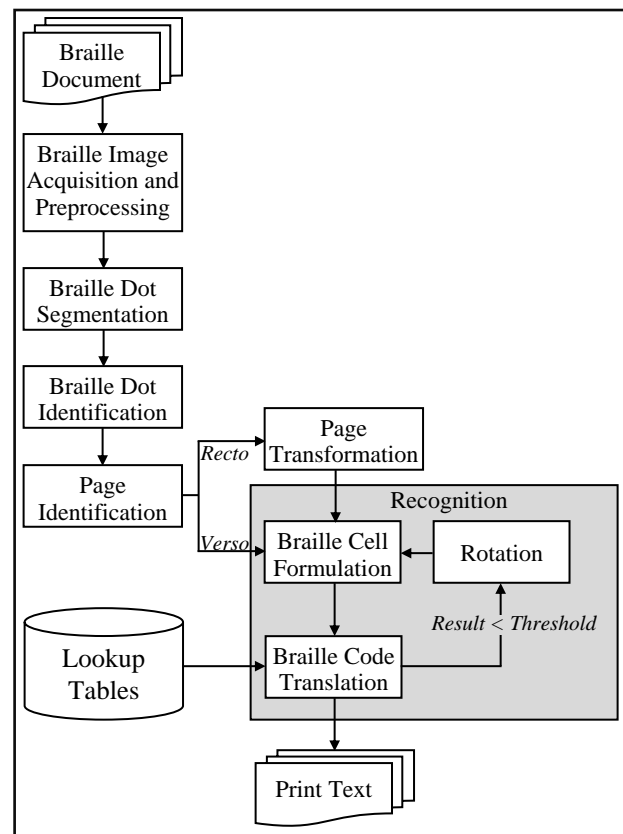


Fig.3. The proposed system architecture.

#### A. Braille Image Acquisition and Preprocessing

In this work, grayscale Braille images are acquired using flatbed scanner. In most low cost scanners, the document page is illuminated from an offset angle [6]. The direct implication for Braille documents is that the illumination of protrusions (recto dots) and depressions (verso dots) in that page will not be even. The face of protrusion or depression, which is angled towards the light source, will be more brightly lit and the face of protrusion or depression angled away from the light

source will be considerably less brightly lit [3, 6]. Accordingly, we exploited this property for recognition of double sided Amharic Braille documents. Fig. 4 shows grayscale image of double sided Amharic Braille document.

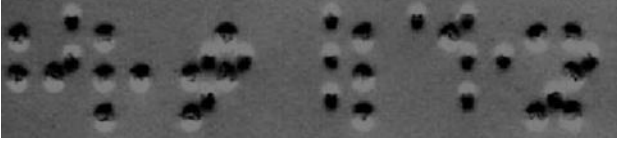


Fig.4. Double sided gray scale braille image.

After images are acquired, we employ preprocessing to make them more suitable for further processes. In this phase, our objective is to remove noise parts and highlight edges of Braille dots. To this effect, we use direction fields computed from structure tensor [19]. For a local neighborhood  $f(x,y)$  of an image  $f$ , the structure tensor  $S$  is computed as  $2 \times 2$  symmetric matrix using Gaussian derivative operators  $D_x$  and  $D_y$ .

$$S = \begin{pmatrix} \iint (D_x f)^2 dx dy & \iint (D_x f)(D_y f) dx dy \\ \iint (D_x f)(D_y f) dx dy & \iint (D_y f)^2 dx dy \end{pmatrix} \quad (1)$$

The integrals are implemented as convolutions with a Gaussian kernel. Direction fields can be estimated from the structure tensor using complex moments which are defined as follows [19].

$$I_{mn} = \iint ((D_x + iD_y)f)^m ((D_x - iD_y)f)^n dx dy \quad (2)$$

where  $m$  and  $n$  are non-negative integers. For our preprocessing task, the order of interest to us is  $I_{11}$  which is derived from Equation (2) as follows.

$$I_{11} = \iint |(D_x + iD_y)f|^2 dx dy \quad (3)$$

$I_{11}$  is a scalar value that measures the optimal amount of gray value changes in a local neighborhood of pixels. When integrals are implemented, due to convolutions with a Gaussian kernel, noise parts (non-linear structures) in the Braille document are suppressed. On the other hand, due to Gaussian derivative operators,  $I_{11}$  optimally highlights edges (linear structures) of Braille dots where high gray value change occurs in a local neighborhood of pixels. Fig. 5 shows  $I_{11}$  of double sided Amharic Braille image.



Fig.5.  $I_{11}$  of double sided Amharic braille image.

### B. Braille Dot Segmentation

Braille dot segmentation separates Braille dots from the background. It is an important step in the system as recognition performance heavily depends on accurate segmentation of Braille dots. To segment Braille dots, the resultant  $I_{11}$  image is used as an input and the algorithm developed by Assabie and Bigun [22] to segment characters from the background is effectively applied for Braille dot segmentation. Fig. 6 shows the result of Braille dot segmentation.



Fig.6. Braille dot segmentation

### C. Braille Dot Identification

Braille dot identification classifies segmented Braille dots (isolated and overlapping) as recto and verso dots. To this end, we use dot attributes (centroid and area) and  $I_{10}$  which is derived from Equation (2) as follows.

$$I_{10} = \iint ((D_x + iD_y)f) dx dy \quad (4)$$

In fact,  $I_{10}$  corresponds to the gradient field of the image. Each pixel in  $I_{10}$  (gradient field) has magnitude and direction (angle) representing pixel intensity changes. Fig. 7 shows  $I_{10}$  of double sided Amharic Braille where colors encode the direction of pixels in the HSV color space [23] with red, yellow, green, cyan, blue and magenta colors corresponding to 0, 60, 120, 180, 240 and 300 degrees, respectively. Fig. 7 (bottom) particularly shows  $I_{10}$  (gradient field) of verso and recto dots. The direction information of pixels that form Braille dot helps to classify whether the dot is recto or verso. Empirical evidence shows that, red and cyan colors in both recto and verso dots are minimal in the gradient field.

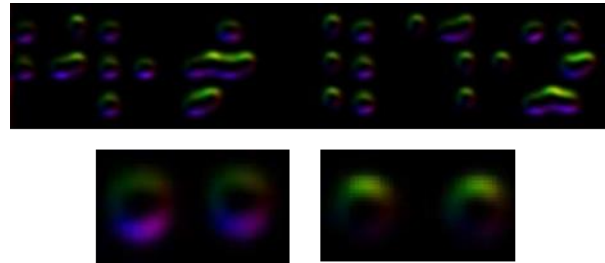


Fig.7. Top:  $I_{10}$  of double sided Amharic Braille; Bottom:  $I_{10}$  of verso dots (left) and  $I_{10}$  of recto dots (right).

However, we can see that pixels of recto dots are dominated by yellow and green colors (0-180 degrees) whereas verso dots are dominated by blue and magenta colors (180-360 degrees). This is so because of the differences in gray intensity variation observed as we go from top to bottom of each dot in the double sided grayscale Braille image (as shown in Fig. 4). It can be seen that the bright-and-dark pair appearance differs for recto and verso dots which results in different directions of pixels in the gradient field. Although  $I_{10}$  (gradient field)



is computed for the entire Braille image, for Braille dot identification, we consider only the pixels that form Braille dots. Thus, we used Algorithm 1 to group pixels that belong to Braille dots into two based on their direction.

Algorithm 1: Classification of Braille dots based on their direction.

```

Input:  $I_{10}$  (gradient field) of Braille image, segmented image
Output: Pixels of Braille dots categorized into two groups

1. Get Braille dot segmentation result
2. Get  $I_{10}$  (gradient field) of Braille image
3. If pixel is part of braille dot //check Braille dot segmentation
4.   If angle is  $\leq 180$  //green and yellow in  $I_{10}$ 
5.     PixelValue=0.5 //Group 1
6.   Else //blue and magenta
7.     PixelValue=1 //Group 2
8.   End If
9. Else
10.  PixelValue=0 //background of the Braille document
11. End If

```

Fig. 8 shows the grouping of Braille dots based on their direction where the gray intensities of Braille dots represent regions whose angles in the gradient field are less than or equal to 180 degrees and white parts represent Braille dot regions whose angles are greater than 180 degrees. During the process of Braille dot identification, the issue of overlapping Braille dots is a crucial concern as two or more dots can appear as physically connected with each other during Braille dot segmentation phase. The overlap may happen in the vertical or horizontal direction.



Fig.8. Braille dots extracted from gradient field.

The issue of overlap is solved by divide-and-conquer method. Vertically overlapping dots are segmented through analysis of the gradient field as the direction of pixels change from blue and magenta (180-360 degrees) to yellow and green (0-180 degrees) at the boundaries. Accordingly, in the resultant image shown in Fig. 8, we change these boundary points into background of the Braille document (thus vertically segmenting connected Braille dots) by assigning pixel values with 0 whenever a white pixel (value=1) is followed by gray pixel (value=0.5) as we go down each column. After removing vertical overlap, we develop algorithms that would recursively segment overlapping Braille dots. The algorithms are developed based on *area* and *centroid*.

*Area:* The area of a Braille dot corresponds to the number of pixels forming the dot. Of course, the area of Braille dot depends on the resolution used for scanning the document. With the Braille document scanned with a resolution of 200dpi, an isolated Braille dot ideally covers

an average area of approximately 250 pixels with verso dots tending to have larger areas than recto dots. On the other hand, the areas covered by overlap of two, three and four dots ideally have an average area of approximately 500, 750 and 1000 pixels, respectively as depicted in Fig. 9. Accordingly, we use Table 2 to identify isolated Braille dots and estimate the number of overlapping dots.

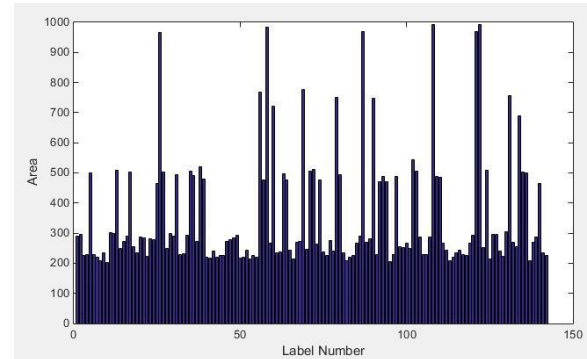


Fig.9. Areas covered by isolated and overlapping Braille dots.

Table 2. Estimation of overlaps using area

Dot Type	Area
Isolated dot	126-375
Overlap of two dots	376-625
Overlap of three dots	626- 875
Overlap of four dots	876-1125

*Centroid:* The centroid of a Braille dot (isolated or overlapping) locates the center of intensity distribution over the dot similar to locating the center of a mass. Once the number of overlapping dots is identified using area, the centroid provides information used to further identify the types of dots. Fig. 10 shows the position of centroids for isolated and overlapping dots.

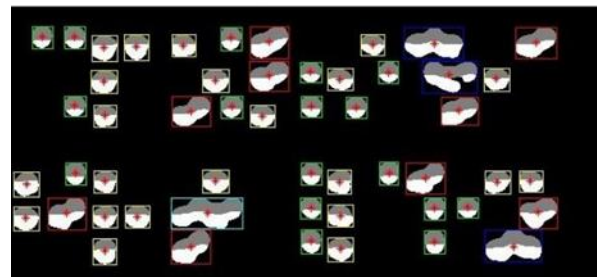


Fig.10. The position of centroids for isolated and overlapping dots.

For isolated Braille dots, the centroid of recto dots lies on the pixel whose angle is less than or equal to 180 degrees (pixel value=0.5 in Fig. 8) since recto dots are dominated by green and yellow. On the other hand, verso dots are dominated by blue and magenta (angle greater than 180 degrees) leading the centroid to lie on such colors in the gradient field. This corresponds to a pixel value of 1 in Fig. 8. However, this is not the case when two or more Braille dots are overlapping where segmentation is performed based on whether the number of overlapping dots is even or odd. When the number of

overlapping dots is odd, the centroid of overlapping dots is taken as the centroid of the middle dot. Accordingly, dots are partitioned into three regions: middle dot, left region and right region. By isolating the middle dot, it is analyzed to identify if it is recto or verso by considering the position of the centroid. When the number of overlapping dots is even, the centroid of overlapping dots is taken as the position where we segment the overlapping dots into two as left region and right region. Fig. 11 shows the arrangements of isolated and overlapping Braille dots along with their identifications in different scenarios.

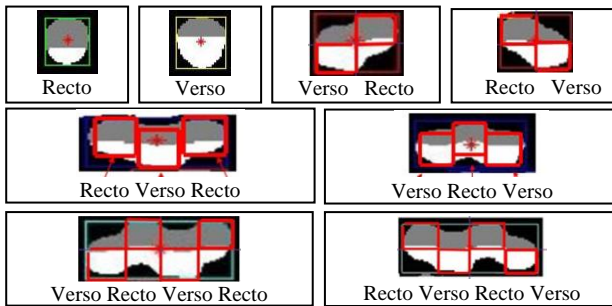


Fig.11. Arrangements of isolated and overlapping Braille dots

Left and right overlapping Braille dot regions obtained from the aforementioned processes are then recursively analyzed until all overlapping dots become isolated. Algorithm 2 shows the recursive process used to identify isolated and overlapping Braille dots. Examples of the segmentation and Braille dot identification results are shown in Fig. 12.

Algorithm 2: Procedure for Braille dot identification.

```

Input: Braille dot regions (along with areas and centroids)
Output: Identified Braille dots (recto and verso dots)
1. Get overlapping Braille dot region
2. Compute number of overlapping dots  $\rightarrow n$ 
3. Compute centroid
4. If  $n==1$  // if the dot is isolated
5.     If pixel at centroid==1
6.         Dot is verso
7.     Else // if pixel at the centroid ==0.5
8.         Dot is recto
9.     End If
10. Else
11.     If  $n$  is odd
12.         Horizontally partition the overlapping dot into  $n$ 
            equally distributed regions where we have  $(n-1)/2$ 
            overlapping dots to the left and another  $(n-1)/2$ 
            overlapping dots to the right of the middle dot
13.         Go to Line #3 to identify the middle dot
14.         Go to Line #1 to identify the left region
15.         Go to Line #1 to identify the right region
16.     Else // if  $n$  is even
17.         Horizontally partition the overlapping dot into  $n$ 
            equally distributed regions where we have  $n/2$ 
            overlapping dots to the left and another  $n/2$ 
            overlapping dots to the right of the centroid
18.         Go to Line #1 to identify the left region
19.         Go to Line #1 to identify the right region
20.     End if
21. End If

```

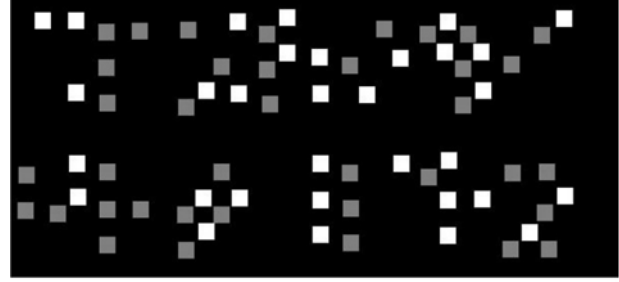


Fig.12. Identified recto and verso dots

#### D. Page Identification

For further processing, recto and verso dots which are found on the same page need to be separated in two different pages: recto and verso pages. Recto page contains only recto dots and verso page contains only verso dots. Thus, by considering the differences in pixel values of recto and verso dots, we identify separate recto and verso pages. Fig. 13 shows separated recto and verso pages constructed from the image shown Fig. 12.

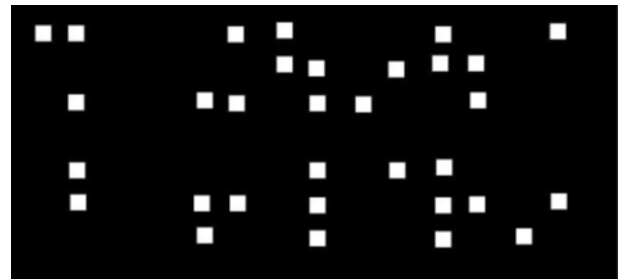


Fig.13. Separated recto (top) and verso (bottom) page.

#### E. Page Transformation

Human beings read Braille documents by touching the protruding dots in the verso page. In the case of double sided documents, reading of recto pages is performed by turning the page so as to make them verso. A similar procedure needs to be performed in automatic recognition of double sided Braille documents. This facilitates single *Braille cell encoding and translation algorithms* for both pages. After Braille pages are separated into recto and verso pages, recto pages are turned from right to left before Braille cells are formulated and encoded into a Braille code. In our case, this is accomplished using reflection of the recto pages about a vertical line constructed from the rightmost column of recto pages. Fig. 14 shows the result of transformation of the recto page shown in Fig. 13.

F. Recognition

Recognition of Braille documents is carried out using *Braille cell formulation* and *Braille code translation*. In the process of Braille cell formulation, up to six Braille dots are identified to construct a Braille cell. Braille cells are then encoded into a Braille code. This is achieved using Braille dot and Braille cell attributes shown in Table 3. The expected values are identified empirically and this is valid for Braille image resolution of 200dpi where the values are to be changed proportionally with the changes in resolution.

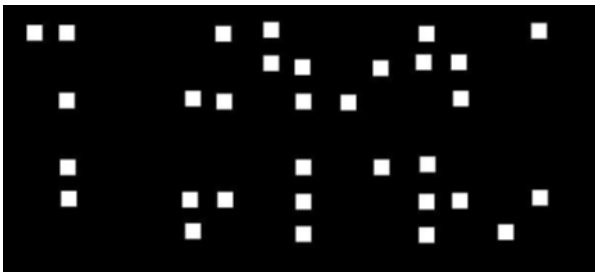


Fig.14. Braille transformation for recto page.

Table 3. Braille dot and Braille cell attributes

Attributes	Expected values in pixels
Dot width and height	12
Braille cell width	52
Braille cell height	90
Space between Braille cells	28
Space between Braille cell lines	44

It can also be seen that the horizontal and vertical spaces between Braille cells is typically greater than that of the Braille dots within the same Braille cell. Furthermore, a space amounting to a Braille cell is used to separate words in a text line. These characteristic features along with the Braille dot and cell attributes are used to identify not only Braille cells but also group Braille cells into words and text lines. After Braille cells are identified, the dots forming Braille cells are encoded into the respective six Braille codes based on their spatial positions. Fig. 15 shows the results of Braille cell identification and encoding.

After Braille cells are identified, the next step in the recognition process is to translate them into print text. Thus, Braille code translation is performed using four lookup tables that are designed based on Amharic Braille character organization. The lookup tables are organized as consonant, syllable (consonant-vowel combination), number (Geez and Arabic) and punctuation marks. The design avoids lookup tables for vowels and mode indicators because both of them are not printable symbols in Amharic language. Designing the lookup in four tables avoids ambiguity during translation because, in Amharic Braille system, different characters can be represented with the same Braille code as discussed in Section III. Subsequently, we applied Algorithm 3 to translate

Braille cells into equivalent print characters.

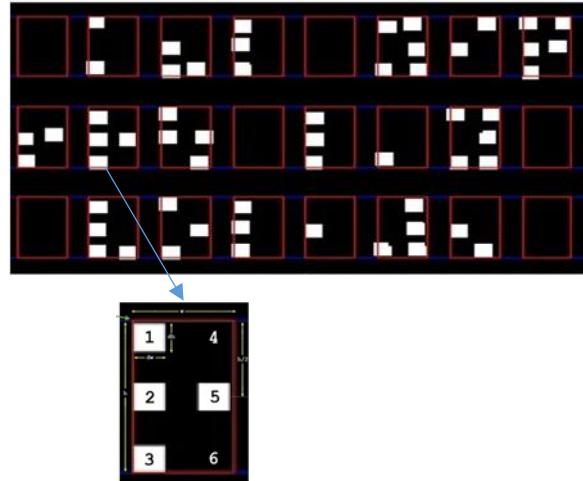


Fig.15. Braille cell identification (top) and encoding (bottom).

Algorithm 3: Procedure for translation of Braille codes.

```

Input: Braille code instances and lookup tables
Output: Print text
1. Get lookup tables: NUMBER, SYLLABLE, CONSONANT, PUNCTUATION
2. Get Braille code instance, code[i]
3. If code[i] is null
4.     Print space character
5. Else if code[i] is Geez number mode indicator
6.     Get code [i+1]
7.     Print the corresponding Geez number from NUMBER
8. Else if code[i] is Hindu-Arabic number indicator
9.     Go code [i+1]
10.    Print the corresponding Arabic number from NUMBER
11. Else if code[i] represents consonant
12.    Get code [i+1]
13.    If code [i+1] is vowel
14.        Concatenate code [i] and code [i+1] as new code
15.        Print the corresponding character from SYLLABLE
16.    Else
17.        Print the corresponding character from CONSONANT
18.    End if
19. Else if code[i] represents punctuation
20.    Print the corresponding symbol from PUNCTUATION
21. Else
22.    Code[i] is not an Amharic Braille code
23. End if
    
```

Braille recognition system is to be used by the sighted persons. However, most of them may not understand the correct layout of the Braille documents when placed on a scanner during the image acquisition phase. This may lead to turn the page upside down which results in a rotation of 180 degrees. To correct a reversed paper, we use a threshold of the translation results to determine if the page is rotated. The translation result that is considered here is not the actual recognition result but the percentage of Braille codes in a page whose corresponding entries are available in the lookup tables. Braille pages with translation results below a threshold value would be reversed by 180 degrees and Braille cells in the reversed page are reformulated for re-translation. In this experiment, a translation result of 68.5% was used as

a threshold value which was set by empirical analysis. The translation result of the reversed page is compared with the previous result and the better one is taken as the final result.

## V. EXPERIMENT

To test the performance of the proposed system, we developed a prototype using MATLAB programming tool. Furthermore, we use MySQL to manage the lookup tables that are used during Braille code translation. Experiments were then conducted using real-life double sided Amharic Braille documents.

### A. The Dataset

Double sided Amharic Braille documents were collected from Kennedy Library of Addis Ababa University, Ethiopian National Association for the Blind and Misrach Center. The dataset evenly contains good and bad quality documents. Braille images were then acquired using flatbed scanner with a resolution of 200dpi in “jpg” format.

### B. Test Result

The performance of the proposed system architecture is measured from two perspectives: dot identification and translation accuracy. Dot identification accuracy (IA) and translation accuracy (TA) of a Braille code into print text are calculated as follows.

$$IA = \frac{(CIRD + CIVD)}{\text{Total Number of Dots}} * 100\% \quad (5)$$

$$TA = \frac{\text{Correctly Translated Characters}}{\text{Total Number of Characters}} * 100\% \quad (6)$$

where *CIRD* is Correctly Identified Recto Dots and *CIVD* is Correctly Identified Verso Dots

Accordingly, the system achieved Braille dot (recto and verso) identification accuracy of 99.3% and an average translation accuracy of 95.6%. The quality of documents had significant effect on the performance of the system. For high quality documents, the system recognized Braille documents with an accuracy of above 99%. However, old and low quality documents posed difficulty for the recognition system. Furthermore, the report on the performance of our system takes overlapping dots into consideration, which is not addressed in many of the Braille document recognition systems developed for other scripts as well.

## VI. CONCLUSION AND FUTURE WORK

Many research and development works have been conducted on recognition of documents. Braille document recognition systems are designed to recognize Braille documents (single and double sided) for different languages. Our work is the first attempt on recognition of

double sided Amharic Braille documents. In this work, direction field tensor combined with gradient field show an interesting result as compared to prior works on different languages. Direction field tensor plays an important role in segmenting dots from the background and gradient field is used for identification of dots as recto and verso. This work also considers identification of overlapping recto and verso dots. The performance of the proposed system can be enhanced further by incorporating linguistic resources. A post processing component at word and sentence level is expected to enhance the performance of the recognition system. Accordingly, future work is recommended to be directed at looking into post processing activities and integration with text to speech synthesis systems.

## REFERENCES

- [1] Miftah Hassen and Yaregal Assabie, "Recognition of Ethiopic Braille Characters," In *Proceedings of the 4th International ACM Conference on Management of Emergent Digital Eco Systems*, 2012.
- [2] A. Antonacopoulos, "Automatic Reading of Braille Documents". In: H. Bunke, P.S.P. Wang (eds.): *Handbook of Character Recognition and Document Image Analysis*. World Scientific Publishing Company, pp. 703–728, 1997.
- [3] A. Al-Salman, Y. AlOhali, M. AlKanhah and A. AlRajih, "An Arabic Optical Braille Recognition System," in *Proceedings of the First International Conference in Information and Communication Technology and Accessibility (ICITA 2007)*, Hammamet, Tunisia, 2007.
- [4] W. Lisa, A. Waleed, and H. Stephan, "A Software Algorithm Prototype for Optical Recognition of Embossed Braille," in *the 17th conference of the International Conference in Pattern Recognition*, Cambridge, UK, August 2004.
- [5] P. Blenkhorn, "A System for Converting Braille into Print," *IEEE Transactions on Rehabilitation Engineering*, vol. 3, no. 2, pp. 215-221, June 1995.
- [6] A. Antonacopoulos and D. Bridson, "A Robust Braille Recognition System," In A. Dengel and S. Marinai (Eds.), *Document Analysis Systems VI*, Springer Lecture Notes in Computer Science, LNCS 3163, pp. 533-545, 2004.
- [7] R. Ritchings, A. Antonacopoulos and D. Drakopoulos, "Analysis of Scanned Braille Documents", In: A. Dengel and A. Spitz (eds.): *Document Analysis Systems*, World Scientific Publishing Company, pp. 413–421, 1995.
- [8] M. Jiang, X. Zhu, G. Gielen, E. Dr̄bek, Y. Xia, G. Tan and T. Bao, "Braille to print translations of Chinese", *Information and Software Technology* 44 (2), 91-100, 2002.
- [9] N. Falc3n, C. Travieso, J. Alonso and M. Ferrer, "Image Processing Techniques for Braille Writing Recognition", *EUROCAST*, LNCS3643, pp. 379-385, 2005.
- [10] A. Al-Saleh, A. El-Zaart and A. AlSalman, "Dot Detection of Optical Braille Images for Braille Cells Recognition" *ICCHP2008*, LNCS 5105, pp. 821–826, 2008.
- [11] C. Ng, V. Ng, and Y. Lau, "Regular Feature Extraction for Recognition of Braille," In *Proceedings of 3<sup>rd</sup> International Conference on Computational Intelligence and Multimedia Applications*, ICCIMA'99, 1999.
- [12] J. Bhattacharya and S. Majumder, "Braille Character Recognition using Generalized Feature Vector Approach", *Computer Networks and Intelligent Computing*, Springer, pp. 171-180, 2011.



- [13] Berihun Girma., "The Educational Situation of the Blind," *Center for educational staff development*, Addis Ababa, EThiopia, 1994.
- [14] G. François and P. Calders, "The reproduction of Braille originals by means of Optical Pattern Recognition", *Proc. 5th Int. Workshop on Computerized Braille Production*, Heverlee, pp. 119-122, 1985
- [15] J. Mennens, L. Tichelen, G. Francois and J. Engelen, "Optical Recognition of Braille Writing Using Standard Equipment", *IEEE Transactions of Rehabilitation Engineering*, 2(4): 207-212, 1994.
- [16] M. Yousefi, M. Famouri, B. Nasihatkon, Z. Azimifar and P. Fieguth, "A robust probabilistic Braille recognition system", *International Journal on Document Analysis and Recognition*, 15: 253, 2012.
- [17] S Al-Shamma and S. Fathi, "Arabic Braille Recognition and Transcription into Text and Voice", In *proceedings of 5th Cairo International Biomedical Engineering Conference* Cairo, Egypt, pp 227-231, 2010.
- [18] T. Shreekanth and V. Udayashankara, "A New Research Resource for Optical Recognition of Embossed and Hand-Punched Hindi Devanagari Braille Characters: Bharati Braille Bank", *IJIGSP*, vol.7, no.6, pp.19-28, 2015.
- [19] J. Bigun, T. Bigun and K. Nilsson, "Recognition by Symmetry Derivatives and the Generalized Structure Tensor," *IEEE TPAMI* 26 (2), pp. 1590-1605, 2004.
- [20] M. Lewis, G. Simons and C. Fennig, *Ethnologue: Languages of the World*, Seventeenth edition. Dallas, Texas: SIL International, 2013.
- [21] Special Education Team, "Braille Teaching Guide (in Amharic)", Curriculum Development and Research Institute of Ethiopia, Addis Ababa, Ethiopia, 1998.
- [22] Yaregal Assabie and J. Bigun, "Offline Handwritten Amharic Word Recognition," *Pattern Recognition Letters*, 32 (2011), pp. 1089-1099, 2011.
- [23] S. Thorstein, "CIE Div 1, R1-47 Hue angles of Elementary Colours," *International Commission on Illumination-CIE*, Norway, 2004.

### Authors' Profiles



**Hassen Seid Ali** received Master's Degree in Computer Science from Addis Ababa University, Addis Ababa, Ethiopia. He received Bachelor Degree in Computer Science from HiLCoE School of Computer Science, Addis Ababa. He also received Bachelor Degree in Physics from Bahir Dar University, Bahir Dar, Ethiopia. His

research interests are pattern recognition and digital image processing.



**Yaregal Assabie** received his PhD in Electrical Engineering from Chalmers University of Technology, Gothenburg, Sweden. He received Master's Degree in Information Science and Bachelor Degree in Computer Science from Addis Ababa University, Ethiopia. He is currently working as an Assistant Professor at the

Department of Computer Science, Addis Ababa University. His research interests are natural language processing, pattern recognition and digital image processing.

**How to cite this paper:** Hassen Seid Ali, Yaregal Assabie, "Recognition of Double Sided Amharic Braille Documents", *International Journal of Image, Graphics and Signal Processing(IJIGSP)*, Vol.9, No.4, pp.1-9, 2017.DOI: 10.5815/ijigsp.2017.04.01