Representation of Web based Graphics and Equations for the Visually Impaired

C.L.R. Gunawardhana, H.M.M. Hasanthika, T.D.G.Piyasena,S.P.D.P.Pathirana, S. Fernando, A.S. Perera, U. Kohomban

Abstract

With the extensive growth of technology, it is becoming prominent in making learning more interactive and effective. Due to the use of Internet based resources in the learning process, the visually impaired community faces difficulties. In this research we are focusing on developing an e-Learning solution that can be accessible by both normal and visually impaired users. Accessibility to tactile graphics is an important requirement for visually impaired people. Recurrent expenditure of the printers which support graphic printing such as thermal embossers is beyond the budget for most developing countries which cannot afford such a cost for printing images. Currently most of the books printed using normal text Braille printers ignore images in documents and convert only the textual part. Printing images and equations using normal text Braille printers is a main research area in the project. Mathematical content in a forum and simple images such as maps in a course page need to be made affordable using the normal text Braille printer, as these functionalities are not available in current Braille converters. The authors came up with an effective solution for the above problems and the solution is presented in this paper.

1. Introduction[1]

In this research our main focus is to make e-Learning a common platform for all learners without any discrimination between normal and visually impaired users. Graphics and web content are considered as an integral part of an e-Learning based learning process. Therefore, the accessibility to tactile graphics and mathematical content are the important requirements for visually impaired people. To support these requirements the authors focused on enabling direct image and mathematical content printing using normal Braille text printers. The solution describes methods of representing edge-based images using Braille symbols and braille text content printing. The integration of these features is done for the Moodlee-Learning system so that the visually impaired learner will have access to images and mathematical content in their learning process.

1.1 The Problem

Images are essential for learning subjects such as science, history and mathematics. Due to the inability to print images using normal text Braille printers, images from these text books are simply ignored and only the textual content is printed to be used by the visually impaired students. Due to this reason they have to learn history without having access to maps and they have to learn mathematics without the knowledge of some shapes used in mathematics or some complex equations.

In order to images accessible to visually impaired people the images should be converted into a tactile format.

Currently tactile image printing is supported by image embossers. The paper used in an image embosser is thicker than the paper used in a text Braille printer. The cost of embosser paper is much higher than that of normal paper. Some popular models of image embossers include the Index Braille 4X4 Pro embosser [1], Pictures in a Flash (P.I.A.F) tactile graphic embosser [2], View Plus [3] and Emprint Spot Dot Colour Braille printer [4]. Due to the high cost of these printers most of the developing countries cannot afford them and they simply ignore the image content when printing documents for the visually impaired.

Most of the formula editors represent MathML content as an image since some browsers such as Internet Explorer 6, Netscape 6 and older versions of Firefox do not support for MathML.Therefore, currently none of the Braille text printers can print the mathematical content in web based learning materials. Since images are not accessible to visually impaired people, mathematical content represented as images is also not accessible to them. Currently direct web content printing is not possible with the available Braille converters. Due to these reasons visually impaired people suffer from not being able to access image content in web based learning materials.

Braille converter is software that converts normal text into Braille text format.WinBraille [5] and Duxbury [6] are examples for converters and both of them are commercial software. They have inbuilt Braille text printing support. Before sending to the printer, a Braille text file should be converted into the Portable Embosser Format (PEF) [7] which is the standard file format for the Braille printers. Images are ignored by these tools and only the text is converted into the Braille text format.

[1]Department of Computer Science and Engineering, University of Moratuwa
Email : lanch.gunawardhana@gmail.com, malshanihm@gmail.com, dilukagp@gmail.com, dileepa88@gmail.com, shantha@uom.lk, shehan@uom.lk, upali@codegen.co.uk

A Braille letter is a 3×2 cell and by using dots in these cells we can represent different Braille letters. A Braille letter corresponds to a normal letter. Representation of Braille letters with the standard gap between each dot and the letters in different directions are shown in figure 1.

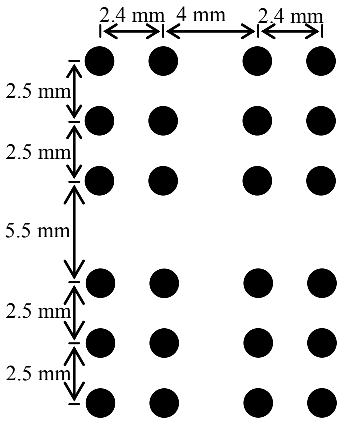

Figure 1: Braille character representation

1.2Problem Case

As our research case, we selected the School for Deaf and Blind, in Rathmalana [8]. Due to the unavailability of image printing functionality in text Braille printers, image content is ignored from the text books provided to the students of their school. Due to the high cost of thermal printers students have very limited access to images printed using them. Not only in Sir Lanka, this is a common problem faced by most of the developing countries, since they cannot afford these high cost embossers which cost more than 7 times a normal text Braille printer. According to the 2012 WHO fact sheet, there are 285 million visually-impaired people worldwide and 90% of them live in developing countries [9]. Therefore this is a problem for all of them.

As a solution for the above problem, we selected Moodle as the e-Learning system and added functionality for Braille image printing and forum post printing in Moodle. Integrated functionalities for image and mathematical printing have been tested with the teachers and students at the School for Deaf and Blind in Rathmalana.

2. Methodology&Implementation

2.1 Approach for printing images using normal text Braille printers

Text Braille printers can only convert text into Braille. In order to print an image using a text Braille printer, the image should be converted into a textual format. Therefore, if the image already has text embedded in it, those are extracted and removed from the image and it is provided separately from the image. Then the image is resized and converted into gray scale and the edge detection is done from the gray image. Then the textual conversion is carried out on the edge image. Once we have the textual image it will be converted to a XML format called Portable Embosser Format (PEF) and this will be sent to the printer for printing. Figure 2 represents the process of printing images using the Braille text printers.

Since our approach is to enable image printing functionality within Moodle, we modified the system to automatically add a link to print an image in Moodle whenever the teacher uploads an image. When a visually impaired user clicks on this link processing is done at the server side to convert the image into Braille textual image. Then the textual image will be returned to the client and the users can send this image to the printer using a User Interface(UI) tool we have created. Using the UI, users can select the required printer type, embosser type and other settings according to their preference. According to the selected printer settings the required PEF file will be generated.

In the UI tool selection of printer options should be supported by Non Visual Desktop Access (NVDA) [10] screen reader. Therefore we used Java Access Bridge [11] to get support from NVDA. Controls in this tool and the navigation can be manipulated by using 'Tab' key or 'Shift+Tab' key.

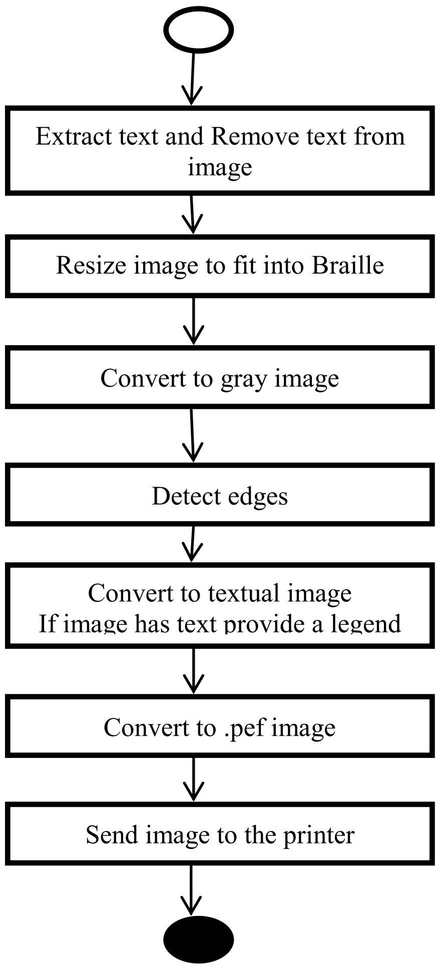

Figure 2: Process diagram for printing images using Braille text printers

2.1.1 Removal of text from image

If an image is converted with text embedded in it to a Braille format, a visually impaired person will have difficulties in identifying whether it is a part of the image or text as we print edges using Braille text. Therefore before printing images with text we should remove text content from the image and the text content will be provided separately as a legend.

For the text area identification we used Tesseract [12, 13] Optical Character Recognition Engine. By comparing OmniPage, Ocrad, OCRFeeder, and Tesseract [14] we selected Tesseract since it gives the best performance, the supportability with various input and output formats and the compatibility with different operating systems among other OCRs. In Tesseract, connected component analysis [15] is performed in order to get outlines of the components. These outlines are then gathered together into blobs, and these blobs are organized into text lines and broken into words based on definite spaces and fuzzy spaces. For the better identification of words it uses linguistic analysis and adaptive classification. Tesseract can be trained for multiple languages which make our solution expandable too.

The methodology of removing text from the image and providing it as a legend with the image will be described in detail in section *2.1.6*.

2.1.2 Resizing image to fit into the Braille paper

Standard Braille paper size is [84, 84] dots. After removing the margin, printable area is [84, 80] dots. Therefore we first resize the image into [84, 80] to fit the image into Braille sheet. We selected bicubic interpolation to resize image since it gives the best result among its nearest neighbors, bilinear and bicubic interpolations [16].

2.1.3Convertinginto gray scale image

If the image is a colour image, that should be converted into gray image before edge detection. A gray scale image only carries the intensity information. The gray level consists of black at the weakest intensity and white at the strongest intensity.

Colours in an image are converted to a shade of gray by calculating the effective brightness or luminance of the colour and using this value it creates a shade of gray that matches the desired brightness.

The effective luminance of a pixel is calculated using the following formula. We use Marvin framework for colour to gray scale conversion in the images [17].

$$Y = 0.3RED + 0.59GREEN + \ 0.11BLUE \text{------------}(1)$$

RED, GREEN and BLUE in equation (1) represent the pixel values of RGB space.

2.1.4 Detecting edges

Edge detection should be done in order to detect the boundaries of the image. The boundary generation of the image as edges is the most important factor for a visually impaired person since they cannot sense the colour information in an image. We looked at Canny, Laplacian of Gaussian(LOG), Robert, Prewitt and Sobel edge detection algorithms and compared the performance according to the previous researches done [18, 19] and selected Canny as the best algorithm.

Canny edge detection includes maximizing the signal to noise ratio, achieving good localization and removing the false edge detection as much as possible. In order to achieve this it smoothens the image with a Gaussian filter, calculates the gradient, applies a non-maximal suppression and performs the hysteresis thresholding.

2.1.5 Converting to textual image

In order to print using the Braille text printer, we should represent the image using a textual format. In order to execute this, for each 3×2 pixel in the edge image we identify a corresponding Braille letter. Since the Grade 0 Braille standard has one to one mapping, we used it for pixel to character mapping. Each 3×2 block of pixels in the edge image is considered to be a Braille cell, effectively rendering the binary image as a 28×40Braille cell grid. Figure 3 represents a part of the grid and it shows how the 84×80 dots divide into 3×2 cells and form a 28×40 grid. Each of these blocks contains pixel values with each pixel value having either 1 or 0. Each block in the image is then traversed from top to bottom and left to right. While traversing the cells, each 3×2 pixel is compared with Grade 0 Braille character representation. Since it consists of all possible 3×2 patterns, everything will map to the characters in Grade 0 Braille. If all the pixels in 3×2 cells are zero then it will be mapped to a space. Then the converted textual matrix is written to a text file. To get the basic idea of this process we referred a research which was done by Microsoft [20].

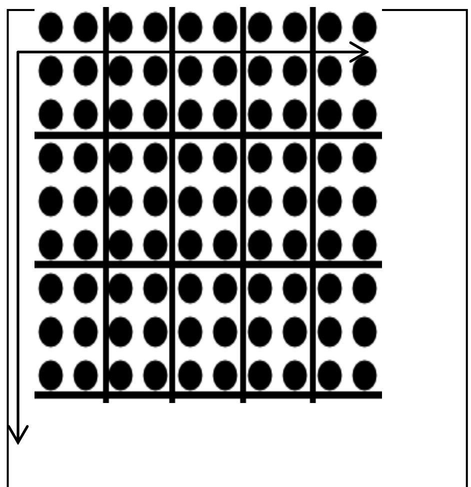

Figure 3: Representation of a part of the Braille cell grid with 3x2 pixels in each cell

Figure 4 represents the original image of a map and figure 5 represents the textual representation of the edge image.

Figure 4 : Original image

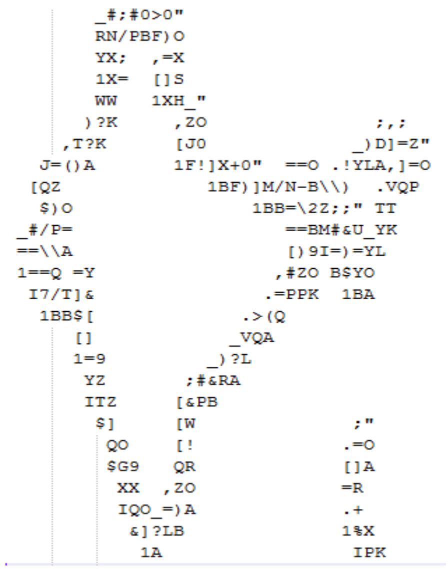

Figure 5: Textual image generated for figure 4

2.1.6 Processing images with text

If an image contains text, the text is removed from the image. For the character recognition we use Tesseract as described in *2.1.1*. This library was written in C++ and for our project we use a java wrapper for Tesseract as our image processing unit is written in java. First the original image is sent to be processed with Tesseractto identify text and the relevant positions of the text. Tesseract gives these positional information and words in the image as a HTML file. After filtering data from the HTML the text is removed from the original image.The original image is replaced with the new gray image after replacing the specific text containing pixels in the gray image to white pixels and then the words are printed below the image as a legend and numbers are added in the start position of each word.

If an image has text, text is detected before resizing and converting into a gray image for better results. In order to sense the image part and letters separately we print image edges using a unique letter while using a numbering technique to indicate the legend as shown in figure 7.

Since Tesseract has some false negatives and false positives, even though we cannot avoid false negatives we carry out filtering to avoid false positives as much as possible. Sometimes when the image has text and dots in the same line, these dots are also identified as text areas. These unwanted recognized words are removed by using regular expression patterns.

Tesseract uses trained data language files to identify words. The accuracy of the words depends greatly on the accuracy of the trained data, quality of the input image and the text embedded in it as well as the classification process done by Tesseract. For our system we used existing language files for the English language. However this needs to be represented by local language specific words with approximations and errors, as illustrated in Figure 8.

Figure 6 is the original image with embedded text and figure 7 is the image after text removal. Figure 8 is the textual image with a legend to provide textual data of the image.

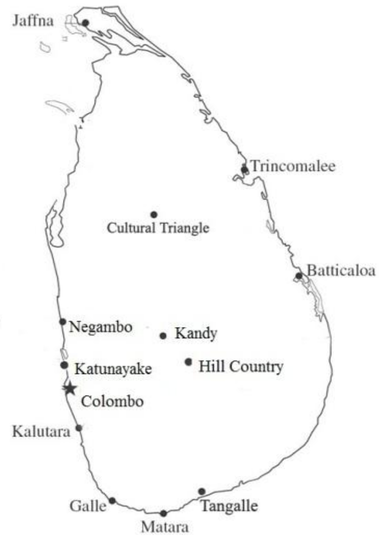

Figure 6: Original image with embedded text

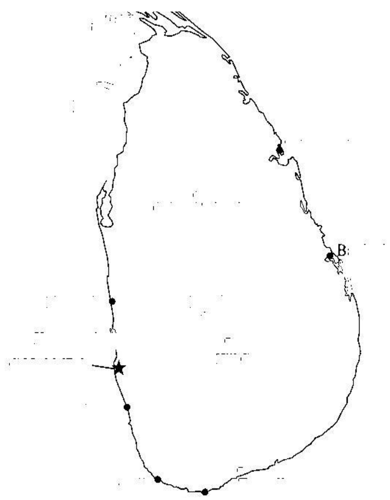

Figure 7: Image after removing embedded text

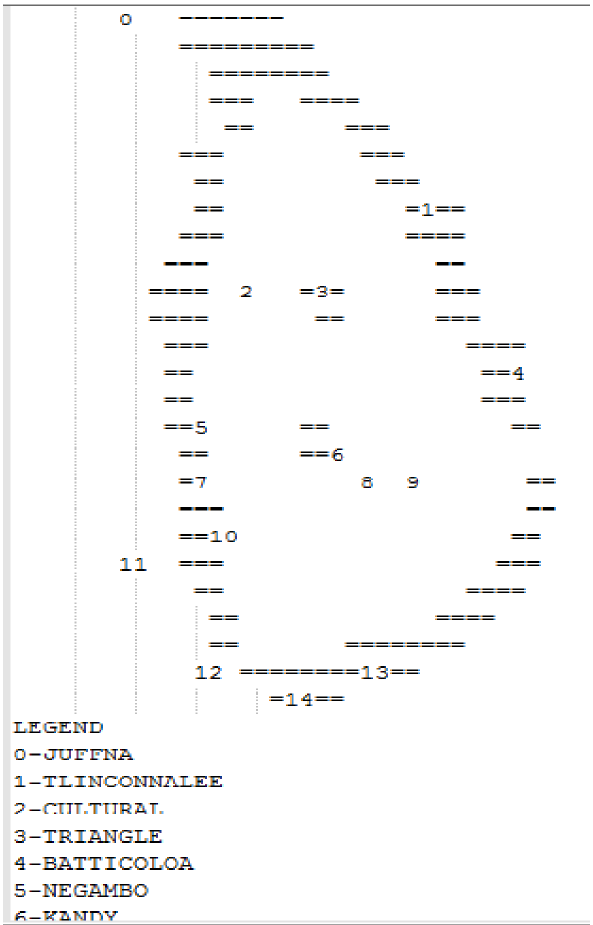

Figure 8: Textual image with providing text as a legend after removing embedded text

2.1.7 Converting to PEF format

PEF is a data format for representing Braille documents accurately and unambiguously regardless of language, location, embosser settings, and Braille code and computer environment [7]. PEF is supported by Braille embossers for embossing documents. It is the proper file format for Braille. PEF is an XML format and contains metadata about the document so that these metadata can be used by printer drivers to emboss the file accordingly. Same text files can be represented in different PEF formats to emboss based on different Braille standards.

We used open source jar archive called 'BrailleUtils' [21] to create the PEF format. Based on the printer type and the paper size we can specify the number of lines and columns to be printed as XML attributes. If image has texts, PEF file should be generated accordingly to print the image and the legend in 2 pages. Figure 9 shows the PEF format of the textual image shown in figure 5.

2.1.8 Sending PEF file to the printer

Once the PEF file is generated, the next process is to send it to the printer. For this process, a relevant printer driver and an embosser should be identified and the number of copies and the paper size should be specified. This task is done using a NVDA supported UI tool we have created. Currently we are supporting 53 types of the most popular embossers. Even if a new embosser type is introduced in the future, we can support that embosser by adding a new class for the embosser in Braille Util's embosser package which will set the embosser header and other relevant properties.

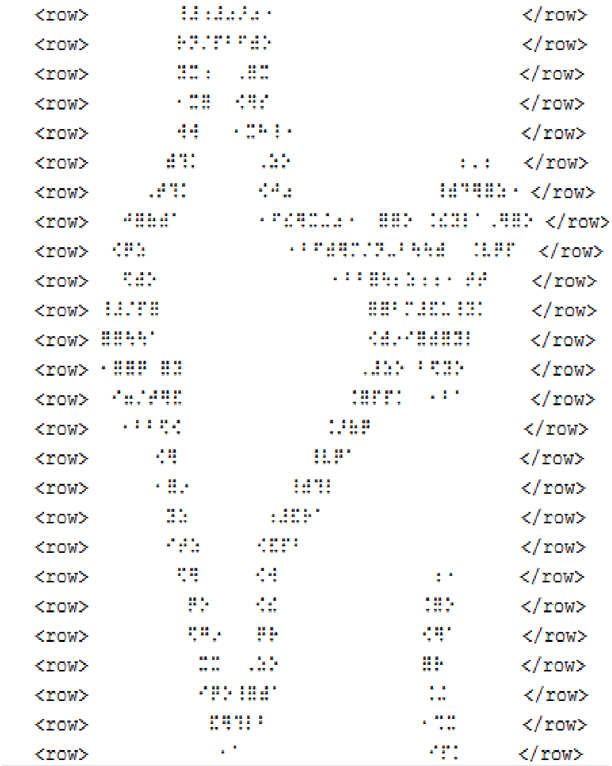

Figure 9: PEF file created for the textual image shown in figure 5

2.2Printing content in the Moodle posts

For HTML to textual transformation we selected an open source library named 'Liblouis' [22].Since it was entirely written in C, we used the Java Native API to access it. Liblouis ignores the images before converting HTML into text. Since most of the equation editors render mathematical content as images, we used WIRIS [23] as theMathML supported equation editor for Tiny MCE editor in Moodle. In WIRISeven though it renders the mathematical content as an image, in its alt attribute we can find the mathematical content represented in MathML. Therefore whenever an image with MathML content set to its alt attribute is found, we replace the image with MathML content in its alt attribute. If it is an image with description without MathML, the image is replaced with the description since the image can be printed separately. Then the filtered HTML content relevant to the post is converted into Braille text according to the relevant Braille standard and then the Braille text document is converted into PEF and finally sent to the printer.

2.2.1 Liblouis

Liblouis is an open source Braille translation and backend translation library [22]. It supports multiple languages and mathematical Braille such as Nemth [24]. It also supports different standards of a Braille language. Liblouis allows us to convert various document types such as XML and text into a Braille Resource Format (BRF).Both the translation and the formatting can easily be adapted to new languages and document formats. Even though there are some software solutions for Braille translation, the usability of these tools is limited as the translation rules differ from language to language. The reason for this is that different languages have different grammars. In order to build an extensible Braille translation system, developer should be provided with an easy interface to add these tables. In Liblouis, different language support is achieved using translation tables. In our case, to transform Moodle posts with mathematical content into Braille, we use the NemthBraille table.

3. Results and Discussion

3.1 Printing images

The above techniques produced clear edge image results for basic shapes and maps. Maps with texts which are not divided into areas also gave good image outline results and because of that the images were easily identified by the visually impaired people. Accuracy of text identification of the image depended on the accuracy of Tesseract. For some colour images with lots of colours and less colour variation differences could not be identified properly through Tesseract. As we noted the colour images give less accuracy than that of a black and white image. Since we resized the image into [84, 80] in order to fit it into the Braille sheet, if the image had a very high pixel density there was a considerable data loss and textual image was not clear. However for the black and white images in the pixel range less than [1000, 1000] it gave the user understandable edge images for maps and basic shapes.

We have tested these images with visually impaired teachers at the Rathmalana Blind School. When they touched the image for the first time, they began to read it as text. But once we said that it is an image, they began to scan the whole image with their hands to get an overall idea of the image and then they went through edges of the image printed in the form of Braille letters. The perceptional capacity of visually impaired students when identifying those images was high after describing the object. Currently maps and images are removed from their text books due to the inability of printing them using normal text Braille printers. But with our research we have printed several images of maps, basic shapes and some simple instruments used in science and made those images available in the blind school. Since we have integrated the image printing functionality with Moodle, teachers were able to upload some simple images and students were able to print them using normal text Braille printers.

3.2 Printing content in the Moodle posts

Currently if a visually impaired user needs to print web content, they should copy the content in the web pages and paste it to Braille translation software. However using this method they cannot print the mathematical content. Therefore in the Moodle system we enabled individual forum printing functionality with the support for mathematical content represented as MathML. Since most of the screen readers do not support MathML, the only means of accessing mathematical content is by printing them. Therefore this feature will be very useful for them.

Two major functionalities integrated into the Moodle for implementing an e-Learning solution that can be accessed by both normal and visually impaired users are discussed in this paper. Image content accessibility and web content accessibility will remove the barriers of a visually impaired learner in the context of e-Learning.

We save each image uploaded into Moodle using a unique name so that once the textual conversion of an image is done the same converted file can be reused to serve multiple requests for the same image. Converted images are deleted from Moodle after a 7 day period to avoid overloading the server with images.

Currently there are some limitations in our system. We can give accurate results only for black and white images with text such as images and basic shapes which do not contain much detail. With the enhancement of edge detection technologies and text extraction technologies we will be able to improve the system in future. An algorithm to detect whether the edge image identifies a fair amount of edges from the actual image is also important since there are false edge detections for some images.

4. Conclusion

In conclusion the authors would like to state that we have been successful in printing images and mathematical content using normal Braille text printers without the involvement of a sighted user. To the best of our knowledge this is the first system that can directly print images using normal text Braille printers without any aid of a sighted user.

In relation to this concept of converting images into textual format, there are many other application research areas that we can focus on. One is that, using this technique we can easily enable printing books with images. We can extract images, paragraphs and then for images we can do preprocessing to convert images into textual images. By having an intermediate format to convert document into Braille text and doing the necessary textual conversions for images we will be able to print documents with images which is currently not possible.

However for the image conversion part we would like to enhance capabilities such as printing colour images and images with high pixel density. Detecting the quality of the transformed edge image is another area that we are interested in. These features will be integrated to the product in the future.

References

[1]“Index Everest.” [Online]. Available: http://www.indexbrailleaccessibility.com/products/indexbraille/everest_v4.htm. [Accessed: 27-Jul-2012].

[2]“PIAF Picture in a Flash Tactile Graphic Maker - HumanWare USA.” [Online]. Available: http://www.humanware.com/en-usa/products/blindness/braille_embossers_and_writers/_details/id_147/piaf_picture_in_a_flash_tactile_graphic_maker.html. [Accessed: 20-May-2012].

[3]“ViewPlus Technologies Inc.,” ViewPlus Technologies Inc. [Online]. Available: http://www.viewplus.com/products/braille-printers/. [Accessed: 27-Jul-2012].

[4]“EmprintSpotDotColour Ink & Braille Printers.” [Online]. Available: http://www.viewplus.com/products/ink-braille-printers/emprint-spotdot/. [Accessed: 27-Jul-2012].

[5]“WinBraille.” [Online]. Available: http://www.indexbrailleaccessibility.com/products/indexbraille/winbraille.htm. [Accessed: 19-Mar-2012].

[6]“Duxbury Products - Duxbury Braille Translator.” [Online]. Available: http://www.duxburysystems.com/dbt.asp. [Accessed: 19-Mar-2012].

[7] “PEF Format | The Portable Embosser Format.” [Online]. Available: http://pef-format.org/. [Accessed: 19-Mar-2012].

[8] “The Ceylon School For The Deaf and Blind » About us.” [Online]. Available: http://csdeafblind.org/?page_id=2. [Accessed: 12-Jan-2012].

[9]“WHO | Visual impairment and blindness,” *WHO*. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs282/en/. [Accessed: 16-Jul-2012].

[10]“NVDA.” [Online]. Available: http://www.nvda-project.org/. [Accessed: 16-Jul-2012].

[11]“Installing Java Access Bridge.” [Online]. Available: http://docs.oracle.com/javase/accessbridge/2.0.2/setup.htm. [Accessed: 16-Jul-2012].

[12] R. Smith, “Tesseract OCR Engine,” *Word Journal Of The International Linguistic Association*, 2007.

[13] Smith, “An Overview of the Tesseract OCR Engine,” *Ninth International Conference on Document Analysis and Recognition ICDAR 2007 Vol 2*, vol. 2, pp. 629–633, 2007.

[14] A. Chitu, “Open-Source OCR Software, Sponsored by Google,” Open-Source OCR Software, Sponsored by Google. 10-Apr-2007.

[15]“Image Analysis - Connected Components Labeling.” [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/HIPR2/label.htm. [Accessed: 13-Jun-2012].

[16] R. Keys, “Cubic convolution interpolation for digital image processing,” *IEEE Transactions on Signal Processing, Acoustics, Speech, and Signal Processing*, vol. 29, no. 6, pp. 1153–1160, 1981.

[17]“Marvin Image Processing Framework.” [Online]. Available: http://marvinproject.sourceforge.net/en/index.html. [Accessed: 27-Jul-2012].

[18] D. H. A. Maini Raman, “Study and Comparison of Various Image Edge Detection Techniques,” International Journal of Image Processing, 2009.

[19] J. Canny, “A Computational Approach to Edge Detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

[20] M. B. Dias, M. K. Rahman, S. Sanghvi, and K. Toyama, “Experiences with lower-cost access to tactile graphics in India,” in Proceedings of the First ACM Symposium on Computing for Development, New York, NY, USA, 2010, pp. 10:1–10:9.

[21]“brailleutils - A cross platform utility package for embossing and converting PEF-files - Google Project Hosting.” [Online]. Available: http://code.google.com/p/brailleutils/. [Accessed: 18-Jul-2012].

[22]“Egli: Liblouis–A universal solution for Brailletranscription services - Google Scholar.” [Online]. Available: http://scholar.google.com/scholar?cluster=14334785390108851712&hl=en&as_sdt=2005&sciodt=0,5. [Accessed: 17-Jul-2012].

[23]“WIRIS editor demo - equations editor - educational mathematics.” [Online]. Available: http://www.wiris.net/demo/editor/demo/en/. [Accessed: 18-Jul-2012].

[24] “Dotless Braille and Nemeth Algebra.” [Online]. Available: http://www.dotlessbraille.org/dbnemeth.htm. [Accessed: 23-Jul-2012].