Sparsha: A Comprehensive Indian Language Toolset for the Blind

Anirban Lahiri[1], Satya Jyoti Chattopadhyay[2], Anupam Basu[1]

[1]Department of Computer Science and Engineering
Indian Institute of Technology
Kharagpur 721 302, India
{anirban, anupam}@iitkgp.ac.in

[2]Cal2cal India Limited
408, Jodhpur Park
Kolkata 700078, India
tutul_ch@rediffmail.com

ABSTRACT

Braille and audio feedback based systems have vastly improved the lives of the visually impaired across a wide majority of the globe. However, more than 13 million visually impaired people in the Indian sub-continent could not benefit much from such systems. This was primarily due to the difference in the technology required for Indian languages compared to those corresponding to other popular languages of the world. In this paper, we describe the Sparsha toolset. The contribution made by this research has enabled the visually impaired to read and write in Indian vernaculars with the help of a computer.

Categories and Subject Descriptors

D.0 **[Software] :** General

General Terms: Human Factors, Experimentation.

Keywords

Visual impairment, Braille, audio feedback, Indian languages.

1. INTRODUCTION

The advent of computer systems has opened up many avenues for the visually impaired. They have benefited immensely from computer based systems like automatic text-to-Braille translation systems and audio feedback based virtual environments. Automatic text-to-Braille translation systems are widely available for languages like English, French, Spanish, Portuguese, and Swedish [7, 26, 18, 16]. Similarly audio feedback based interfaces like screen readers are available for English and other languages [ref c, 8, 20]. These technologies have enabled the visually impaired to communicate effectively with other sighted people and also harness the power of the Internet.

However, most of these technologies remained unusable to the large visually impaired population in the Indian sub-continent [17]. This crisis can be attributed to primarily two reasons. First, the languages in the mentioned region differ widely from other popular languages in the world, like English.

These languages or vernaculars also use relatively complex scripts for writing. Hence, the technologies used for English and other such languages cannot be easily extended to these languages. Secondly, the development of these technologies for Indian languages, right from scratch, is not trivial as the various Indian languages also differ significantly amongst themselves.

The Sparsha toolset uses a number of innovative techniques to overcome the above mentioned challenges and provides a unified framework for a large number of popular Indian languages. Each of the tools of Sparsha will be discussed in detail in the following sections. Apart from English the languages supported by Sparsha include Hindi, Bengali, Assamese, Marathi, Gujarati, Oriya, Telugu and Kannada. The motivation for this work is to enable the visually impaired to read and write in all Indian languages. The toolset set has been named Sparsha since the word "Sparsha" means "touch" in Hindi, something which is closely associated with how Braille is read.

2. BHARATI BRAILLE TRANSLITERATION

Bharati Braille is a standard for writing text in Indian languages using the six dot format of Braille. It uses a single script to represent all Indian languages. This is done by assigning the same Braille cell to characters in different languages that are phonetically equivalent. In other words, the same combination of dots in a cell may represent different characters in each of the different Indian languages. However, a single character in an Indian language may be represented by more than one Braille cell.

The above mentioned characteristics of Bharati Braille code is illustrated in Figure 1. There are many other issues and rules related to Bharati Braille. These will be discussed in the following sections along with the methods used for implementing them.

Braille	a	k	T	" }
Hindi	अ	क	त	ढ़
Bengali	অ	ক	ত	ঢ়
Kannada	ಅ	ಕ	ತ	
Oriya	ଅ	କ	ତ	ଢ଼

Figure 1. Examples of characters in Indian languages and their corresponding Bharati Braille representation

2.1 Transliteration to Bharati Braille

As shown in Figure 1, characters from different Indian languages can be mapped to the same Braille representation. Thus, in order to implement this, the system uses separate code tables for each of the languages and depending on the users choice of input language the corresponding code table is used. The said method of implementation also makes the system highly scalable and allows the inclusion of more languages in future if required. For instance this technique is being used successfully to extend the system to include Urdu and Sinhala. This work is expected to be completed in the near future.

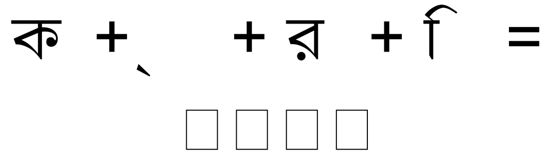


Figure 2. Formation of Conjugates

Another important aspect of Indian languages is the formation of consonant clusters or conjugates. In traditional hand written text this may be expressed conceptually as the first consonant followed by a special character called *halanth* which in turn is followed by the second character. The consonant cluster may again be followed by a vowel. However, the visual representation of such a consonant cluster or conjugate may be quite different from the visual representation of each of the individual consonants included in it, as shown in Figure 2. However, while translating the same text into Bharati Braille the special character *halanth* must precede both the consonants to be combined into a single conjugate.

The above constraints necessarily mean that the Braille translation for a particular character also depends on the sequence of characters preceding and following it.

Hence, in order to perform the tasks efficiently the system uses a finite state machine based approach similar to that of lexical analyzers [3, 6]. The mentioned approach also proves to be suitable for handling other issues associated with standard Braille translation like detection of opening and closing quotation marks, string of uppercase characters.

Apart from Indian languages the Sparsha system supports the translation of English language texts into grade 1 and grade 2 Braille. The system maintains a database of all standard Braille contractions which is used for generating grade 2 Braille. Furthermore, the system allows the user to add new contractions to the existing database.

The Sparsha system also supports the proper translation of a document containing text both in English as well as an Indian Language. According to standard Braille notations [11] the change in language is indicated through the proper use of the *letter* sign. However, a single document containing text in more than one Indian language cannot be translated into Braille, such that the reader is able to distinguish each of the languages correctly. This is due to the following reason. As mentioned previously the same Braille representation can refer to different characters in different Indian languages, this leads to the inherent ambiguity.

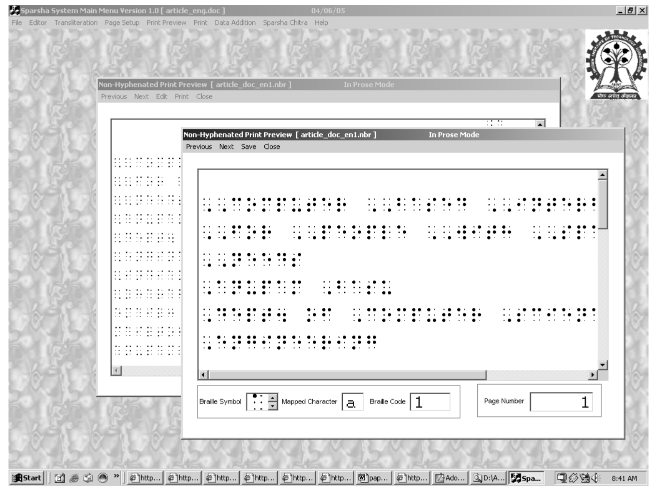


Figure 3. A screenshot of the interface for translating and editing Braille in the Sparsha system

2.2 Reverse transliteration

The Sparsha system allows reverse transliteration of Braille to text both for Indian languages as well as English. This allows the visually impaired to communicate seamlessly with other sighted people. The Braille code to be translated may be entered into the computer using a standard six key Braille keyboard. After translating the Braille code into text, the visually readable text may then be checked for correctness using a file reading system which will be described in later section.

In order to achieve reverse translation from Braille to text, the system uses a finite state machine based approach similar to that used for translating text to Braille as described previously. The task of reverse translation also uses the code tables corresponding to the language to which the text is being translated. Thus the system can easily be extended to other languages just by adding the corresponding code tables to achieve both forward and reverse translation.

2.3 Methods of Input - Output

Sparsha can accept English text, for translation, in the form of plain text files, HTML (hyper text markup language) files and Microsoft Word documents. Apart from English the Sparsha Braille translation system, as described, can take input text in Indian languages. This input can be given to the system in a number of forms as follows:

- ISCII (Indian Script Code for Information Interchange) [24] documents generated by applications like iLeap [10]
- LP2 documents generated by iLeap [10]
- Unicode text – generated by any standard editor supporting Unicode [25]. This technique will be discussed in detail in a later section

The output of the Braille translation can be obtained on a large variety of commercial Braille embossers [23]. The Sparsha system has been tested on the following Braille embossers:

- Index Basic-S
- Index Basic-D
- Index 4X4 PRO

- Braillo 400
- Modified Perkins Brailler [15]

Alternatively the output may be obtained on tactile Braille displays [1].

3. BRAILLE MATHEMATICS

At the time of this development there existed a few translators for converting mathematics to Braille [b]. However, these were found to be unsuitable for the visually impaired in the Indian sub-continent due to a number of reasons. Firstly the Braille code used for mathematics in India is slightly different from those used in other parts of the world [4], however, it bears close resemblance to the Nemeth code [5]. Secondly the interleaving of Braille mathematics with text in Indian languages was also not possible with the available systems. Thirdly many of these systems require a working knowledge of LaTex [13]. This cannot be expected from every user. Finally, most of these systems are unaffordable to the visually impaired in the Indian sub-continent.

The above mentioned reasons warranted the development of a mathematics-to-Braille translation system for the Indian sub-continent. The system thus developed can translate almost all mathematic and scientific notations. It also allows the user to interleave mathematic and scientific expressions with text in both Indian languages and English.

In order to allow the user to write complex mathematic and scientific expressions, the system provides a special editor for the purpose. The above mentioned editor is named “Nemeth editor” after Abraham Nemeth [t]. Thus the user is exempted from the task of learning LaTex. The editor provides a GUI (Graphic User Interface) as shown in Figure 4 for writing a mathematic or scientific expression in a form similar to that used by LaTex. This string can then be readily converted into Braille by the translation engine. However, the mathematical expression formed by the editor must be enclosed within a pair of special character sequences. This needs to be done so that when the mathematic or scientific expression is embedded within another English or Indian language text, it is properly translated to Braille using the standard for mathematic and scientific notation.

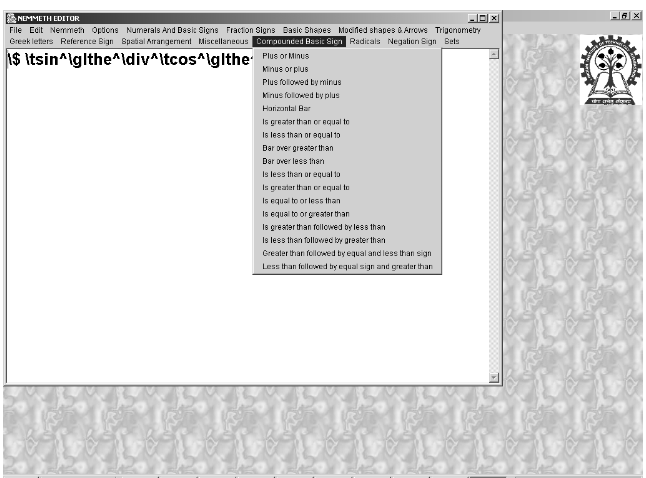


Figure 4. Screenshot of the Nemeth Editor

The selection of mathematical symbols and notations is done by the user in a menu driven fashion using the GUI. The set of all mathematic and scientific notations is partitioned in to separate collections, each consisting of similar notations. Alternatively the text may be entered by the user in a LaTex like format using any standard text editor.

4. SPARSHA CHITRA

Elementary tactile graphics is one of the best methods for introducing certain subjects, like geometry, to visually impaired students. However, such tactile graphics have remained outside the reach of the common man. This is due to the fact that sophisticated Braille embossers and expensive image conversion software are necessary for the purpose. Sparsha Chitra aims to provide relatively simple tactile graphics which can be obtained even by using low cost Braille embossers like the modified Perkins Brailler [15]. In other words no assumptions have been about any special feature of the Braille embosser being used. This allows tactile graphics to be embossed using just the Braille embossing capability of the embosser. The tactile graphics obtained from any image may be viewed and edited before finally being embossed. The image may also be scaled up or down to a size suitable for embossing. The system also allows the image color to be inverted in order to improve the contrast.

Sparsha Chitra takes its input in HTML format such that additional text can be included along with the tactile representation of the image. Sparsha is the feeling of touch and “Chitra” in Hindi means “picture” and thus this tool is named Sparsha Chitra.

The primary limitation of this tool is that complex images cannot be represented very clearly. However, the effect of this drawback is mitigated by the fact that the amount of detail that can be observed through touch is also limited. Furthermore the size of the tactile image is restricted by the bounds imposed by the sheet on which it is embossed. The functions for scaling the tactile image may prove to be useful in such a case.

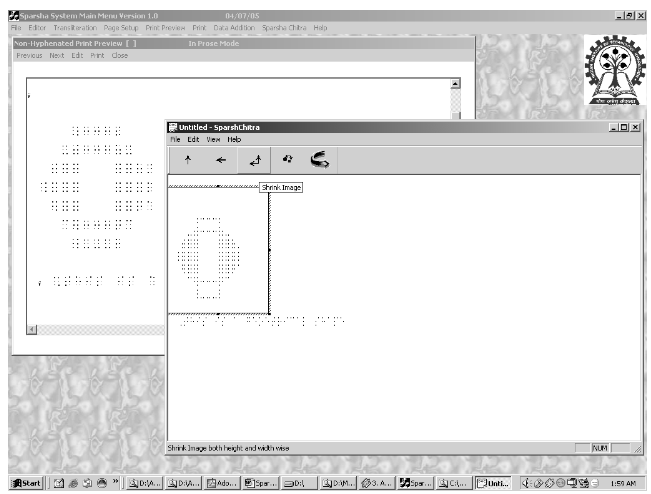


Figure 5. Screenshot of Spasha Chitra

5. FILE READER

5.1 Introduction

In order to enter text into the computer, in English, a visually impaired user can take the help of any standard screen reader [12, 8, 20]. Screen readers have proved to be vital to visually impaired computer users [19].Such screen readers are commercially available. However, such screen readers are not available for Indian languages.

This was primarily due to the reasons mentioned at the beginning of this paper.

The file reader which will be described in this section will redeem the situation and allow the user to type in text in Indian languages using Microsoft Word. For performing other tasks related to the operating system the user may use any of the standard screen readers. The construction of such a file reader requires a number of vital components [2]. These include text-to-speech engines for Indian languages, fonts for Indian languages, keyboard layouts for them, proper rendering engines and a text editor which can support Indian languages. Each of these components will be described briefly in the following sections. This will be followed by a description of the overall architecture of the system and its functioning.

5.2 Speech synthesis system

A speech synthesis system is vital for the functioning of any screen reader. It is responsible for producing human voice rendition of the text provided to it by the screen reader. In case of screen readers the speech synthesis system should be able to deliver the voice in real-time. This is necessary for visually impaired users to get instantaneous audio feedback.

A multilingual screen reader necessarily needs a speech synthesis system for each of the languages that it supports.

The mentioned file reader uses a speech synthesis engine for Indian languages called Shruti [22]. Shruti support two popular Indian languages namely Hindi and Bengali. It uses a method of di-phone concatenation for speech synthesis. This allows the speech synthesis system to produce reasonable real-time performance, at the same time maintaining a low memory space requirement.

5.3 Fonts and Rendering

There are number issues involved with Indian language fonts and their rendering. This is due to the fact that Indian language scripts are generally complex in nature. The Microsoft Windows system can be configure for correctly rendering these complex Indian language scripts. Correct rendering of fonts is achieved through the use of *Uniscribe* (Unicode Script Processor) and OTLS (OpenType Layout Services) libraries [9, 14]. Furthermore glyph substitution and glyph repositioning, as shown in Figure 2, are closely associated with the rendering of text in Indian languages. For this reason OpenType fonts have been found to be suitable for Indian languages as they carry, within the font file, explicit information about glyph substitution and glyph positioning. This maintained in the form of two tables namely GSUB (Glyph Substitution) and GPOS (Glyph Positioning).

5.4 Editor for Indian Languages

A number of text editor are available for Indian languages. Many of these editors are difficult to use and are non-intuitive. On the other hand it has been observed that Microsoft Word XP (Word 2002) performs reasonably well for Indian languages when proper fonts and rendering engines are used. Thus, Microsoft Word has been used instead of creating a new editor for the file reading application as shown in Figure 7. Microsoft Word also provides certain additional features which have been used extensively for the development of the file reader. These features have been discussed in detail in the following paragraphs. The use of Microsoft Word also motivates visually impaired users to switch to main stream applications and also eliminates the effort of learning another system.

Microsoft Word supports Unicode [25], hence it can accept text in any Indian language. However, in order to enter text in an Indian language in the Windows system a keyboard layout or IME (Input Method Editor) [21] for that language is required. Keyboard layouts are available for some popular Indian languages like Hindi. For other Indian languages it may have to be created. In our case a keyboard layout had to be created for Bengali.

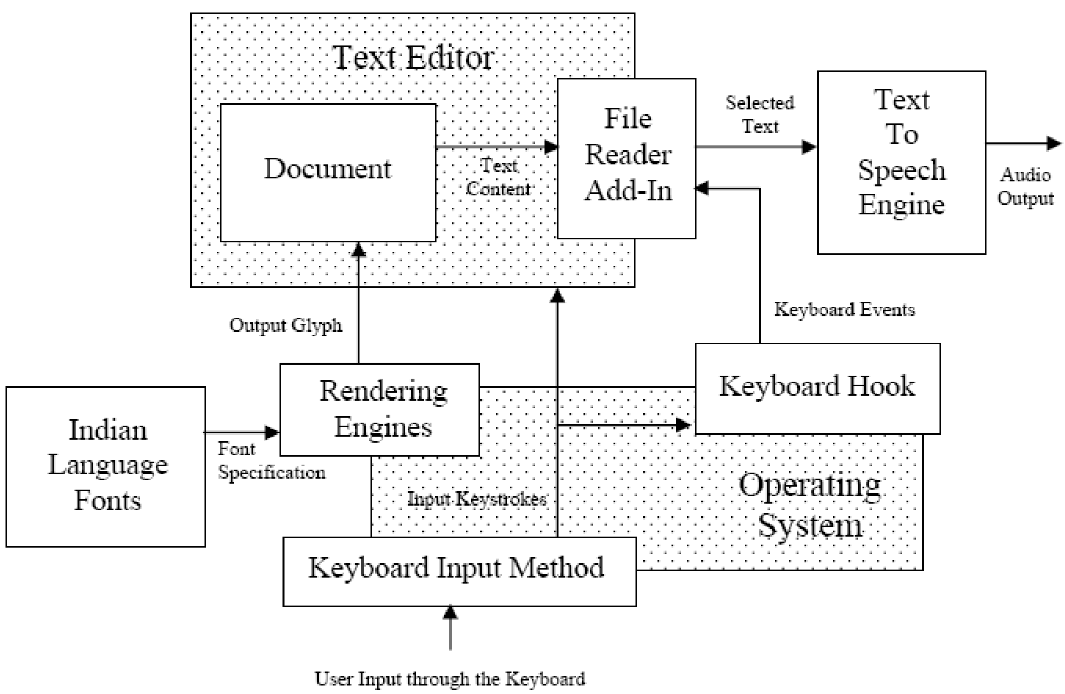


Figure 6. File reader System Architecture

The capabilities of the Microsoft Word can be extended using COM (Component Object Model) Add-Ins. Such Add-Ins are basically programs that run within the framework provided by Microsoft Word. The file reader has been developed in the form of such an Add-In. It interacts closely with the editor to provide necessary audio feedback for text in Indian languages. Such interaction takes place through the object model exposed by Microsoft Word. The file reader may be configured to start up every time Microsoft Word is used.

5.5 Overall System Structure and Operation

The overall architectural structure of the file reader system is shown in Figure 6. Most of the components of the system shown in the figure have been discussed in the last few sections. The interaction between the different components and how they operate as a system will be discussed in this section.

Keyboard hooks are placed within the operating system by the file reader Add-In. The keyboard hooks are responsible for trapping the keystrokes entered by the user through the keyboard. A copy of the entered keystrokes is passed to the file reader Add-In. The keystrokes are then passed to the keyboard layout or IME which is integrated with the operating system. The keyboard layout translates the keystrokes into Unicode characters and passes them to the editor.

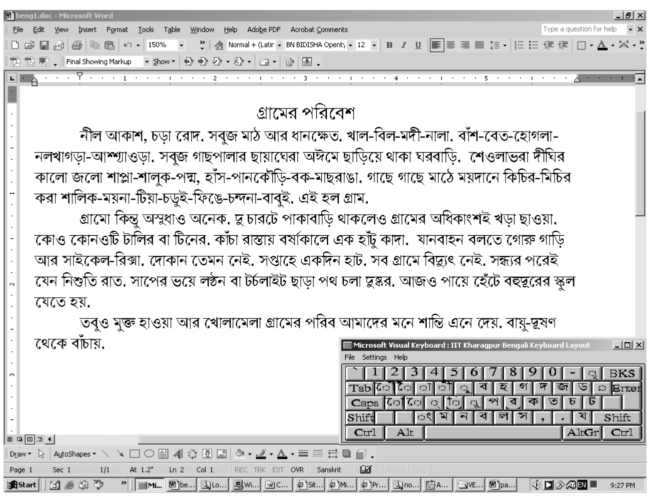


Figure 7. Screenshot of the file reader in operation using Microsoft Word

In the mean while the file-reader Add-In, on receiving the keystrokes, provides appropriate audio feedback by invoking the speech synthesis engine. Again, certain combinations of keystrokes are recognized by the file reader Add-In as special commands. These request the file reader to read out a certain portions of the text. This selected text is then passed to the speech synthesis system for producing human voice rendition. Thus providing an audio feedback based virtual environment for Indian languages. The file reader can be further extended to provide full screen reading functionality by using Microsoft Active Accessibility.

6. SYSTEM EVALUATION

6.1 Field Testing and Deployment

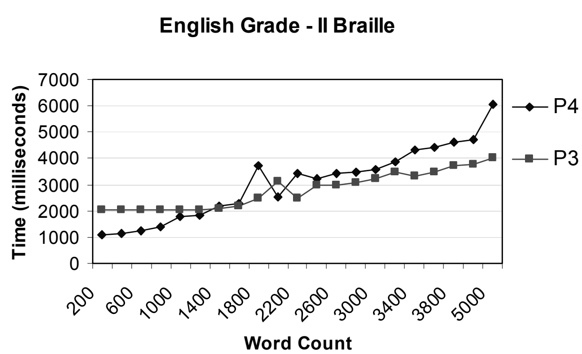
A subset of the Sparsha system known as the Bharati Braille Transliteration System* has been deployed by Webel Mediatronics Limited in a number of organizations for the visually impaired all over India as a part of a project sponsored by the Ministry of Communication and Information Technology, Government of India. As a result of these field tests the system underwent an iterative process of refinement to reach its current form. A plethora of request and suggestions from visually impaired users led to the development and inclusion of a number of additional features and tools that were added to the toolset. These include the Sparsha Chitra and the file readers for Indian languages. The process of continuous feedback helped the Sparsha toolset mature over the years. It also helped in weeding out many bugs and shortcomings of the initial versions of the system.

6.2 Obtained Results

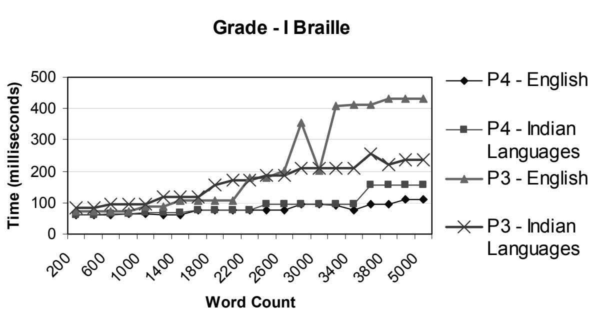
The Sparsha system is under a continuing process of use and evaluation. This feedback is being used to make the system more usable to the visually impaired and to enhance the features provided by the system. Training and deployment of the system has also been carried out at a number of premier organizations for the visually impaired. These include

- The National Association for the Blind, Delhi
- Blind Peoples' Association, Ahmedabad
- National Institute for the Visually Handicapped, Dehradun

The Braille translation system has been tested on a large number of computers in these organizations. The typical performance characteristics of the Sparsha Braille translation system is as shown in Figure 8. The performance characteristics have been measured for two different personal computer systems.



(a)



(b)

Figure 8. Graphs showing the computation time taken during Braille translation for (a) Grade – II English (b) Grade – I English and Indian languages

The computers have the following specifications (Processor Type, Primary Memory, Hard disk):

- Intel Pentiun 4 – 3GHz, 512MB, 80GB – referred to as P4
- Intel Pentium III – 550MHz, 256MB, 40GB – referred to as P3

The Sparsha Chitra tool was tested by visually impaired users and the obtained results are given in Figure 9. This was done by handing them sheets of paper Braille paper with tactile diagrams created by Sparsha Chitra and asking them to guess the image on the sheets.

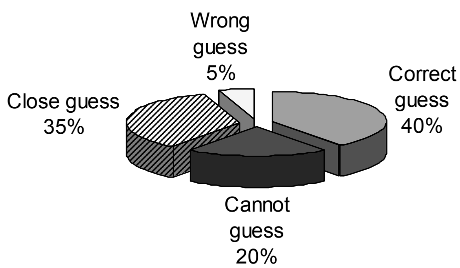


Figure 9. User response to tactile images generated by Sparsha Chitra

Most of these images were geometric figures. The majority of the guesses were correct while a large percentage of the guesses were very close like identifying a rectangle as square or a triangle as a mountain. Such misinterpretations often occur due to lack of color information or misjudging dimensions which is indeed quite difficult estimate from tactile representations.

The file reader tool needs extensive training before a naïve user can use it efficiently. Visually impaired users who are already familiar with Jaws or other screen readers can adapt to this system very quickly. This tool was primarily tested by visually impaired users having reasonable experience with Jaws.

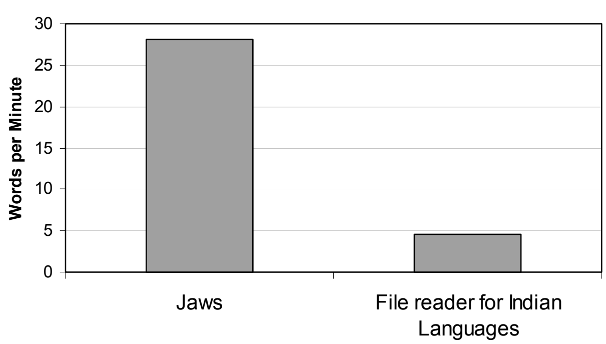


Figure 10. Comparison of the typing speed of a visually impaired user using Jaws and the Indian language file reader

The Indian language file reader could not be experimented with a large number of users since a good level of expertise with screen readers is required for using the file reader efficiently. The experimental results shown in Figure 10 and 11 pertain to a particular visually impaired user having some experience with Jaws. The experiments were carried out by dictating a paragraph of about hundred words to the user while he typed it into the computer using the Indian language file reader. However, this is only a preliminary experiment. It was also found that both the typing speed and the error rates improved significantly with practice.

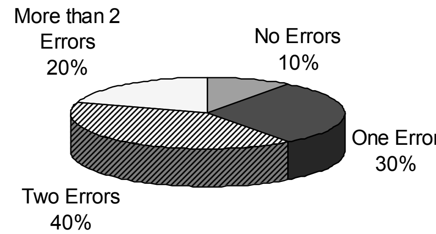


Figure 11. Number of words with errors for every ten words

7. LIMITATIONS AND FUTURE WORK

The Sparsha toolset is in the process of being extended to a number of other languages in the Indian subcontinent. This includes Urdu and Sinhala. The file reader in the Sparsha toolset is also limited by the availability of text to speech synthesis engines for all Indian languages. Hopefully these will be available in the near future and allow the system to be extended to more languages. It is also envisioned that the Sparsha system will be ported to mobile handheld systems. This will enable the visually impaired to communicate on the move. As of now, the required text to speech synthesis engines have been ported onto the Microsoft Pocket PC platform as well as on an ARM-Linux platform. It is only a matter of time before the file reader becomes functional on such mobile platforms.

8. CONCLUSION

The Sparsha system named after the feeling of touch has been the first attempt to help visually impaired users, in the Indian subcontinent, read and write in their native tongues. In this paper the various aspects of Indian languages and how they differ from other languages in the world has been explained. It also been discussed how these issues have been tackled in the Sparsha system.

The paper describes in depth the various tools included in the Sparsha toolset. These tools form a comprehensive toolset for Indian languages. It can be hoped that the Sparsha system would help increase the literacy rates among the 13 million visually impaired in the Indian subcontinent.

9. ACKNOWLEDGMENTS

The authors would like to thank Media Lab Asia for sponsoring a part of the work related to the file reader. The authors would also like to thank the National Association for the Blind, Delhi and many other organizations for the blind for their sustained help and cooperation during the entire development process. The authors owe special thanks to Mr. Samit Patra, Director, Electrosoft Consultants for his enormous help with many technical aspects of the work.

10. REFERENCES

[1] Basu Anupam, Roy S., Dutta P. and Banerjee S., “A PC Based Multi-user Braille.Reading System for the Blind Libraries”, IEEE Transactions on Rehabilitation Engineering, Vol. 6, No. 1, March 1998, pp.60—68

[2] Blenkhorn, P. “Requirements for Screen Access Software using Synthetic Speech”. Journal of Microcomputer Applications, 16, 243-248, 1993.

[3] Blenkhorn Paul, “A System for Converting Braille to Print”, IEEE Transactions on Rehabilitation Engineering, Vol. 3, No. 2, June 1995, pp. 215-221

[4] Braille Mathematics Code for India Manual, Prepared under the project “Adoption and Introduction of an Appropriate Braille Mathematics Code for India”, sponsored by UNICEF, Published by National Institute for Visually Handicapped, Dehra Dun and National Association for the Blind, Bombay, India

[5] Cranmer T. V. and Abraham Nemeth, A Uniform Braille Code, memo to the members of the BANA Board (January 15, 1991); Available at: http://www.nfb.org or http://world.std.com/~iceb/

[6] Das, P.K.; Das, R.; Chaudhuri, A., “A computerised Braille transcriptor for the visually handicapped”. Engineering in Medicine and Biology Society, 1995 and 14th Conference of the Biomedical Engineering Society of India. An International Meeting, Proceedings of the First Regional Conference, IEEE. 15-18 Feb. 1995 Page(s):3/7 - 3/8

[7] Duxbury Braille Translator, 2000, http://www.duxburysystems.com/products.asp

[8] HAL. Dolphin Computer Access, http://www.dolphinuk.co.uk/products/hal.htm

[9] Hudson, John for Microsoft Typography, “Windows Glyph Processing : an Open Type Primer”, November 2000, http://www.microsoft.com/typography/glyph%20processing/intro.mspx

[10] iLeap. Centre for Development of Advanced Computing. http://www.cdacindia.com/html/gist/products/ileap.asp

[11] International Council on English Braille (ICEB), Unified English Braille Code (UEBC) Research Project, http://www.iceb.org/ubc.html

[12] JAWS for Window. Freedom Scientific. http://www.freedomscientific.com/fs_products/software_jaws.asp

[13] Lamport L. LaTeX - A Document Preparation System, Addison-Wesley, 1985, ISBN 0-201-15790-X.

[14] Microsoft Typography, “Specifications : overview”, http://www.microsoft.com/typography/SpecificationsOverview.mspx

[15] Modified Perkins Brailler, Webel Mediatronics Limited. http://www.braille-aids.com/emboss.htm

[16] MONTY, VisuAide. http://www.visuaide.com/monty.html

[17] National Association for the Blind, India, 2002. Available at http://www.nabindia.org/sited/infor06.htm

[18] NFBTRANS. National Federation of the Blind, 2004, http://www.nfb.org/nfbtrans.htm

[19] Pennington C.A. and McCoy K.F., Providing Intelligent Language Feedback or Augmentative Communication Users, Springer-Verlag, 1998.

[20] Raman T.V. (1996). “Emacspeak – a speech interface”. Proceedings of CHI96, April 1996

[21] Rolfe, Russ “What is an IME (Input Method Editor) and how do I use it?” http://www.microsoft.com/globaldev/handson

[22] Shruti, Media Lab Asia Research Laboratory, Indian Institute of Technology, Kharagpur. http://www.mla.iitkgp.ernet.in/projects/shruti.html

[23] Taylor Anne, “Choosing your Braille Embosser”, Braille Monitor, October 200. Available at http://www.nfb.org/bm/bm01/bm0110/bm011007.htm

[24] Technology Development for Indian Languages, Department of Information Technology, Ministry of Communication & Information Technology, Government of India. Available at http://tdil.mit.gov.in/standards.htm

[25] Unicode. http://www.unicode.org

[26] WinBraille. Index Braille.

http://www.braille.se/downloads/winbraille.htm