AKSHAR: A mechanism for inputting Indic scripts on digital devices

Pranav Mistry
MIT Media Laboratory
20 Ames St. E15-320F
Cambridge, MA 02139 USA
pranav@media.mit.edu

Niranjan Nayak
Microsoft India Development Center
Microsoft Campus, Gachibowli
Hyderabad 500046 India
niranjan@microsoft.com

Abstract
Text input in Indian languages on digital devices presents unique challenges to the field of human-computer interaction. Part of the complexity arises due to the structure of the Indic scripts, which are derived from the ancient *Brahmi* script. Mechanisms to input text in Indian languages on mobile phone have been around for a while but none of these mechanisms are usable enough to emerge as the de-facto standard. These mechanisms fail to give clear justice to the structure, inherent to the *Brahmi* script that is radically different than the Roman and the Latin scripts. In this paper, we present a new text input mechanism that is based on the phonetic structure of the Indic scripts. The main goal of the research was to devise a new text input method that is targeted to the novice users and that can be learnt quickly, retained and fair speed can be achieved. Preliminary evaluation tests indicate that the new proposed method for inputting text in Indian languages is very intuitive and easy to use.

Keywords
Indic script input mechanism for digital devices, Brahmi script input method, text input method for mobile phone.

Introduction

Text entry is one of the most frequent human-computer interaction tasks. Although great strides have been made towards speech and handwriting recognition, typewriting and inputting text using key-presses remains and will likely be the main text entry method in the near future. Computers, mobile phones and other digital devices demand text input schemes that can be learnt quickly and retained to achieve a fair speed. These mechanisms should be easy to use rather than "hunt and peck".

Of India's nearly one thousand million people, a sizeable amount of over nine hundred million is currently excluded from participating in the so called information or digital revolution. One of the main reasons for this is the absence of appropriate input mechanism that can deal with information in the languages that majority of the Indians speak. On the other end, the mobile phone revolution has caught on in India. The numbers of mobile phones that are currently available in India hugely outnumber the number of desktop computers and other computing devices available in the country. However, majority of the mobile phones still have English as the input and interaction language. A small number of mobile phone manufacturers have come up with the support for a couple of Indian languages in their phones. However, inputting text using one of these mechanisms is a real challenge even for the experienced mobile phone users.

Text entry methods for Indian languages like Hindi, Gujarati, Marathi, Punjabi, Bengali and Telugu are not keyboard friendly and their entry using QWERTY keyboards is cumbersome and complicated; it involves the use of multiple shift modes. Furthermore, there are a number of competing keyboard designs and no national standard. QWERTY keyboards are not particularly amenable to accommodate the phonetic, non-alphabetic script like Indic. Many alternate layouts exist which are mapped on the Roman alphabets of QWERTY keyboards. All these approaches have fallen short. The probable reason for that is, without any consideration of the core structural features of Indic scripts, these methods have been localized from mechanisms for Latin or Roman based scripts. The same problem applies to the input methods for mobile phone and other interactive digital devices. Localization to Indic scripts is a non-trivial task due to the large number of alphabet set and the significant structural differences as compared to Roman or Latin based scripts.

Hence, there is a great need for an appropriate input mechanism/scheme for digital devices for various Indian languages. In this paper, AKSHAR is proposed as a generic input mechanism for inputting Indic scripts on digital devices like mobile phones, kiosk booths, WLLs, interactive TVs, and personal computers. Most of the scripts in India have their base in the ancient *Brahmi* script. Akshar uses the unique feature – phonetic grouping of 5 consonants- of Brahmi script as the central design element. The paper discusses the basic structure of Indic scripts, the AKSHAR mechanism and its unique features. User evaluation has shown that in terms of ease of use and learning, this new mechanism is notably better than any other input methods in use today.

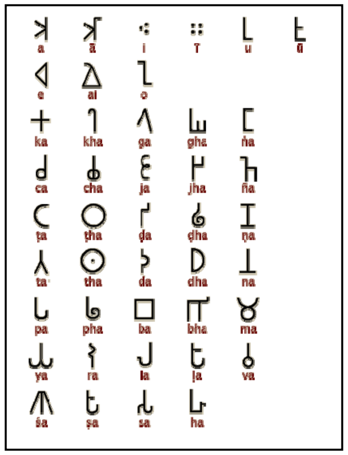


figure 1: Brahmi script alphabets.

Indic scripts and text input methods

Phonetic basis of Indian languages

India is a multilingual country with 23 officially recognized languages [2]. A vast majority of these languages have their orthography derived from the ancient *Brahmi* script. These scripts are cumulatively referred to as 'Indic Scripts.'

Panini's phonetic classification [2] of the Indian alphabets into vowels (V: A,, Aa…) and consonants (C: k Kh ga Ga…) serves as a common base for all Indian languages. In addition, there are also a few graphical signs (G), which are used for denoting nasal consonants and nasalization of the vowels. Figure 1 illustrates the basic standalone vowels and consonants in the ancient *Brahmi* script. In *Brahmi* script, alphabets are phonetically classified into groups of consonants and vowels. Consonants and vowels combine together to describe a word in the language. The shapes of these vowels and consonants could be different in different Brahmi derived scripts, but their base is still the same. As shown in the Figure 1, there are 9 basic standalone vowels. The consonants can be grouped into 6 groups of 5 consonants each and a last group consisting of 4 consonants.

Gut.	DEV	GUJ	PUN	BEN	ORI	TEL	KAN	TAM	MAL	SINH	URD	SIND
k	क	ક	ਕ	ক	କ	క	ಕ	க	ക	ක	ک	ڪ
kh	ख	ખ	ਖ	খ	ଖ	ఖ	ಖ	–	ഖ	ඛ	کھ	ک
g	ग	ગ	ਗ	গ	ଗ	గ	ಗ	–	ഗ	ග	گ	گ
gh	घ	ઘ	ਘ	ঘ	ଘ	ఘ	ಘ	–	ഘ	ඝ	گھ	گھ
ṅ	ङ	ઙ	ਙ	ঙ	ଙ	ఙ	ಙ	ங	ങ	ඞ	ں	ڱ

figure 2: First five consonants in different Indic scripts.

Figure 2 shows an example of the first five basic consonants used in the *Brahmi* script rendered differently in some other Indic scripts. The columns in the figure represent different Indic scripts while the 5 rows denote the first 5 consonants. A single syllable used in an Indian language consists of either a lone vowel optionally followed by a graphical symbol or a consonantal syllable consisting of a consonant followed by a vowel and optionally followed by one or more graphical symbols. It is also possible to have two consonants in a syllable. This methodology of having consonants and vowels in a syllable is unique to scripts that are derived from the *Brahmi* script. Figure 3 shows the consonant-vowel combinations for the letter Ka in different Indic scripts.

Indic scripts have a unique structure which necessitates the user to type multiple keystrokes to enter one character. We will explain this with respect to *Devnagari*, the script used in Hindi, Marathi, Konkani and Sanskrit languages. Figure 4 shows the Devnagari alphabet set with frequently used consonants and vowels. Devnagari has 53 base letters – 34 consonants and 19 vowels in addition to numbers and punctuation marks. The last nine of the 34 consonants are also known as semi-vowels.

Issues with current input mechanisms for Indic scripts

Theoretically, infinite number of Devnagari characters can be generated by combinations of consonants and vowels. Actually, a smaller subset of characters is currently used in Devnagari. But even this small subset is quite large that makes it impossible to design a practical input mechanism where one key corresponds to one character. It is necessary to have multiple

अ आ इ ई उ ऊ
ए ऐ ओ औ अं अः

क ख ग घ ङ
च छ ज झ ञ
ट ठ ड ढ ण
त थ द ध न
प फ ब भ म
य र ल व श
ष स ह ळ

क्ष त्र ज्ञ श्र

figure 4: The Devnagari *Varnamala* showing the most frequently used 12 vowels, 34 consonants and 4 conjuncts.

keystrokes of the base consonants and vowels to generate most Devnagari characters. However, enough keys are not available on QWERTY keyboards, even to represent the base consonants (34), vowels (19), and the *halant*; here we require 54 keys against the 26 that are available. This problem intensifies when we think of inputting Devnagari on devices like mobile phones where only 9 or 10 keys are available on the keypad to accommodate all these characters [6].

Non-keyboarded devices such as mobile phones, Kiosk booths, Interactive TVs, and WLLs have good mechanisms for inputting English language through a number-pad. However, using the same mechanism to input text in Indian languages is a non-trivial challenge because of the huge matrix of consonant-vowel and consonant-consonant-vowel combinations that are possible in Indian languages.

The most common method for inputting English on a mobile phone is the multitapping ('*abc*', '*def*', ….) input mechanism. Here the user presses the key **2**('*abc*') on the number-pad of the mobile thrice in succession (with as little time gap as possible) to enter the character **'c'**. If a similar mechanism is used for inputting languages based on Devnagari script, then it requires having 5 consonants on each number of the keypad and 12 vowels on the last keypad number. Because of the timeout problem, entering these numbers becomes really cumbersome, in case if a mistake is made. For example if a user wants to enter the standalone vowel ए which comes 7th in the vowel list, then he needs to tap a number-key on the keypad 7 times. In addition, he needs to tap the key fast enough that can avoid the timeout problem. If by mistake, he taps the key 8 times, then he needs to add 12 more taps so that he can come back to the correct character again. Slightly better mechanisms for entering texts in Indian languages such as EziText [3], Inscript [4], PenfoPad [7], and KeyLekh [5] exist today. However, the issues that these methods try to address are slightly different and a few of them are meant for desktop computers.

AKSHAR input mechanism

Challenges

Three issues related to Indian languages pose challenges for the design of the input mechanism for Indic Script:

1. Structural complexities of the Indic scripts
2. Cognitive styles of writing and typing
3. Large number of characters

One of the major problems confronting the development of an input mechanism for mobile phone which is Indic scripts compliant is that of the key-pad. One of the problems is numerical in nature: how to load a large number of characters onto the 10 key key-pad and at the same time ensure that texts can be entered with the least number of key presses as well as avoiding user-irritation. On the other hand, providing a highly unfamiliar key-pad based on character frequencies would mean user resistance, since the user would have to learn a new alphabet order alien to him. Theoretically two simple models exist: a simple solution is to emulate the English keyboard style and load on the keys the different characters. However this method of multitapping does not work for Indic scripts which admit a large number of characters. The user has to press too many keys to enter simple words. The second set of methods includes two-key input method, eZiText [3], PenfoPad [7] and Pronunciation-based method. Though these methods are fast as compared to the simple multitapping mechanism, they fall short to provide a de-facto input mechanism which can take advantage of the inherent structural features of the Indic scripts.

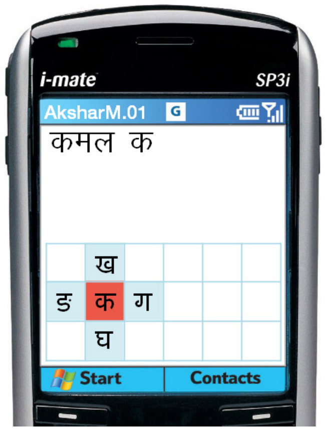
The basic goal of the research was to devise a new input mechanism based on the structural features of the Indic scripts. It was aimed to come up with a generic text input mechanism which can be ported to various digital devices.

Solution details
The basic design concept of the AKSHAR is to make use of the -UP-DOWN-LEFT-RIGHT- arrow keys, available in mobile phones or devices such as TV remotes.

The AKSHAR interface has two main elements:

1. Number keys and UP-DOWN-LEFT-RIGHT arrow keys of keypad
2. Display Screen (text region and 6X3 grid)

Since Indic scripts have an intuitive phonetic grouping system of 5 consonants in a group, we have assigned each of these groups to a single number-key on the keypad. The display screen has two main regions: text region and 6X3 grid. The user is given visual feedback in the 6x3 grid at the bottom part of the screen. The UP-DOWN-LEFT-RIGHT arrow keys or the D-pad (also known as direction pad, joystick, navigation keys) are used to traverse through the grid.

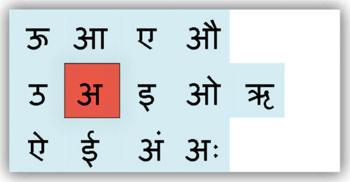


कमल क

	ख				
ङ	क	ग			
	घ				

figure 5: Visual feedback available on the mobile device when number 2 on the keypad is pressed.

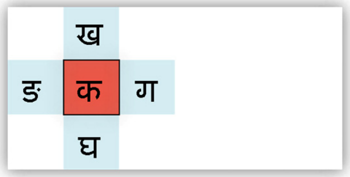
The Figure 5 shows the 6X3 grid and the text region on the screen of a mobile phone.

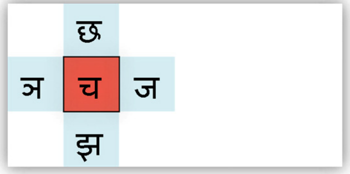


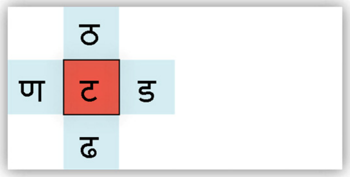
ऊ	आ	ए	औ		
उ	अ	इ	ओ	ऋ	
ऐ	ई	अं	अः		

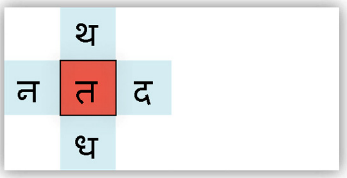
figure 6: 6X3 grid representation on keypress 1.

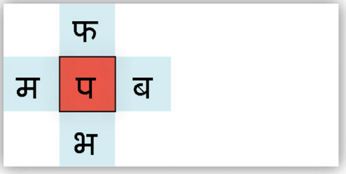
Pressing a number key on the keypad displays the group of characters, which can be entered using that key, in the 6X3 grid region. Figure 6 shows the 6X3 grid representation, which shows the 12 standalone vowels that will be displayed on the screen, on pressing the key **1** on the keypad.

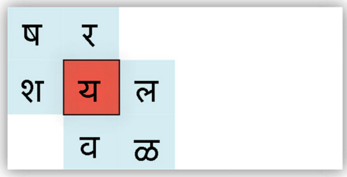












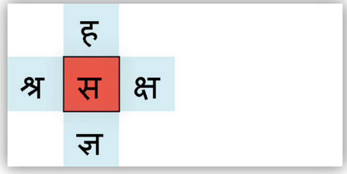


figure 7: 6X3 grid representations on keypress 2 to 8.

Figure 7 shows the 6X3 grid representations that will be displayed on the screen, on pressing the key **2,** key **3,** key **4,** key **5,** key **6,** key **7,** and key **8** on the phone keypad, respectively. Each of these groups is phonetic group of 5 consonants in the Devnagari script, so it is very intuitive for the user to refer them by the first character of the group.

When a particular number on the keypad is entered, the user sees all the possible vowels/consonants that can be entered using that number. For example, when the number **2** on the keypad is entered, the screen provides visual feedback as shown in the Figure 8. By default the first character (क in this case) of the group, which is at the center in the grid, is inputted in the text region. The D-pad can then be used to select the actual character that is desired from the characters displayed in the 6X3 grid. For example if the user wants to enter the letter ख, then he types 2 on the keypad (which

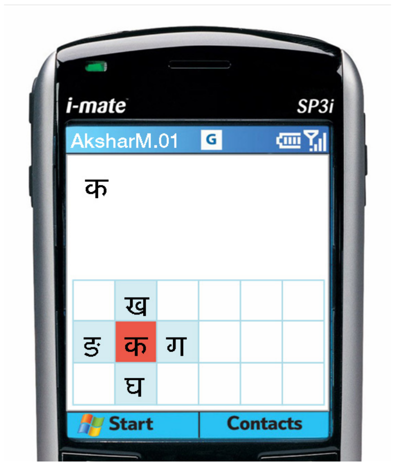


figure 8: Visual feedback available on the mobile device when number 2 on the keypad is pressed.

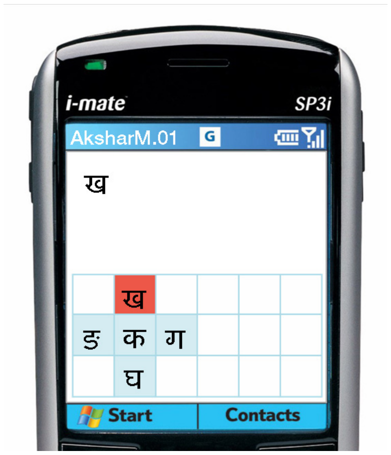
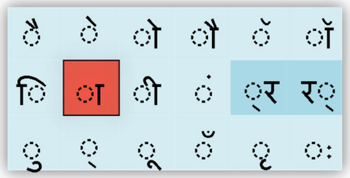


figure 9: Visual feedback available on the mobile device when number 2 on the keypad is pressed, followed by the up arrow key.

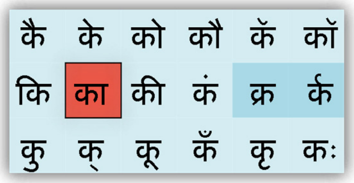
shows क and related consonants of the phonetic group) followed by the up arrow key, which would input ख as shown in the Figure 9.



◌ै	◌े	◌ो	◌ौ	◌ॅ	◌ॉ
ि◌	◌ा	◌ी	◌ं	◌्र	र्◌
◌ु	◌्	◌ू	◌ँ	◌ृ	◌ः

figure 10: 6X3 grid representations on keypress 9 the first time.

The mechanism for entering vowel *matras* is also unique and simple. Figure 10 shows the 6X3 grid that gets displayed to the user when the number 9 on the keypad is entered.



कै	के	को	कौ	कॅ	कॉ
कि	का	की	कं	क्र	र्क
कु	क्	कू	कँ	कृ	कः

figure 11: 6X3 grid representations on keypress 9 the first time when क (Ka) has been entered as the earlier character.

In case of the vowel *matras*, slightly more feedback is provided to the user. The grid shows the rendered forms of the consonant/consonant combination that was entered along with the different possible *matras* applied to that consonant/consonant combination. For example, on pressing the key **9** on the keypad, the user will see the grid as displayed in the Figure 11, if the previously inputted character is क.

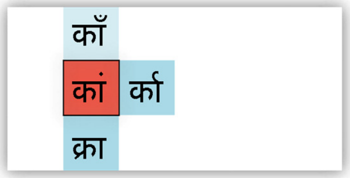


figure 12: 6X3 grid representations on keypress 9 the second time when का (Kaa) has been entered as the earlier character.

Graphical symbols (such as the *chandrabindu* and the *anuswar*) can also be entered using the key **9** on the keypad. Pressing the key **9** on the keypad the first time allows the user to select a vowel *matra* and pressing the key **9** the second time allows the user to enter the graphical symbols. For example, on pressing the key **9** second time on the keypad the user will see grid as displayed in the Figure 12, if the previously typed character is का (entered by pressing the key **2** following the key **9** on the keypad).

Some commonly used signs and symbols such as question mark, comma, semi colon, sentence end and blank space are mapped on the key **0**, key **#**, and key * of the keypad as shown in the Figure 13.

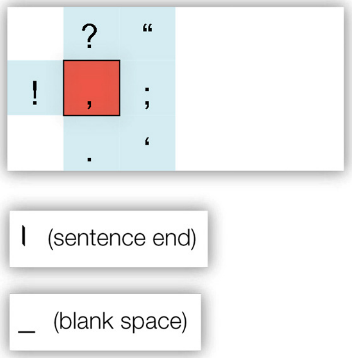


figure 13: Signs and symbols mapped with key 0, # and *.

The major advantages of the AKSHAR mechanism in comparison to the other exiting input mechanisms are:

- **No timeout problems**, since the user uses the arrow keys on the device to select the character (unlike multi-tapping, where the same key has to be tapped multiple times in order to select a particular character).
- **Visual feedback** to the user, by providing the group of characters, which can be entered using a key, in the 6X3 grid region.
- **Less number of clicks,** in order to select a particular vowel or consonant.
- **Clear mental model**, since the alphabet grouping model is the same model that people learn in their schools.
- **Smaller learning curve**, since the visual feedback and intuitive phonetic grouping model makes it simple to use for the novice users.

The other important feature of the AKSHAR implementation is the Unicode [9] compliancy. Having support of the Unicode standards enables it to be a generic mechanism across different platforms and devices. The Unicode also enables the users to type a message or text in multiple languages is another interesting feature, the mechanism provides.

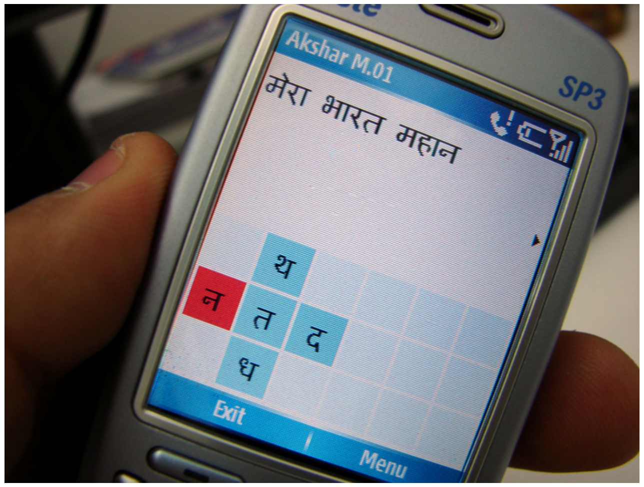


figure 14: Working prototype of the AKSHAR.

We implemented a fully working version of AKSHAR to input Indic scripts on mobile phones. Current version of AKSHAR runs on Microsoft Windows Mobile 5.0 platform and has been implemented on .NETCF. Figure 14 shows a working prototype of AKSHAR on a mobile phone.

User evaluation and feedback

To evaluate the effectiveness of the AKSHAR mechanism, we tested the mechanism with several users of varied profiles. We gave different words and sentences of varied complexities to the users to type in using the AKSHAR.

The main goal of user evaluation was to evaluate the effectiveness of the mechanism in terms of ease of use, speed, learning speed and user's mental model. We also wanted to have qualitative feedbacks about the interaction experiences of the first time users and users who have used the mechanism for a phase of time.

From our user studies we can conclude that:

- Akshar could achieve its primary design objective – to devise an easy to use and easy to learn input mechanism for Indian languages. Some first-time users could type with no instructions. All could type with minimal instructions. Instructions seemed to be easy to remember – users who typed once could spontaneously help others.
- The user evaluations demonstrated that first-timers can achieve acceptable speed of use for small typing tasks. Users also perceived the mechanism to be very intuitive to use on the first glance.
- The task list given for the user testing showed that a motivated user can achieve acceptable speed of use with a few hours of practice on Akshar.
- The most significant success of AKSHAR was the empowerment and liberation experienced by the users because they could actually type in their mother tongue.

User evaluation studies over a time inferred that:

- The system has drastic improvement in the terms of number of key-presses for typing some message in mobile phones, as compared to other current methods in use.

- The system is also powerful than the two key inputting mechanisms, because of visual support of the interface in selecting the character.
- Unlike other methods, the system has a great advantage of having no timeout problem for words with adjacent characters which are on the same key.
- The learning curve for AKSHAR is very small, as the model of grouping of 5 consonants is the same as that of the users' mental model for remembering Indic script alphabets.
- The interaction is very intuitive even for first time users.

In overall, the new mechanism is very efficient as compared to the existing inputs methods in use for inputting Indic scripts.

Conclusion and future work

In this paper, we described a mechanism that can be used to input text in Indian languages, which are derived from the ancient *Brahmi* script. In the world, the percentage of people who speak languages that use scripts derived from *Brahmi* script is also a huge 22 percent. This is also an advantage as the input mechanism we are proposing can be used to input a large number of scripts with some parameterization of the software. In addition to mobile phones, the scope of the AKSHAR mechanism can be extended to other devices and platforms like Kiosk booth, Interactive TV, WLL phones, and also the personal computers.

We are currently conducting a longitudinal study to plot the learning curve of AKSHAR mechanism over a period of time. We plan to compare AKSHAR with other current mechanisms for inputting text on digital devices like KIOSK, WLLs and Interactive TVs. Pilot deployments of AKSHAR have been made in several locations in urban and rural areas. We hope to commercialize AKSHAR soon and get response from the market.

Currently, AKSHAR supports Gujarati, Hindi, Marathi, Punjabi, Telugu, Kannada and Bengali as input languages. We are working on the next prototype version of AKSHAR that supports other Indian languages, too. In the next prototype, we are exploring the possibilities of combining the AKSHAR mechanism with other powerful input mechanisms like T9 [8] in order to make it even more powerful. On other end, we also plan to support word-prediction and dictionary support.

Acknowledgements

We wish to acknowledge the special contributions of Prof Anirudha Joshi, for his inputs in the design of the AKSHAR mechanism. We also thank Abhijeet Rokade, Abhishek Thakkar, Amisha Banker of Industrial Design Center, IIT Bombay for their valuable inputs and support during the project. We are grateful to Mr. Srini Koppplu, Mukund B., Pritkamal Singh and Vijayshree Choudhary of Microsoft India Development Center for their support in the implementation of the working prototype of the AKSHAR.

References

[1] *C-DAC multilingual technologies* [Online]. Available from: http://www.cdac.in/html/mlingual.asp [Accessed 5th March 2006].

[2] *Department of Official Languages* [Online]. Available from: http://rajbhasha.nic.in [Accessed 12th January 2006].

[3] *eZiText* [Online]. Available from: http://www.zicorp.com/eZiText.htm [Accessed 10th January 2006].

[4] *Inscript keyboard layouts* [Online]. Available from: http://www.mit.gov.in/tdil/keyoverlay.htm [Accessed 10th January 2006].

[5] Joshi, A., Ganu, A., Chand , A., Parmar, V. and Mathur, G. Keylekh: a keyboard for text entry in indic scripts, In *CHI '04 extended abstracts on Human factors in computing systems*, ACM Press (2004).

[6] MacKenzie, I. S., and Soukoreff, R. W. Text entry for mobile computing: models and methods, theory and practice. *Human-Computer Interaction*. 17(2002), 147-198.

[7] *PenfoPad* [Online]. Available from: http://www.penfosys.com/penfopad.htm [Accessed 12th January 2006].

[8] *T9 Text Input* [Online]. Available from: http://www.t9.com [Accessed 23rd February 2006].

[9] *Unicode standard for Indic scripts and languages* [Online]. Available from: *http://unicode.org/faq/indic.html* [Accessed 19th January 2006].