An Intelligent Multi-layered Input Scheme for Phonetic Scripts

Shrinath Shanbhag
NCST
Gulmohar Cross Road No. 9
Juhu, Mumbai 400049 India
+91-22-6201606 Extn. 323
shrinath@ncst.ernet.in

Durgesh Rao
NCST
Gulmohar Cross Road No. 9
Juhu, Mumbai 400049 India
+91-22-6201606 Extn. 371
durgesh@ncst.ernet.in

R. K. Joshi
NCST
Gulmohar Cross Road No. 9
Juhu, Mumbai 400049 India
+91-22-6201606 Extn. 314
rkjoshi@ncst.ernet.in

ABSTRACT
We introduce a new scheme for stylus-based input of phonetic scripts such as Indic, using a compact smart soft-keyboard. Phonetically related characters are grouped into layers and become dynamically available when the "group-leader" character is accessed. This scheme allows rapid input using taps and flicks. We have developed a prototype for Devanagari which covers the complete script using just 21 virtual keys, and preliminary tests indicate that it is very easy to use with little or no training.

This scheme seems to be optimal for compact keyboarding of phonetic scripts, such as Indic, on hand-held and mobile devices. It can be extended to other phonetic scripts such as IPA. It can also be used equally well as an alternate, simpler soft keyboard for conventional desktop systems.

Keywords
Soft-keyboard, pen-based computing, text-entry, Indic scripts, self-disclosing characters, hand-held devices, stylus-based input

1. INTRODUCTION
Mobile computing devices are now a part of life. Text being the main medium of communication, these devices demand compact, quick and easy to use text input schemes. Keyboards are not particularly amenable to mobile computing. Many alternate stylus based schemes exist for the Roman alphabet. However, no such known work exists for Indic scripts.

Localisation to Indic scripts is a non-trivial task due to significant differences from Latin based scripts and writing systems. Efforts have been made in the recent past [6, 7] to build standards based architectures for Indic text representation and shaping. These architectures are primarily designed for desktop computing. They can be adapted to work on mobile devices with one exception - the text input mechanism. Here we propose a new compact soft keyboard based on principles of phonetic encoding, that can fill this gap.

Symp. on Smart Graphics, June 11-13, 2002, Hawthorne, NY, USA

1.1 Stylus-based text input methods
In this section we investigate existing stylus based input methods for Latin-based script on mobile devices.

A conventional *soft keyboard* is a graphical representation of a desktop keyboard on the screen, activated by tapping keys with a stylus. In a soft "QWERTY" keyboard all alphabets are visible on the screen. The shift key is merely a mechanism to change case. For an equivalent Indic keyboard, many alphabets are hidden in the shift positions. The hidden alphabets are hard to guess, which makes such keyboards hard for "hunt and peck" as is common for Latin-based scripts. An additional issue is the small screen sizes on most mobile devices. Displaying a 12 column by 5 row keyboard takes up significant screen space. The resulting keys are small and hard to use.

Unistroke[1] is a new alphabet designed to increase accuracy of handwriting recognition techniques. Here each letter is created with a single stroke. The stroke begins when the pen touches the surface of the tablet and ends when the pen is raised. This method requires learning new strokes. The number of strokes or symbols must be reasonably small. If too many symbols are used, recognition rates suffer due to lack of distinctness between them. Examples of single-stroke alphabets are Unistroke, Graffiti[3].

T-Cube [2] is a fast, self-disclosing pen-based alphabet wherein the user selects a character by flicking in one of eight directions from one of nine regions on the screen.

Quickwriting [4] is a technique for continuous stylus-based text entry wherein the user works with a very simple stylized alphabet, in which each character represents one character on the standard typewriter. User enters text by resting the stylus on a central resting zone and then dragging the stylus into various zones, finally returning to the resting zone.

Cirrin[5] is a system for pen input of ASCII characters using word-level unistrokes. Characters are arranged along the circumference of a circle. Users input a word at a time by resting the stylus on the touch screen and dragging it through various characters. Input is accepted when the user lifts up the stylus.

The schemes described above are unlikely to work well for Indic scripts. Conventional soft keyboards, as well as the above tap-and-flick approaches such as T-Cube, Cirrin and Quickwriting, would be unwieldy due to the larger alphabet size. Stroke-recognition based methods such as Graffiti are likely to be even more complicated because of the complex letter shapes and syllabic nature of these scripts. Exploiting the phonetic nature of these scripts, as in our scheme, handles this complexity well.

In the following sections, we examine the phonetic nature of Indic scripts and mechanisms in use for Indian language text input. We then describe our proposed input scheme, its prototype implementation and finally list our conclusions and future plans.

Figure 1 Alphabets of Devanagari Script

2. INDIC SCRIPTS

2.1 Phonetic basis of Indian languages

Most of the 18 major languages spoken in India have their orthography derived from the ancient Brahmi script. The others are Perso-Arabic in origin. Panini's phonetic classification of the Indian alphabets into vowels (V) and consonants (C) serves as a common base for all Indian languages derived from Brahmi scripts. In addition, there are also a few graphical signs used for denoting nasal consonants, nasalization of vowels etc (G). This scheme is phonemic in nature. Figure 1 show the different alphabets of our encoding for the Devanagari script. The effective unit of the writing system for all these Indian languages is the orthographic syllable, consisting either of a lone vowel, optionally, followed by a graphical sign with the structure (V)(G) or a consonantal syllable consisting of a consonant and a vowel (CV) core and, an optionally following sign (G). The canonical structure for a syllable is thus of the form ((C) C) CV (G), as listed in the Unicode Standard (or [C]* CV [G]* in standard regular expression format). Two consonants in a syllable is a common phenomenon. In some syllables, the number of consonants can go even up to five.

2.2 Indian language text input

There are three different approaches to Indian language text input for digital systems. These are Direct Entry, Graphic Entry and Phonetic composition.

Direct entry is a direct adaptation of Hot Metal technology methods that were used in earlier days in printing Indian language texts. The basic idea is to have various type forms, in a font, corresponding to graphically distinct syllables which are then linearly composed to form words.

Graphic entry is similar to the manual typewriter approach. Here a minimal set of graphic primitives is provided and syllables are composed as combinations of these primitives.

Phonetic composition is a scientific approach to the composition of syllables. This method is endorsed by Unicode [12] and is rapidly becoming the input method of choice. Here consonants, vowels and other characters are encoded separately and syllables are dynamically composed at run time. Phonetic input has three variants – full consonant, pure consonant and transliterated Roman. The former two are distinct encoding techniques while the latter is just a convenience mapping.

The variants *full consonant* and *pure consonant* arise because the same vowel may appear in different graphic forms depending on the context. Indian languages are vowel dominant. The consonants in their pure form do not include the vowel sounds. Vowels have to be attached to the consonants to make them complete.

In the *pure consonant* approach, consonants are always in pure form and the vowels (including the 'A') are always explicitly added to the consonant to make it a full consonant.

In the *full consonant* approach, the consonant is assumed to have the vowel 'A' attached to it by default. Grammatically, another vowel cannot be attached to it. One uses the matra forms of other vowels as a separate entity which attaches to the full consonant.

The full consonant approach needs encoding of a minimum of 66 basic primitives whereas the pure consonant approach needs only 50. A complete discussion of the two different encodings is provided in [8].

3. PROPOSED INPUT SCHEME

Our new smart soft keyboard is based on the pure consonant approach and composes the entire repertoire of syllables using minimal primitives. Our encoding is a superset of the pure consonant encoding. It can address the exhaustive syllabic range. This set is in accordance with Unicode code page for Devanagari Script (0x900 – 0x97f). It has three layers – alphabet layer, symbol layer and number layer. It operates in two modes – the alphabet mode and the numeric mode.

The alphabet layer (fig. 2) provides fast access to all alphabets. The top row of this layer comprises of vowel groups. The middle row comprises of five consonant groups corresponding to the middle five rows of the alphabet. The third row comprises of the remaining consonants. This layer also consists of the backspace key, previous and next cursor movement keys and the mode key.

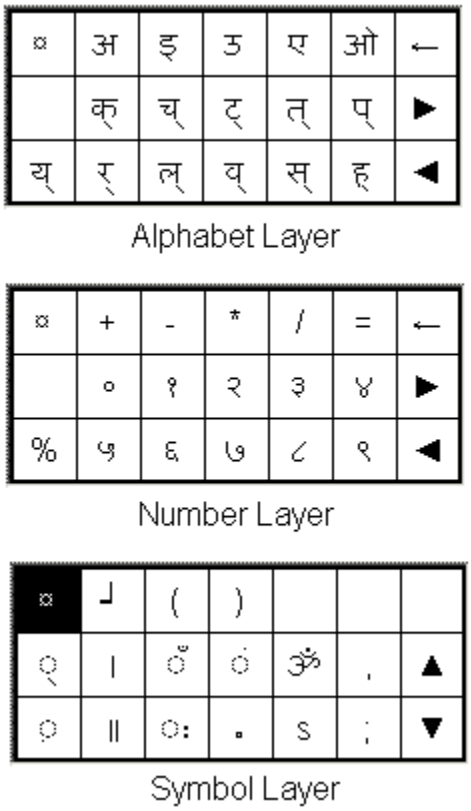

Figure 2 Alphabet, Number and Symbol Layers

The symbol layer consists of commonly used symbols and punctuation marks. The symbol layer is accessible from both modes. The characters in this layer are always a flick away.

The number layer consists of the digits 0-9 and associated symbols like '+', '-' etc.

The mode, backspace, previous and next keys are directly accessible from both modes. The mode key is used to alternate between the alphabet and numeric modes.

3.1 User Input

To enter a character, the user first selects the appropriate mode. The default start-up mode is the alphabet mode. Modes alternate on tapping the mode key. The user rests the stylus on the desired character group leader. All characters accessible from this group are then dynamically disclosed to the user. The layout changes to show only the current group, and remains so till the input gesture is complete. Figure 4. shows the layout when the 'K' key is accessed. The user drags the stylus to any desired character from the group. An input gesture is complete when the user lifts the pen. The system accepts the last character under the pen. In order to cancel the operation, the user moves the stylus to a blank cell. Punctuation marks and symbols are accessed using a similar mechanism with the mode key as the group leader. Figure 5 shows the input sequence for the word "Joshi" (typed as J + O + Sh + II).

3.2 Contributions of our scheme

Our soft keyboard has the following significant advantages over conventional soft keyboards.

1. Our keyboard requires lesser number of keys. Our keyboard is just 7 columns wide and 3 rows tall for a total of 21 virtual keys. It requires 48% lesser width and 33% lesser height than the conventional soft keyboards available on mobile devices such as Window Palm-size PC 1.2, which typically comprise of 12 columns and 5 rows. Alternatively, the keys can be made larger, to allow better legibility and easier operation.
2. Characters are grouped together, based on
    a. their systematic grouping in the alphabet
    b. their phonetic associations with other characters and
    c. their frequency of usage

    This allows users to locate hidden characters intuitively. It eases the learning curve and facilitates faster hunt and peck for beginners.
3. Access to most characters is just a simple tap or flick away. Flicks are particularly fast with a pen. We expect a trained user to be able to operate using taps and flicks.

Figure 3 Special Keys

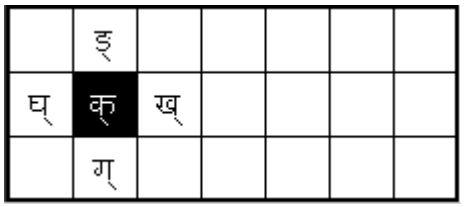

Figure 4 K Group Characters

3.3 Prototype Implementation

We have implemented our current scheme for the Windows desktop and Palm-size PC 1.2 platforms. Our system is fully compatible with Unicode character encoding. The system currently simulates pure consonant encoding for the Devanagari script using the existing Unicode full consonant encoding (code page 0x900 – 0x97f).

3.4 Evaluation

We have not yet done systematic evaluation. Initial user feedback, however, has been very encouraging. Users familiar with the script have been able to input text with just a simple briefing. We feel that our scheme will require very little training. A few hours of practice should suffice to attain proficiency in this scheme.

4. CONCLUSION AND FUTURE WORK

We have proposed a new input scheme that exploits the systematic grouping of phonetic scripts. It is smart, easy to use, compact and yet exhaustive. This scheme seems to be optimal for stylus-based input of Indic scripts, particularly on mobile and hand-held devices such as the Simputer. This also seems to be a promising approach for the International Phonetic Alphabet (IPA). Our preliminary analysis indicates that the scheme may also be useful as an alternate, simpler virtual keyboard for conventional desktop systems. We are working towards a more formal evaluation of the scheme, extensions to other Indic scripts and deployment and testing on actual hand-held devices.

5. REFERENCES

[1] Goldberg D., Richardson, C., Touch-Typing With a Stylus, Proceedings of the INTERCHI'93 Conference on Human Factors in Computing Systems, 1993, ACM, pp. 80-87

[2] Venolia, D., Neiberg, F., T-Cube:A Fast, Self-Describing Pen-Based Alphabet. Proceedings of CHI 94, 1994, ACM, pp. 265-270

[3] Mackenzie, I. S., Zhang, S. The Immediate Usability of Graffiti, Proceedings of Graphics Interface '97, 1997, Canadian Information Processing Society, pp. 129-137

[4] Perlin, K. Quikwriting : Continuous Stylus-based Text Entry,Proceedings of UIST'98,1998,ACM

[5] Mankoff, J., Abowd, G. D. Cirrin: A word-level unistroke keyboard for pen input, Proceedings of UIST'98, 1998, ACM, pp. 213-214

[6] Mudur, S.P., Nayak, N., Shanbhag, S., Joshi, R. K. An architecture for the shaping of Indic texts, Computers & Graphics, 23(1), 1999, Pergamon, pp. 7-24

[7] Shroff K., Mudur S.P., Gaur R., Vinod K., Ketkar Y. Indic Script processing under X11, STRANS 2001, IIT Kanpur

[8] Mudur, S.P., Pattanaik, S.N., Nath, S. J., Computer Processing of Indian Scripts – A pure consonant approach. Proceedings of the National Seminar on Computer Aided Language Processing, Delhi, 1987.

[9] Department of Official Languages, Prabodh Primer, Ministry of Home Affairs, Government of India

[10] IS 13194, Indian Script Code for Information Interchange ISCII. Bureau of Indian Standards, December 1991

[11] The Simputer, http://www.simputer.org

[12] The Unicode Standard Version 3.0. The Unicode Consortium. Addison-Wesley Developers Press, Reading, MA, 2000.

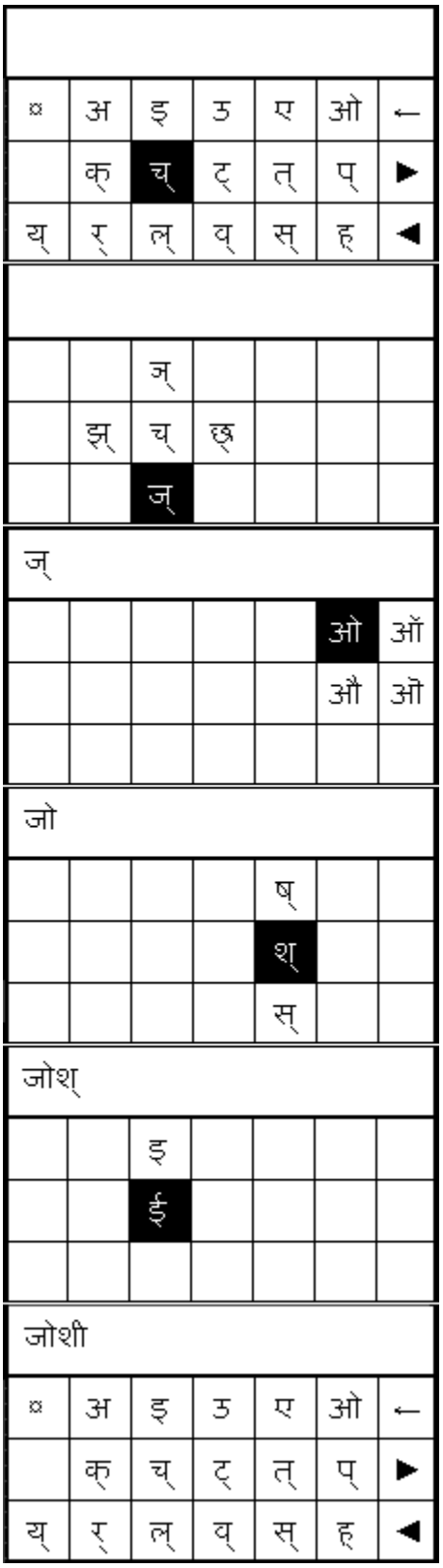

Figure 5 Input sequence for the word "Joshi"