Designing an Efficient Virtual Keyboard for Text Composition in Bengali

Soumalya Ghosh
IIT Kharagpur
Kharagpur - 721302
West Bengal
sghosh@sit.iitkgp.ernet.in

Sayan Sarcar
IIT Kharagpur
Kharagpur - 721302
West Bengal
sayans@sit.iitkgp.ernet.in

Debasis Samanta
IIT Kharagpur
Kharagpur - 721302
West Bengal
dsamanta@sit.iitkgp.ernet.in

ABSTRACT
Recent advancement in communication and information technology (ICT) flourishes the computing and handheld devices in urban and rural areas in India. Now people all over the world communicate each other using hand-held digital gadgets like PDA, cell phone etc. in addition to desktop PC and laptop. But, text entry task remains critical in English as well as Indian languages. Moreover, non-availability of standard mechanism in Indian languages makes obstruction in better text composition. Text entry through standard hardware keyboard is not viable in many digital gadgets because of size, mobility restriction etc. As an alternative to hardware keyboard, virtual keyboard based text entry has been advocated with easy to personalize, low cost and user friendliness. This paper proposes an approach to design a virtual keyboard for text entry in Bengali language. The proposed approach can be extended to other Indian languages with a minor modification.

Categories and Subject Descriptors
H.1.2 [**Information Systems**]: User/Machine Systems – *Human factors, Human information processing.*

H.5.2 [**Information Interfaces and Presentation (e.g., HCI)**]: User Interfaces – *Graphical user interfaces, User-centered design, Input devices and strategies (e.g., mouse, touch screen)*

General Terms
Design, Human Factors.

Keywords
Virtual keyboard, text entry in Indian languages, user interface design, design optimization, human computer interaction.

1. INTRODUCTION
The virtual keyboard also called soft keyboard is one of the most primitive alternative text entry mechanisms for entering text. It is basically replaces the hardware keyboard through on screen character key image map. User has to select the keys from the interface for typing text. Special helping and rate enhancement strategies like word prediction, word completion, scanning [2], adaptation and personalization [7] can be incorporated with the virtual keyboard, which are difficult to incorporate with hardware keyboard.

The efficiency of a virtual keyboard is measured by its text entry rate. In an efficient virtual keyboard, keys should be arranged in such a way that it leads to maximize typing speed and minimize motor movement. So far the key arrangement of virtual keyboard is concerned, English language keyboards are based on some common design strategies like character frequency, maintaining a minimum distance between each character pair etc. The standard rules of English virtual keyboard design not necessarily applicable in Indian languages as the structure of the languages are completely different from English. Inflexions (“matra”) and complex characters (“yuktakshar”) in Indian languages make the key arrangement a tedious task. Moreover, number of characters in Indian languages is more than double than English. Optimal arrangement of so many characters into a small sized virtual keyboard layout is, in fact, a hard problem.

Bengali, the national language of Bangladesh and mostly spoken in some states (West Bengal, Tripura, Assam etc.) in India, is world’s sixth most spoken language [16]. It consists of 61 characters including independent and dependant vowels (matra), consonants and space character. In this paper, we propose to design an optimal Bengali virtual keyboard so far text entry rate is concerned. To reduce the design complexity with the large character set, we arrange the character set into multiple zones. We then find an optimal arrangement of characters in each zone to reduce motor movement time and hence text entry rate gets increased.

The rest of the paper is organized as follows. In Section 2, we discuss the existing approaches for designing virtual keyboard. Proposed approach of designing Bengali virtual keyboard is presented in Section 3. User testing task is discussed in Section 4. Finally, Section 5 concludes the paper.

2. RELATED WORK
Several works have been reported to design virtual keyboards. In this section, we discuss about the existing virtual keyboards, their design principles and compare their performances.

Most popular keyboard layout is QWERTY keyboard [17]. It is designed based on “ten finger” typing which is used in hardware keyboard. For this reason, QWERTY is not at all effective in designing single pointer typing virtual keyboard. The Fitaly [4] keyboard is a commercial product of Textware Solution. The design is based on placement of more frequent keys into center and dual size space keys at the two end of home row to minimize hand movement during text entry with one finger, a stylus or a

IndiaHCI2011, April 7–10, 2011, Bangalore, Karnataka, India.

pen. MacKenzie and Zang [12] introduced a new keyboard layout, named Opti. They first put ten most frequent characters in the center of the keyboard. Then rest of the most diagraph frequent top ten characters are assigned to the top ten keys and followed by replacement of remaining characters. All arrangement is done by trial and error method. They further improve it in 5×6 layout. Mankoff and Abowd [13] proposed a new system, called Cirrin, for pen input of ASCII characters using word-level unistroke. They put all characters present in English alphabet set into the perimeter of a circular shape in order to minimize the average distance between any pair of letters. Lewis et al. [11] proposed a 5×6 virtual keyboard layout called Lewis keyboard with a strictly alphabetical sequence, which is suffered in the alphabetical discontinuity caused by row breaks problem. S. Zhai et al. [18] design virtual keyboard using Metropolis algorithm of statistical physics. Metropolis algorithm is adopted on the basis of Fitts-Digraph energy [18] derived from Fitts-Digraph model [15] to design Metropolis keyboard. Jan Eggers et al. [3] propose a mechanism to solve keyboard optimization problem through Ant Colony algorithm. They take ergonomic criteria into evaluation function of designing a keyboard, which are not applicable in single pointer virtual keyboard. GregoryW. Lesher et al. [10] proposed an approach to make optimized single finger keyboard using Greedy algorithm. Huge computation will be required to reach an optimal state by following this Greedy approach. M. Raynal and N. Vigouroux [14] propose a mechanism to solve the keyboard optimization problem using the concept of genetic algorithm. In this approach, initialization of the first population is done by placing the key randomly. The individuals can live, reproduce and die following the fitness function. The process is executed iteratively until an optimal layout has come up. They calculate "Improvement potential" for each character in every individual and sort the characters accordingly. This sorting results many individual with same key arrangements. So, it should reduce the search space. Avro keyboard [1] of OmicronLab, used the arrangement of Bengali alphabet set. The consonant set is divided into two sub parts. The first left sub part is arranged according to the Bengali alphabet set, and right subpart also can be filled in by using the same procedure. This idea is not helpful to decrease the motor movement at the time of typing.

iLEAP keyboard [9] from C-DAC is used Bengali Qwerty image map and that is developed for ten finger typing not for single pointer. In fact, work on designing of virtual keyboard for Indian languages is scarcely reported in the literature till date.

3. PROPOSED APPROACH

Our approach to design a virtual keyboard for text entry in Bengali is discussed in this section. We examine "Wikipedia" Bengali language corpus, compute the frequency of occurrences of alphabets and perform language modeling task to find unigram and bigram probabilities of occurrences. Our analysis of the corpus reveals that "space" character is most frequently used. We assign space character at the centre key position of the keyboard's main character panel to minimize distance to access it from all the other keys.

We then place the characters into three different panels (character, inflexion, punctuation) in the layout as shown in Fig.1.

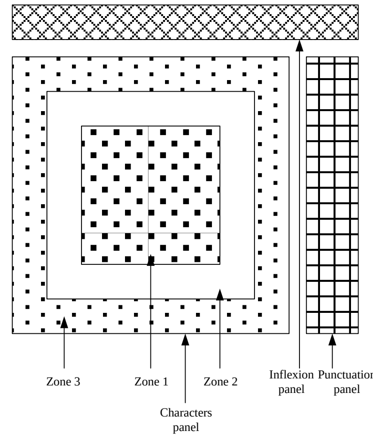

Figure 1. Character panel architecture of proposed keyboard

For the spatial arrangement of character keys in the character panel, we propose to divide the panel into three zones (Zone 1, Zone 2 and Zone 3) as shown in Fig. 1. We assign the characters belong to higher frequency values (like: র, ক, ম, স, ত etc) in central zone (Zone 1). Medium frequent characters (like: শ, জ, দ, য়, ভ etc) are placed in Zone 2 surrounding the central zone. The rest of the characters (like: ঈ, ঐ, ণ, ঢ etc) are resided at Zone 3 which is outside of the Zone 2.

Our aim is to arrange the keys spatially in the proposed character panel of the layout so that their placement is optimum with respect to the number of hand movements. We use genetic algorithm to solve the optimization problem. Here, we employ the algorithm for each zone separately and final possible solutions for each zone are merged together for obtaining the optimal solution. The first population is generated by positioning all the keys randomly. In order to compare the different individual of a population, we model their performance (fitness function) according to *mean motor movement time* (MT_{MEAN}) of Soukoureff and MacKenzie's model [15] shown in Eqn.1. This is calculated by summing the Fitts' law [5] movement time between all bigrams weighted by the frequency of occurrence of the bigrams. Individual of a generation is selected according to their fitness values.

$$MT_{MEAN} = \sum_{i=0}^{n}\sum_{j=0}^{n}\left(MT_{ij} \times P_{ij}\right) + RT \qquad (1)$$

where, n is the number of keys in the keyboard, movement time from i^{th} key to j^{th} key is MT_{ij}, probability of typing i^{th} key after j^{th} is represented by P_{ij} and reaction time (RT) is calculated through Hick-Hyman law [6], [8] as shown in Eqn.2, 3, 4 respectively.

$$MT_{ij} = a + b\log_2\left(D_{i,j}/W_j + 1\right) \qquad (2)$$

Hence, a and b are two empirically determine constants, $D_{i,j}$ is the Euclidean distance between i^{th} key to j^{th} key and W_j is width of the target key (j^{th} key).

$$P_{ij} = f_{i,j}\Big/\sum_{i=1}^{n}\sum_{j=1}^{n} f_{i,j} \qquad (3)$$

Where, $f_{i,j}$ is digraph frequency of i^{th} key to j^{th} key.

$$RT = a + b + \log_2 n \qquad (4)$$

Hence, a, b are two empirically determined constant and number of keys present in keyboard is n. In case of expert user the value of reaction time is zero.

We consider randomly chosen crossover point for a pair of individuals in our GA framework. To provide more randomness, we mutate the individuals by randomly selecting two pairs of keys and swapping between them. All the above mentioned steps have been executed iteratively with proper testing for convergence. We further decide convergence criteria for stopping the iteration.

In our approach, we apply genetic algorithm for Zone 1 characters and a solution is obtained. Then apply the same to Zone 2 characters along with Zone 1 solution and finally genetic algorithm is applied to Zone 3 with the solutions of Zone 1 and 2. The Bengali virtual keyboard with our proposed approach is shown in Fig 2. We name our proposed Bengali virtual keyboard as *Multi zonal GA keyboard* (MZG keyboard).

4. EXPERIMENTAL RESULTS

4.1 Designs for testing

To experiment the effectiveness of our proposed MZG keyboard, we design two different types of keyboard on the basis of taking different chromosome set namely *All vowel consonant GA keyboard* (VCG keyboard), *All consonant GA Keyboard* (CG keyboard). Also A popular Bengali keyboard named "*Avro*" [1] is being used for the experiment. In VCG keyboard all consonant, vowel, space and full stop key are considered only in single zone. In CG keyboard only consonants are taken into one zone. All vowels are placed at the top of this zone and they are in alphabetical order. We consider these two design to test the effectiveness of our proposed multiple zonal arrangement of keys.

4.2 Design evaluation

We propose to evaluate the designed keyboards by users through form filling method. To evaluate the effectiveness of the design, we have to decide, which types of user are the end users for the system and measure how they have performed with the interface.

Participant users: In every user-centric design, the users play a significant role in evaluating the product which follows the design. For user selection task, we have selected users from different domains like local markets, schools, and domestic helps etc. Experiment has been done with three types of users (according to level of Bengali literacy) of different occupation and 22 people in average belonging to each user group. User category U_1 consists of 13 users who are extremely well familiar with Bengali language such as office executives and school teachers. People belong to U_2 user category have good knowledge in Bengali.

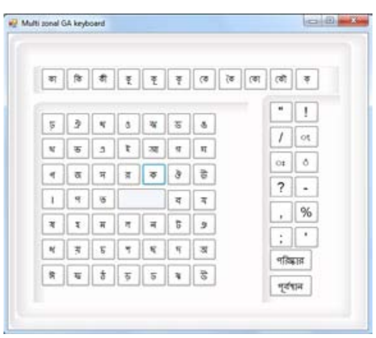

Figure 2. Multi zonal GA keyboard (MZG keyboard)

We select 28 users in this group including businessmen and office staff. U_3 user class consists of 25 users who have knowledge in Bengali language in its writing form but not well education background like shopkeepers, domestic help person, rickshaw pullers.

Text composition: In our evaluation procedure, users have to fill the form with appropriate data on the basis of their knowledge without any supplementary help to fill it. Here, we design three different types of forms namely *General* ($Form_G$), *Social* ($Form_S$) and *Selected text* ($Form_T$) typing form. General form contains text fields like – name, father name, mother name, city/village name, state, high school, college, occupation. Ethnicity, religion, about me, sports, authors, movies, and cuisines fields are incorporated in Social form. In selected *text* typing form, a portion of text selected from popular Bengali novel "Kapalkundala" by Bankim Chandra Chattopadhyay is augmented with the form and user has to type the specified text in field of form by employing our virtual keyboards. More detailed statistics about users typed text (average text length, number of complex, number of inflexions) are shown in Table 1.

Evaluation: We have conducted experiments with the above mentioned users to evaluate the keyboards. Before the actual experiment, and text composition tasks we give enough introduction about those virtual keyboards and conduct a short training session for increasing familiarity with those keyboards and forms. The interface is developed with Microsoft C#.NET 3.5. The host systems are Intel Core 2 Duo running Microsoft Windows. The experiment is conducted in the HCI Lab, School of Information Technology, IIT Kharagpur. The time taken to build entire experiment is about four weeks. The experiments are conducted with our three developed Bengali virtual keyboards layouts. 20 sessions are being conducted. Users are requested to fill those forms by typing Bengali text using those virtual keyboards. In every different session with respect to one user, the testing form and system for evaluation are changed. The average text entry rate (wpm) has been calculated from the total time to fill the forms by the program in background. The results are also being documented in a log file automatically. User performances in those Bengali virtual keyboards to fill the form are listed in Table 2.

Table 1. Text typing

Form Under Test	Avg. no. of Characters	Avg. no. of Inflexions	Avg. no. of Complex Characters
$Form_G$	95	37	11
$Form_S$	138	52	19
$Form_T$	150	61	22

Table 2. Text entry performance evaluation

Form Under Test	Text Entry Rate (Word per Minute)			
	MZG keyboard	VCG keyboard	CG keyboard	Avro keyboard
$Form_G$	9.73	8.93	8.17	7.14
$Form_S$	9.61	8.85	7.83	7.06
$Form_{ST}$	8.26	7.84	6.72	5.68

5. CONCLUSIONS AND FUTURE WORK

There are many virtual keyboards available but they are mainly for text entry in English. Virtual keyboard for text composition with Indian languages in general and Bengali in particular is yet to be designed. Existing virtual keyboard design principles, however, are not properly applicable to design an efficient virtual keyboard in Bengali. This work addresses this limitation and proposes to design an efficient virtual keyboard suitable for text entry in Bengali. Our proposed keyboard is also comparable to existing '*Avro*' virtual keyboard which is designed in an ad-hoc manner. By analyzing the learning curves in Fig. 3, we can easily comprehend that performance of our keyboard shows better result and more user friendly than '*Avro*' keyboard. Figure 3 shows that after a number of sessions, the users get more acquainted with the keyboard and performs better. We conclude that the proposed MZG keyboard is a better candidate for composing text in Bengali.

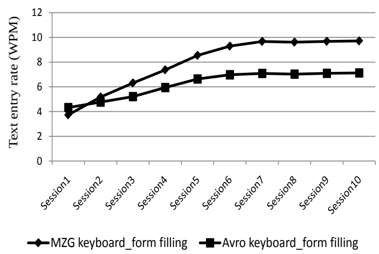

Figure 3. Learning Curve for Bengali virtual keyboard

There is more scope for improving our work. In this work, we have used the traditional approaches to evaluate a design based on measurement of text entry rate, which is done by measuring movement time only. In fact, design related user's error should be judged in order to measure the efficacy of a design. This is particularly significant for a Bengali virtual keyboard, where it contains a large alphabets set with linguistic features such as matra, complex characters, phonetic and graphical similar characters etc. The lack of consideration of user error behaviour confined the practical usefulness of the resulting optimal virtual keyboard. We would study user's error behaviour on the virtual keyboard layout as our next work. From the study, a predictive error model would be proposed. We will consider the user error prediction model with movement time model to evaluate a design of a virtual keyboard. In fact, we design an optimum Bengali virtual keyboard to maximizing text entry rate as well as errors while typing.

6. REFERENCES

[1] Avro Keyboard. Features of avro keyboard. Website. http://www.omicronlab.com/

[2] Bhattacharya, S., Samanta, D., and Basu, A. 2008. User errors on scanning keyboards: Empirical study, model and design principles. *Interacting with Computers (IwC), Elsevier press*. 20(3):406–418, 2008.

[3] Eggers, J., Feillet, D., Kehl, S., Wagner, M.O., and Yannou, B. 2003. Optimization of the keyboard arrangement problem using an ant colony algorithm. *European Journal of Operational Research*. 148, 3:672–686, 1 August 2003.

[4] Fitaly. The Fitaly One-Finger Keyboard. Website. http://www.fitaly.com/fitaly/fitaly.htm

[5] Fitts, P. M. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*. 47:381–391, 1954.

[6] Hick, W. E. 1952. On the rate of gain of information. *Quarterly Journal of Experimental Psychology*. 4:11–26, 1952.

[7] Himberg, J., Häkkilä, J., Kangas, P., and Mäntyjärvi, J 2003. On-line personalization of a touch screen based keyboard. *In Proceedings of the 8th international conference on Intelligent user interfaces*, pages 77–84. New York, NY, USA:ACM

[8] Hyman, R. 1953. Stimulus information as a determinant of reaction time. *Journal of Experimental Psychology*. 45:188–196, 1953.

[9] ileap. Website. http://tdil.mit.gov.in/download/ileap.htm

[10] Lesher, G.W. and Moulton, B.J. 2000. A method for optimizing single-finger keyboards. *In Proceedings of the RESNA Annual Conference*. Pages 91–93, 2000.

[11] Lewis, J. R., LaLomia, M. J and Kennedy, P. J. 1999. Evaluation of typing key layouts for stylus input. *In Proceedings of the Human Factors and Ergonomics Society 43rd Annual Meeting*. 1999.

[12] MacKenzie, I.S. and Zhang, S.X. 1999. The design and evaluation of a high performance soft keyboard. *In Proceedings of the CHI 99 Conference on Human Factors in Computing Systems*. 1999.

[13] Mankoff, J. and Abowd, G.D. 1998. Cirrin: a word-level unistroke keyboard for pen input. *In Proceedings of the 11th annual ACM symposium on User interface software and technology - UIST*. Pages 213–214. New York, NY, USA: ACM, 1998.

[14] Raynal, M. and Vigouroux, N. 2005. Genetic algorithm to generate optimized soft keyboard. *In CHI 2005*. Portland, Oregon, USA. ACM, 2005.

[15] Soukoreff, R. W. and MacKenzie, I. S. 1995. Theoretical upper and lower bounds on typing speed using a stylus and soft keyboard. Behaviour and Information Technology, 14:370–379, 1995.

[16] Wikipedia. List of languages by number of native speakers Website. http://en.wikipedia.org/wiki/List_of_ languages_by _number_of_native_speakers.

[17] Wikipedia. Qwerty. Website. http://en.wikipedia.org/wiki/ QWERTY.

[18] Zhai, S., Hunter, M., and Smith, B. A. 2000. The metropolis keyboard - an exploration of quantitative techniques for virtual keyboard design. *In Proceedings of the ACM Symposium on User Interface Software and Technology - UIST*, pages 119–128. New York: ACM, 2000.