

Towards the Optimization of Gaze-Controlled Human-Computer Interaction: Application to Hindi Virtual Keyboard for Stroke Patients

Yogesh Kumar Meena*, *Member, IEEE*, Hubert Cecotti, *Senior Member, IEEE*,
KongFatt Wong-Lin, *Member, IEEE*, Ashish Dutta, *Member, IEEE*, and Girijesh Prasad, *Senior Member, IEEE*

Abstract—Virtual keyboard applications and alternative communication devices provide new means of communication to assist disabled people. To date, virtual keyboard optimization schemes based on script-specific information along with multimodal input access facility are limited. In this work, we propose a novel method for optimizing the position of the displayed items for gaze-controlled tree-based menu selection systems by considering a combination of letter frequency and command selection time. The optimized graphical user interface (GUI) layout has been designed for a Hindi language virtual keyboard based on a menu wherein 10 commands provide access to type 88 different characters along with additional text editing commands. The system can be controlled in two different modes: eye-tracking alone and eye-tracking with an access soft-switch. Five different keyboard layouts have been presented and evaluated with ten healthy participants. Further, the two best performing keyboard layouts have been evaluated with eye-tracking alone on ten stroke patients. The overall performance analysis demonstrated significantly superior typing performance, high usability (87% SUS score), and low workload (NASA TLX with 17 scores) for the letter frequency and time-based organization with script specific arrangement design. This work represents the first optimized gaze-controlled Hindi virtual keyboard, which can be extended to other languages.

I. INTRODUCTION

OVER 20 million people suffer from stroke annually and due to loss of motor and/or speech skills, they face a lot of difficulties in communicating effectively [1]. Augmentative and alternative communication (AAC) based methods and technologies have been designed to provide new effective means of communication to assist these people [2]. In general, AAC systems include a large set of gestural, alphabetical, and iconic communications [3]. This area of research is growing rapidly to meet the needs of severe speech and motor impaired individuals, such as aphasia and tetraplegia patients [4]. An AAC system can be divided into two distinct components: input device(s) and communication software. The input devices are used to capture any type of voluntary intent of patients (e.g. electrodes for electromyography (EMG) and/or electrooculography (EOG) signals). One of the most popular input devices is the eye-tracker that captures eye gaze of the user in real-time [5]. The eye-gaze based AAC systems have

been developed previously for patients suffering from locked-in syndrome [6], and amyotrophic lateral sclerosis [7]. In addition to eye-gaze, brain-computer interfaces (BCIs) have been implemented for communication by using non-invasive electroencephalography (EEG) recordings [8], [9]. Moreover, dedicated communication software, e.g. virtual keyboards [10], [11], and complex communication spreadsheets [12], have been proposed to generate explicit information regarding the users' intent by analyzing the data captured by input devices.

A conventional virtual keyboard includes a specific language-based keyboard layout [13]. This type of keyboard is one of the most primitive mechanisms for alternatively entering text [14]. The design of an efficient virtual keyboard is mainly driven by its corresponding language script [15]. Currently, the majority of the available virtual keyboards are targeted to the English language and likewise, the optimization schemes are developed in accordance with this language (for review see [4], [16]). However, changes in the nature and number of the alphabets, which may accompany with a change in language, may render the techniques developed for the English language ineffective or even inapplicable for other languages. In particular, there is a lack of virtual keyboard applications specific to the Hindi language, which is the official language of India (490 million speakers). Further, the disability ratio is very high in developing countries, such as India, where more than 30 million people suffer from speech and motor disabilities (reported by Census, 2001 [17]). Thus, there is a pressing need for the development of efficient Hindi language based virtual keyboards.

The Hindi language contains more than 60 basic letters which are significantly larger than the number of letters in the English language (i.e. 26). These letters can be further divided into three distinct categories of vowels (11), consonants (33), and matras (17) (i.e. diacritics) including halant (i.e. killer stroke) along with 3 other complex characters (i.e. a combination of two consonants) [18], [19]. The matras are a unique set of the Hindi alphabets commonly used with the consonants, also called modifier consonants. Most of these matras are represented as dependent forms of the vowels. There are three other consonants apart from 33 regular consonants, known as conjuncts or irregular consonants (a combination of two consonants). This happens when successive consonants with no vowel between them physically join together, for example, क्ष is a combination of क and ष. There are two other special matras used to modify the consonants. The first is the nukta,

Y. K. Meena, H. Cecotti, K. Wong-Lin, and G. Prasad are with the Intelligent Systems Research Centre, School of Computing & Intelligent Systems, Ulster University, Derry~Londonderry, N. Ireland, UK. A.D is with Indian Institute of Technology, Kanpur, India, e-mail: (meena-y@email.ulster.ac.uk).

Manuscript received February XX, 201X; revised August XX, 201X.

represented by ः. For instance, क़ can be written as क + ः. The second is halant, it represented by ्. Thus, the complexity of the Hindi language structure is a primary constraint for efficient typing using a regular QWERTY keyboard and a user may need significant training before composing the text [20], [21].

The QWERTY keyboard has been utilized previously in hand-operated virtual keyboards for English language (i.e. Dvorak, FITALY, OPTI, Cirrin, Lewis, Hookes, Chubon, Metropolis, and ATOMIK keyboards) and for Hindi/Bengali language (i.e. Google, iLeap, Keylekh, MyLanguage, Microsoft, Avro, Lipik, Lookeys, Guruji, and gate2home keyboards) with highest typing speed of 42.16 and 5.45 wpm (words per minutes), respectively (for review see [16]). An optimal layout has been presented for the Hindi language with typing speed of 6.14 wpm [16]. Also, the QWERTY keyboard has been optimized to design virtual keyboard applications in Meitei language with an average typing speed of 7 wpm [22]. The significant decrease in performance of virtual keyboards with Hindi/Bengali/ Meitei language compared to the English language clearly show the language specific dependency in these applications. Thus, a similar relationship may also associate with eye-tracker/soft-switch controlled virtual keyboards.

A recent study found significantly faster text entry rates for Dasher keyboard when compared to a dwell-keyboard [23], [24]. Dasher resulted in approximately twice as fast an entry rate as Eye-Typing. The entry rate in the first session was 6.6 wpm for Dasher and 4.6 wpm for Eye-Typing. Entry rates increased with practice. In the last session, the entry rate had increased to 12.4 wpm for Dasher and 6.0 wpm for Eye-Typing. Another study has presented multimodal virtual keyboard system using eye-tracking alone and eye-tracking in combination with soft-switch for the English language and reported typing speed of 9.3 and 15.26 lpm (letters/characters per minutes), respectively [25]. A head mounted gaze controlled text entry interface for the Hindi language has been developed with an average typing speed of 9.63 wpm [26], however, the usability of the system was questionable due to the incompatibility of the hardware for disabled people. Moreover, the system was evaluated using word prediction approach and short sample size (only 6 healthy participants) without considering the typing errors. The word prediction approach can be used to improve the typing speed in wpm. However, studies have shown that the typing rate is affected by word completion and word prediction methods [27], [28].

Eye-tracker based virtual keyboard interfaces face major challenges for item selection wherein gaze and dwell time durations are used for pointing and selecting items simultaneously. On the one hand, the typing speed can be increased by choosing a short dwell time duration, although it may lead to false item selections due to involuntary eye movements. On the other hand, if the dwell time is too long, it may increase the user's discomfort. This issue is known as the Midas Touch problem in detection theory [29]. Moreover, the majority of the prior works are mainly focused on developing novel designs of graphical user interface (GUI). However, their optimization has been largely ignored when it is used with various methods of controlling an AAC device such as

eye-gaze and soft-switches. Furthermore, no previous work is known to have developed Hindi virtual keyboard with multimodal input access facility.

In terms of optimizing the GUI in the English language, frequency based arrangement of letter has been previously utilized for virtual keyboard design. This system was evaluated using a soft-switch device with four different scan periods (2100 ms, 1800 ms, 1500 ms, and 1200 ms). However, the performance was not reported in terms of the typing speed. Only task completion time for both alphabetical layout and frequency based arrangement layout were reported [30]. This particular study showed that the time to complete a task can be reduced by frequency based arrangement layout. Further, this optimised virtual keyboard layout was evaluated with eye-tracking and soft-switch input modalities. The average typing speed using eye-tracking with dwell time (i.e. 1.5 s) and eye-tracking with soft-switch were reported 7.25 lpm and 12 lpm, respectively [31]. However, such approaches have not been utilized for the Hindi language. In our previous study, we have proposed a Hindi virtual keyboard based on a basic alphabetical layout with multimodal facilities. The average typing speed using eye-tracking with dwell time (i.e. 1.5 s) and eye-tracking with soft-switch were reported 10.62 lpm and 13.50 lpm, respectively [32]. Thus, previous studies of gaze based virtual keyboard application has not been fully focused and developed for complex structure languages (i.e. Hindi language). Also, optimization based on estimation of layout parameters was not considered.

The proposed system in this work is designed to overcome the shortcomings of our previous study [32] by improving the GUI using a novel technique. In relation to the previous work, the following elements correspond to the novel aspects of this work. At the method level, a novel approach to take into account both the probability of occurrence of the items to be selected and the performance of the eye-tracking detection is proposed; this method is fully independent of the chosen language script. At the application level, the halant (i.e., ्) based approach is considered with this application wherein a single letter can be used for consonant clusters when typesetting conjunct ligatures is not feasible. For instance, अघ्य can be written as अ + घ + ् + य. Similarly, क्ष can be written as क + ् + ष. A similar process can be applied to three character words (e.g., character 1 + halant + character 2 + halant + character 3). Another crucial character, known as nukta (ँ) is included in the application. For instance, क़ can be written as क + ः. The most prolific diacritic has been the subscript dot (nukta). Hindi uses it for the Persian, Arabic and English sounds, and for the allophonic developments. Therefore, while designing a virtual keyboard application for the Hindi language, these nukta and halant based approaches must be considered. Nine additional required punctuation marks and the 10 digits (i.e., numerals) are also added with this virtual keyboard application. Therefore it could be used to type digits and punctuation marks for any Hindi language sentence. Furthermore, other necessary functionalities such as delete all, new line, and go-back commands for corrections are included. The go-back function can be used to shift the GUI back in case of a wrong selection. The last five typed characters are

also displayed in the GUI at the bottom of each command box that improve the system performance by reducing the need of excessive eye movements. A typical virtual keyboard consists of two sections: the key display section and the text display section; and the characters can only be placed in the key display section. Thus, it is a challenging task to place all the characters in a clear and convenient manner and to type with minimal search time. In order to design a robust and efficient GUI, this challenge can be overcome by implementing a two-step process. First, we investigate how a convenient and efficient GUI can include a large set of symbols on its visual display unit. Second, we develop a strategy that can locate individual symbols to a particular location of the GUI. These issues can be resolved collectively by designing a multi-level virtual keyboard [33] and the characters can be placed on the layout based on their probability of occurrence and the constraints of the input device [30]. However, the probability of characters in the large corpus is not readily available for the Hindi language and it must be determined. Therefore, this work provides the estimation of the letter frequencies for the Hindi language script using a large corpus of characters. The variety of themes included in the database corpus was considered to avoid biases related to the frequency of appearance of the Hindi language characters.

The aim of the work is therefore to design an optimized gaze-control virtual keyboard that takes into account both the gaze characteristics and the used language script. The major contributions of this work are: 1) the estimation of the letter frequencies for the Hindi language script using a large corpus of characters; 2) a method to design an efficient layout for gaze controlled virtual keyboard based on frequency of the letters and the selection time duration of each letter, and 3) an experimental evaluation of the performance of different possible layouts of the proposed Hindi virtual keyboard (HVK) with ten healthy participants and ten stroke patients.

This paper is organized as follows: Section II describes the system. Section III proposes a tree-based menu optimization method. Section IV discusses the parameter estimation approach for GUI optimization by the creation of the letter frequencies of Hindi letters and average time to produce the commands. Section V presents the proposed virtual keyboard GUIs. Section VI discusses our empirical study with performance evaluation methods. The results are then presented in Section VII and discussed in Section VIII. Finally, Section IX concludes the paper.

II. SYSTEM OVERVIEW

A. Description

The main two components of the multimodal system are two different input devices (i.e. eye-tracker and single input switch) and a text entry application (see Fig 1). Further, the developed GUI comprised of two display components: a multi-level menu that comprised of ten commands at each level and an output text display (i.e. message box) where the user can see the typed text in real-time. The positions of the ten commands (i.e. c1 to c10) are depicted in Fig 2(a). The tree-based structure of the ten commands is presented in Fig 2(b). The resolution

of each rectangular command box is approximately 14% of the optimum screen resolution. All the command boxes are placed on the periphery of the screen while the output text box is placed at the center of the screen (see Fig 2(a)).

The tree-based structure of the GUI provides the ability to type 45 Hindi language characters, 17 different matras and halants, 14 punctuation marks, and the 10 digits. Moreover, other functionalities such as delete, delete all, new line, space and go-back commands for corrections are also included. The selection of a particular character requires the user to follow a two-step task. In the first step, the user has to select a particular command box where the desired character is located. The successful selection of a command box shifts the GUI to the second level, where the ten commands on the screen are assigned to the ten characters which belong to the selected command box in the previous level. In the second step, the user can see the desired character and finally select it for typing in the text box. After the selection of a particular character at the second level, the GUI goes back to the first level automatically. The placement and size of the command boxes are identical at both levels.

With a virtual keyboard using eye-tracking, it is necessary to provide an efficient feedback to the user that the intended command box/character has been selected in order to avoid mistakes and increase efficiency. A visual feedback is provided to the user as a change in the color of the border section of the observed command box. If the user gazes to a particular box for a duration of time t , the color of the border changes linearly in relation to the dwell time (Δt). An audio feedback is also provided to the user through an acoustic beep after successful execution of each command. This beep sound makes them proactive so that they can prepare easily for the next character. The color-based visual and acoustic beep based auditory feedbacks allow the user to continuously adjust and adapt his/her gaze to the intended region on the screen. Moreover, to improve the system performance by reducing the need of excessive eye movements, the last five typed characters are also displayed in the GUI at the bottom of the each command box [25] (Fig 1). This helps the user to see the previously written characters without shifting their gaze to the output display box. Also, the size of the command boxes and the distance between them were optimized for increasing the robustness of the system to involuntary head and body movements.

B. Command selection with eye-tracking

An item was selected based on the directed gaze of the user to the corresponding command box for Δt . The selection of a particular box is achieved by selecting the closest box using the Euclidean distance between the center of the box and the gaze coordinates. Thus, if the gaze coordinates remain in the same region (i.e. nearest to the target item) for Δt duration, this particular item is selected. When the gaze coordinates change from one region to another in less than Δt ms, the timer for the selection is reset to zero. The addition of the soft-switch with eye-tracking has helped to overcome the Midas touch problem, as the user can search the target item with

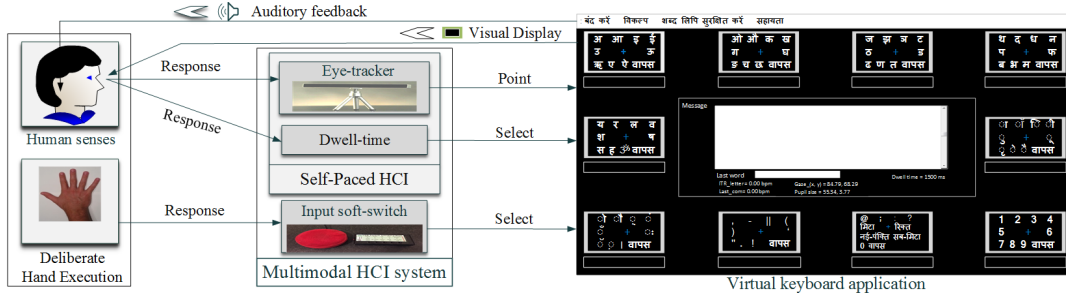


Fig. 1. The main components of the proposed multimodal human computer interaction (HCI) system wherein the eye-tracker only and eye-tracker with a single input switch are used for searching and selection of the items on virtual keyboard application.

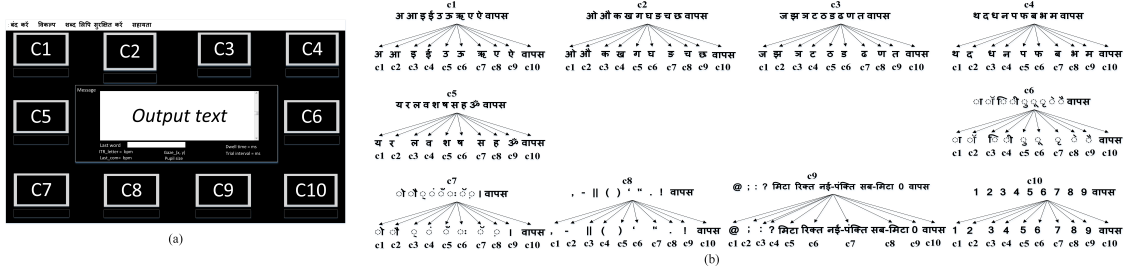


Fig. 2. Tree structure based letter selection of the Hindi virtual keyboard application. (a) The position of ten commands, (b) The tree structure depicting the command tags used for letter selection.

the eye-tracker, and the selection can be done directly via the soft-switch. In the experiments with healthy participants, the soft-switch was pressed by the users' dominant hand.

III. TREE-BASED MENU OPTIMIZATION

For a tree-based menu structure with L levels and M_{com} available commands at each level, the maximum number of possible symbols is $M_{symb} = (M_{com})^L$. Thus, the sequence of commands for each symbol can be defined as $S_m = (cs_1, \dots, cs_l, \dots, cs_L)$ where $m \in \{1 \dots M_{symb}\}$, $l \in \{1 \dots L\}$ and cs_l is the command number at the l^{th} level. For instance, with $L = 2$ and $M_{com} = 4$, a sequence of commands for a particular symbol, e.g. S_1 , can be (1,3) wherein the user has to select command 1 at the first level and then command 3 at the second level. Let us assume $S_C(1, \dots, N)$ and $S_D(1, \dots, N)$ to be the sequence of commands and the sequence of their time durations, respectively, for an experiment involving N produced commands. The $S_D(n)$ represents the duration that was needed to select the command $S_C(n)$ immediately following the selection of the command $S_C(n-1)$. To estimate the average time to produce a command i at n based on the command j at $n-1$, we considered conditional command durations, which are defined as:

$$\bar{A}(i, j) = \frac{A_D(i, j)}{A_N(i, j)} \quad (1)$$

where $A_D(i, j)$ is the sum of the duration times to produce a command i at n based on the command j at $n-1$ for all the N commands, and $A_N(i, j)$ is the number of times when it occurs. They can be computed as follows:

$$A_D(i, j) = \sum_{n=1}^N \begin{cases} S_D(n), & \text{if } S_C(n) = i \text{ and } S_C(n-1) = j \\ 0, & \text{otherwise} \end{cases}$$

$$A_N(i, j) = \sum_{n=1}^N \begin{cases} 1, & \text{if } S_C(n) = i \text{ and } S_C(n-1) = j \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

Finally, the estimated duration corresponding to the selection of a symbol m is given as:

$$D(S_m) = \frac{1}{M_{com}} \sum_{k=1}^{M_{com}} \bar{A}(S_m(1), k) + \sum_{l=2}^L \bar{A}(S_m(l), S_m(l-1)) \quad (3)$$

where the first part is the average duration to select a command while the next terms are for the conditional command durations. If we consider the prior probability (P_m) to select a symbol m , then the average time to select a symbol is defined as:

$$E = \frac{1}{M_{symb}} \sum_{m=1}^{M_{symb}} D(S_m) \cdot P_m \quad (4)$$

The goal of GUI optimization is to minimize the value of E . This can be achieved by measuring the values of $D(S_m)$ using Eq. 3 and the probabilities of occurrence of letters P_m . The values of $D(S_m)$ can be coupled in inverse order to the values of P_m . In brief, the highest probabilities of a letter are assigned to the positions of conditional events $\bar{A}(i, j)$ that belong to the lowest values of $D(S_m)$. We assumed there is no error during the command selection. In the next section, this optimization approach is utilized to place characters/letters on the GUI, based on their selection time.

TABLE I
THE HINDI ALPHABET SYSTEM, THE TRANSLITERATION OF ALPHABETS IN ENGLISH LANGUAGE PRESENTED AFTER THE \rightarrow SIGN, AND RELATIVE FREQUENCY (IN %) OF CHARACTERS IN TEXT CORPUS.

Vowels Frequency	अ \rightarrow a 0.53	आ \rightarrow A/aa 0.62	इ \rightarrow i 0.27	ई \rightarrow I/ii 0.32	उ \rightarrow u 0.72	ऊ \rightarrow U/uu 0.04	ऋ \rightarrow ri 0.01	ए \rightarrow e 0.55	ऐ \rightarrow ai 0.05	ओ \rightarrow o 0.10	औ \rightarrow au 0.38
Dependent matras Frequency		ा \rightarrow A/aa 6.81	ि \rightarrow i 2.15	ी \rightarrow I/ii 3.80	ू \rightarrow u 1.33	ु \rightarrow U/uu 0.61	ृ \rightarrow ri 0.05	े \rightarrow e 5.11	ै \rightarrow ai 1.00	ो \rightarrow o 2.22	ौ \rightarrow au 0.17
Ext. matras & sign Frequency	ं 2.44	ँ 0.49	ः 0.00	ः 0.01	॰ 0.49	ॱ 2.07	ँ 0.01	ॡ 0.00	@ 0.00	 0.00	
Consonants Frequency	क \rightarrow ka 4.84	ख \rightarrow kha 0.69	ग \rightarrow ga 1.28	घ \rightarrow gha 0.16	ङ \rightarrow "na 0.00	च \rightarrow ca 0.83	छ \rightarrow cha 0.33	ज \rightarrow ja 1.27	झ \rightarrow jha 0.19	ञ \rightarrow "na 0.01	ट \rightarrow .ta 0.57
Consonants Frequency	ठ \rightarrow .tha 0.26	ड \rightarrow .da 0.57	ढ \rightarrow .dha 0.09	ण \rightarrow .na 0.12	त \rightarrow ta 2.52	थ \rightarrow tha 0.79	द \rightarrow da 1.52	ध \rightarrow tha 0.30	न \rightarrow na 3.24	प \rightarrow pa 1.70	फ \rightarrow pha 0.30
Consonants Frequency	ब \rightarrow ba 1.58	भ \rightarrow bha 0.58	म \rightarrow ma 2.44	य \rightarrow ya 1.43	र \rightarrow ra 4.55	ल \rightarrow la 2.16	व \rightarrow va 1.06	श \rightarrow "sa 0.44	ष \rightarrow .sa 0.15	स \rightarrow sa 2.95	ह \rightarrow ha 3.60
Punctuations Frequency	 1.39	, 0.79	" 0.60	- 0.55	. 0.31	? 0.16	! 0.11	' 0.02	; 0.02	: 0.01	() 0.01
Numbers & Space Frequency	0 0.02	1 0.01	2 0.01	3 0.01	4 0.01	5 0.01	6 0.00	7 0.00	8 0.00	9 0.00	Space 20.83

IV. PARAMETER ESTIMATION FOR GUI OPTIMIZATION

A. Estimation of relative frequencies of letters in Hindi language

The performance of a virtual keyboard can be affected by the letter frequencies in the considered script, thus we initiated our study with the estimation of the occurrence of all the Hindi language characters along with other characters required for sentence completion. The text corpus included 27 national and international news articles published in a vernacular daily newspaper in Hindi language (i.e. Rajasthan Patrika) [34], 23 randomly chosen Hindi poems, 50 randomly chosen Hindi classic, contemporary, mythological, short/long stories, and folk tales from an online florilegium (i.e. Bharat Darshan Hindi literary magazine) [35]. The variety (100 in total) of themes for the database corpus was considered to avoid biases due to the frequency of characters. The total size of the corpus was 975707 characters including non-alphanumeric characters (i.e. line breaks, spaces and punctuation marks). While a previous study reported letter frequencies for the Hindi language, the data has been estimated with only 41,114 characters including vowels and consonants only [36]. Likewise, a similar study reported the letter frequencies for the English language with approximately 800,000 characters (i.e. Cornell University Math Explorer's Project) [37]. Thus, the number of characters included in our study is relatively higher than these previous studies. Also, our methodology for estimation of letter frequencies is similar to these studies. We denote by P_i the probability of occurrence of the i^{th} letter. The relative letter frequencies (in %) in the Hindi script and additional symbols (i.e. 87 letters) are shown in Table I. Apart from the probability of characters, the performance of a gaze-based virtual keyboard can also be affected by visual search and learning time.

B. Experiment 1

An initial study was conducted to estimate the duration corresponding to the selection of a symbol $D(S_m)$ as explained in Section III. An alphabetic organization with script specific arrangement (AOSSA) layout was used in this experiment

as explained in Section II. However, for initial estimation of average time for each command AOSSA layout was selected. It does not involve frequency of letters and/or command selection time optimization. Thus, the baseline layout has been utilized for estimation of these parameters. Our approach is in line with a previous study involving unigram/bigram-based adaptive virtual keyboard layout for the Bengali language [38]. The tree structure depicting the command tags used for letter selection is shown in Fig 2(b).

1) *Participants*: Twelve healthy volunteers (2 females) in the age range of 21-32 years (27.05 ± 2.96) participated in this experiment. No participant had prior experience of using an eye-tracker with the application. Participants were informed about the experimental procedure and purpose prior to the experiment. The Helsinki Declaration of 2000 was followed while conducting the experiment.

2) *Data acquisition*: A portable eye-tracker (The Eye Tribe Aps, Denmark), was used for pursuing the eye gaze of the participants [39]. The eye-tracker data was recorded at 30 Hz sampling rate. Participants were seated in a comfortable chair in front of the computer screen. The distance of the participant from a computer screen (PHILIPS, 23 inches, 60 Hz refresh rate, optimum resolution: 1920 * 1080, 300 cd/m², touch-screen) was about 800 mm. The vertical and horizontal visual angles were measured at approximately 21 and 36 degrees, respectively.

3) *Experimental paradigm*: Each participant was asked to type a predefined sentence, given as 'कबतक जबतक अभ्यास करते रहो. 44-4455-771' The transliteration of the task sentence in English is *Kabtak Jabtak Abhyaasa Karate Raho. 44 - 4455 - 771* and the direct translation in English is *Till When Until Keep Practicing. 44 - 4455 - 771*. This predefined sentence consists of 29 characters from the Hindi language and 9 numbers. The complete task involved 76 commands in one repetition if performed without committing any error. This predefined sentence was formed with a particular combination of characters in order to obtain a relatively equal distribution of the commands for each of the ten items in the GUI. Prior to the experiment, the average command frequency of 7.60 ± 0.84 was measured over the ten command

boxes (items) to type a predefined sentence. Thus, the adopted arrangement provides an unbiased involvement of the different command boxes. The eye-tracker was used for both pointing and the selection of items (see Fig 1). During the experiment, participants directed their gaze at the target item for 1.5 s (dwell time (Δt)) [32]. Prior to each experiment, a calibration session lasting about 20 s, using a 9-point calibration scheme was conducted. No pre-training session was performed.

4) *Average time for each command*: From experiment 1, we recorded the time taken during the selection of each command for all participants. As a total of ten commands ($M_{com} = 10$) were considered at each level (here, $L = 2$) the maximum number of possible symbols that can be located in the GUI design was 100 ($M_{symp} = 100$). Thus, we estimated all the sequences of commands (S_m) for each symbol as described in section III. Furthermore, we measured the average time to produce a command i at time n based on the command j at time $n - 1$ and created a $M_{com} \times M_{com}$ matrix by using Eq. 1. The grand average time to select a command in layer one and two was estimated and assigned to the command positions, which were not selected during the experiments. The average command frequency of 8.33 ± 1.68 was achieved over the ten command boxes (items) to type a predefined sentence. In particular, the c10 command box achieved relatively higher average command frequency (i.e. 12.66) as this command box was used for go-back to the previous menu when the participant made an accidental mistake. Therefore, the command distribution (irrespective the character distribution) over the ten items in GUI remains uniform. Finally, the estimated duration corresponding to the selection of a symbol was calculated using Eq. 3. The average time to select each command is presented in Fig 3. As the goal of the GUI optimization was to minimize E according to Eq. 4, the average time to select a symbol was combined with the letter frequencies given in Table I. This combination was performed in different ways in order to compare various possible GUI layouts for gaze-controlled HVK in next section.

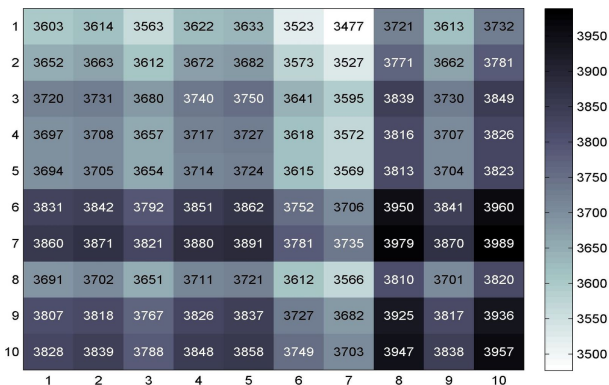


Fig. 3. $M_{com} \times M_{com}$ matrix each cell represents average time (ms) to produce the commands i (row) and j (column) knowing that j is before i .

V. PROPOSED VIRTUAL KEYBOARD GUIs

Five different layouts were designed and evaluated in order to find out an optimal GUI by considering efficiency-usability

trade-off. The objective of these layouts was to observe the change in performance in terms of text entry rate to validate their performance. All these layouts are based on a two-layer selection method with ten commands at each level and the selection of each character requires execution of two commands. The positions of the command boxes are depicted in Fig 2(a). The tree-structure of the five menus for selecting the different characters and symbols is given in Fig 2(b) and Fig 4(a-b). These layouts were developed based on the five distinct strategies that can arrange the different items into a particular subset using the proposed optimization approach and script specific arrangement of the characters. In all the five GUI layouts, the 10th position of the second level was assigned to a go-back (i.e. वापस) function i.e. to shift the GUI back to the first level in case of a wrong selection. As each layout consists of 100 symbols ($M_{symp} = 100$), the total number of letters (M_{letter}) can be less than or equal to 100 i.e. $M_{letter} \leq M_{symp}$. Here, we considered 88 symbols for assigning Hindi characters, matras, halants, basic punctuation, and the space command (i.e. $M_{letter} = 88$). The remaining 12 symbols were assigned to delete, clear-all, and 10 go-back commands.

A. Alphabetic organization with script specific arrangement (AOSSA)

An alphabetically arranged layout is typically easier to learn and remember. Consequently, its usability improves. However, it is a challenging task for these layouts to accommodate the different symbols based on the alphabetical order of the symbols. Fig 2(b) depicts the command tags used for letter selection. The vowels and consonants are placed in an alphabetical order to the first five command boxes (i.e. c1–c5), respectively and the Om symbol is placed at c5 command box. The command boxes c6–c7 comprise all the matras, halant and nukta. The command boxes c8–c9 include the most frequently used punctuation marks, digit 0, space, and other functionalities such as delete, new line, and delete all to correct errors. The last command box c10 is used to access the digits 1 to 9.

B. Frequency and time-based organization (FTO)

The letters frequency-based approach is used to improve the text entry rate of virtual keyboard application. However, utilization of frequency-based approach with the time taken to select a command from a particular location is yet to be implemented for eye-gaze based virtual keyboard application. This study is the first that uses a combination of letters frequency and time-based approach for eye-tracking based virtual keyboard. This layout mainly considers two factors; the relative letter frequencies in the Hindi language script (see Table I) and the average time to produce a command (i.e. a sequence of commands for each symbol) for each location of the GUI (see Fig 3). The new line command is considered with zero frequency. Based on the prior probability of the symbols and the average time to produce a command, a matrix is created using Eq 4. The symbols are assigned accordingly on the location of GUI from command c1–c9. The tree-based

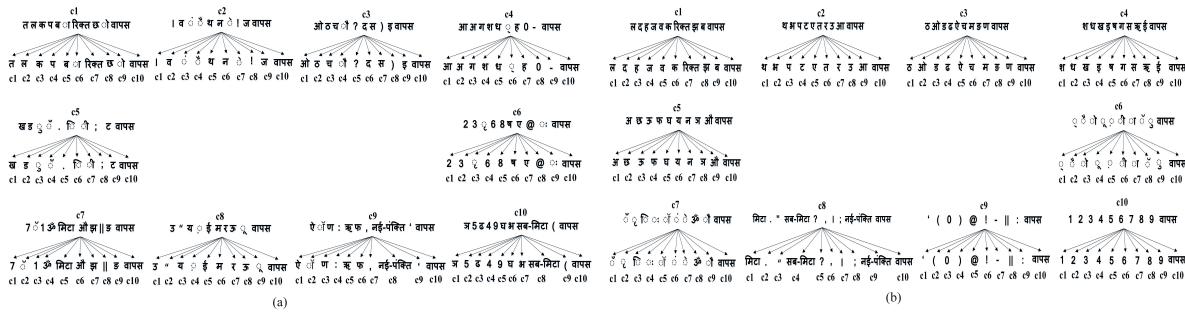


Fig. 4. Tree structure based letter selection of the HVK depicting command tags used for letter selection in FTO layout (a) and in FTOSSA layout (b).

TABLE II
HEALTHY (H) AND PATIENT (P) GROUPS DEMOGRAPHIC AND CLINICAL CHARACTERISTICS.

Variables	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10
Age (year)	29	26	24	23	21	23	30	26	28	22	35	48	59	24	55	38	58	72	62	51
Gender	M	M	M	M	M	F	M	M	M	M	M	M	M	M	M	M	M	M	F	F
Dominant side	R	R	R	R	R	R	R	L	R	M	R	R	R	R	R	R	L	R	F	R
Vision correction	Yes	Yes	Yes	No	Yes	No	Yes	No	Yes	No	No	No	Yes	No	Yes	No	Yes	No	Yes	Yes
Impaired side	-	-	-	-	-	-	-	-	-	-	R	L	L	L	R	R	L	R	L	R
Time since stroke (months)	-	-	-	-	-	-	-	-	-	-	3	8	17	8	8	4	1	1	6	48
Loss of motor skills	-	-	-	-	-	-	-	-	-	-	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Loss of speech skills	-	-	-	-	-	-	-	-	-	-	Yes	No	No	No	No	Yes	No	No	No	No
Grip strength (Kg)	-	-	-	-	-	-	-	-	-	-	3.9	9.2	7.2	11	10	4	4.6	6.7	4.8	5.1
ARAT score	-	-	-	-	-	-	-	-	-	-	7	31	26	40	36	10	14	30	16	22
HMMS score	-	-	-	-	-	-	-	-	-	-	-	10	10	10	10	10	10	10	10	10
Cognitive impairment	-	-	-	-	-	-	-	-	-	-	-	No	No	No	No	No	No	No	No	No

structure is shown in Fig 4(a) depicting the command tags used for letter selection.

C. Frequency and time-based organization with script specific arrangement (FTOSSA)

The Hindi alphabet system is composed of several types of characters which can be grouped according to their usage in the language script. Thus, the inclusion of these characters on a single GUI interface using only FTO based optimization may not provide the best performance. In order to resolve this issue, a script specific arrangement of all the symbols was considered for designing the layout, including the constraints from the FTO layout. The tree-based structure of GUI is shown in Fig 4(b) depicting the command tags used for letter selection. As the GUI contains 100 symbols, the complete set of symbols was divided into five groups. First, the go-back function was incorporated for each command box at the c10 position. Second, we formed a group of 45 characters including vowels, consonants, and space button and place these characters to command boxes c1–c5 based on letter frequencies and the time taken to select a command from a particular location. Similarly based on letter frequencies and their selection time third, fourth, and fifth groups were created. We formed the third group that includes all the matras and om symbol, and placed these characters into the command boxes c6–c7. Fourth, the punctuation marks, number 0 and extra command for the error correction are assigned to the command boxes c8–c9. Fifth, the numbers are placed in the command box c10.

D. Random order of frequency and time-based organization (RaOFTO)

In this layout, all the characters and numbers, except go-back, are randomly placed in each of the ten commands.

E. Reverse order of frequency and time-based organization with script specific arrangement (ReOFTOSSA)

This layout comprised of a reverse arrangement of the FTOSSA. The main emphasis of designing this layout is to measure the virtual keyboard performance in the worst case scenario.

VI. EMPIRICAL STUDY

In order to find the most effective keyboard layout, the five proposed layouts were evaluated through an empirical study based on the performance indices mentioned in the previous section.

A. Participants

A total of twenty adult volunteers participated in this study (see Table II). These participants were divided into two groups (i.e. Group A and Group B). The group A was formed with ten healthy volunteers (one female) in the age range of 21-30 years (25.02±3.08), namely H1 to H10. The group B was formed with ten stroke patients (2 females) in the age range of 24-72 years (50.2±14.32), namely P1 to P10 having variable time durations since the stroke. Their cognitive impairment was tested by Hodgkinson mini-mental test score (HMMS) [40]. The Grip strength was measured using dynamometer (CamryTM Electronic Hand Dynamometer) which is a standard

technique of assessing the muscle strength of an individual after stroke. Action Research Arm Test (ARAT) measurement was also done for each stroke patients, it is a reliable way of testing the upper-limb functionality by checking the grasp, grip, pinch and gross-movement activities [41]. The apparatus used for the test are blocks of wood of different sizes, sharpening stone, cricket ball, glass and jar of water, hollow tubes of different height and thickness, washers, ball bearings, and marbles of different dimensions. The total score associated with the test is 57 which is distributed among different tasks with different apparatus listed above, where each task is scored between 0 and 3. All tests were administered under the supervision of an occupational therapist stationed at the hospital where the clinical trials were conducted. No participant had prior experience of using an eye-tracker and soft-switch with any application. Other experimental parameters of participants were considered same as mentioned in Section IV (B).

B. Data acquisition

The same eye-tracker device was used in this study as mentioned in Section IV (B). A soft-switch (The QuizWorks Company, USA) [31] was used in a hybrid mode with eye-tracker device to search and select a command. Participants were seated in a comfortable chair in front of the laptop screen (DELL, 15.6 inches, 60 Hz refresh rate, optimum resolution 1920×1080 , touch-screen). The distance between the participant and the laptop screen was about 800 mm. The vertical and horizontal visual angles were considered same as mentioned in Section IV (B).

C. Experimental paradigm

The typing task involved a predefined sentence with 104 characters that includes Hindi alphabets, numbers, punctuation mark, and space, written as 'खबर 10 अगस्त 2016, खाना खाने के बाद अगर आप फल खाते हैं तो सही तरीके से यह इंटेस्टाइन तक नहीं पहुंच पाता।' The transliteration of the task sentence in English is *khabar 10 august 2016, khaana khaane ke baad agar aap phal khaate hain to sahee tareeke se yah intestain tak nahin pahunch paata*]. The direct translation in English is *news august 10, 2016, if you eat fruit after dinner it does not reach the intestine in a right way*. The complete task involved the execution of 208 commands in one repetition if performed without committing any error. This predefined sentence was taken from an Indian newspaper (i.e. Amar Ujala) and considered based on its practical flow, meaning, and coverage of all ten commands boxes in the GUI. Thus, the adopted typing task provides an unbiased involvement of command boxes and eye-gaze distribution over the virtual keyboard GUI.

Two input modalities were used to type the text that creates two different conditions of experimental design, shown in Fig 1. The eye-tracker was used for both pointing and the selection of items. During the experiment, participants directed their gaze at the target item for a specific period of time. Second, the eye-tracker along with the soft-switch was used in a hybrid mode wherein the user directed the eye-gaze to the target item, and the selection occurs via a soft-switch. This

multimodal facility was incorporated to overcome the Midas Touch problem of HCI systems. Prior to each experiment, participants were advised to avoid moving their body and head positions during the tests as far as possible. One session was performed for each condition.

The estimated duration of the experiment was about 3-4 hours, making the task difficult and tedious for the participants. Therefore, we divided the experimental conditions and the participants into separate experiments. Experiments 2 and 3 were performed by Group A and Group B, respectively. Furthermore, the order of the GUIs was randomized across participants.

1) *Experiment 2*: In this experiment, the performances of five layouts of the virtual keyboard were evaluated with two conditions. First, the eye-tracker was used by the participants with 1.5 s dwell time. Second, the eye-tracker was used with soft-switch in a hybrid mode. During the tests, the participants used the soft switch once they received the visual feedback (i.e. the color of the gazed item begins to change). The soft-switch condition was incorporated to find out the performance with GUI without dwell time.

2) *Experiment 3*: Based on the results of experiment 2, the two best performing layouts were selected for this experiment with stroke patients. The performances of these two layouts were evaluated using eye-tracker with a 1.5 s dwell time.

D. Performance Evaluation

Several performance indices such as the number of letters spelled-out per minute, the information transfer rate (ITR) at the letter level ITR_{letter} and command level ITR_{com} [25], [32], and the mean and standard deviation (mean \pm SD) of the time to produce each command were used to evaluate the performance of all the proposed GUI layouts. The ITR is calculated based on the total number of actions (i.e. basic commands and letters) and the total time duration that is required to perform these commands. To estimate the ITR all the different commands and letters were assumed as equally probable and there is no typing error. The ITR is given as follows: $ITR_{com} = \log_2(M_{com}) \cdot N_{com}/T$ and $ITR_{letter} = \log_2(M_{letter}) \cdot N_{letter}/T$, where N_{com} is the total number of commands produced by the user to type N_{letter} characters. T is total time to produce N_{com} or type all N_{letter} .

VII. RESULTS

For computing statistical significance, the Wilcoxon signed-rank test was applied using false discovery rate (FDR) correction method for multiple comparisons on various performance measurements across different GUI layouts in experiment 2 and 3.

A. Experiment 1

This experiment was conducted to estimate the average time for each command (see Section IV (B)). The outcomes from experiment 1 and the estimation of relative frequencies of letters of Hindi language were utilized to design the proposed virtual keyboard GUIs in Section V.

TABLE III

TYPING PERFORMANCE (MEAN AND STANDARD DEVIATION (STD.)) ACROSS HEALTHY PARTICIPANTS) FOR EYE-TRACKER WITH SOFT-SWITCH AND EYE-TRACKER ALONE WITH ALL DESIGNS IN EXPERIMENT 2.

Layout		Eye-tracker with soft-switch				Eye-tracker alone			
		Speed (letter/min)	ITR_{com} (bits/min)	ITR_{letter} (bits/min)	Average time (ms)	Speed (letter/min)	ITR_{com} (bits/min)	ITR_{letter} (bits/min)	Average time (ms)
FTOSSA	Mean	18.99	124.81	115.87	1546±1331	11.39	80.06	73.17	2486±1078
	Std.	5.06	37.40	30.25	420±582	0.84	4.97	5.54	158±204
AOSSA	Mean	17.08	126.04	107.61	1618±1537	10.01	79.65	66.32	2482±1124
	Std.	4.16	29.42	26.95	373±628	1.49	2.76	9.42	81±230
FTO	Mean	12.47	92.62	79.18	2339±3354	9.72	71.36	62.21	2786±1828
	Std.	4.27	33.20	27.85	732±1259	1.72	5.71	10.87	228±570
RaFTO	Mean	10.25	76.18	65.47	2758±3945	8.30	63.81	53.22	3160±2468
	Std.	2.59	20.01	16.56	729±1603	1.53	9.07	9.77	447±720
ReFTOSSA	Mean	9.58	69.32	61.40	2956±4499	7.93	62.83	50.81	3186±2488
	Std.	1.84	12.26	11.75	654±1591	1.16	7.01	7.59	373±706

TABLE IV

TYPING PERFORMANCE ACROSS STROKE PATIENTS FOR THE EYE-TRACKER ALONE WITH FTOSSA DESIGN IN EXPERIMENT 3.

Performance indexes	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10	Mean±Std.
Speed (letters/min)	12.04	7.36	6.21	8.26	5.78	9.22	11.52	9.82	8.27	12.26	9.07±2.33
ITR_{com} (bits/min)	80.82	68.37	48.76	70.38	53.68	67.60	82.61	68.67	68.78	81.46	69.11±11.20
ITR_{letter} (bits/min)	76.57	47.02	38.18	53.06	36.64	58.88	74.44	63.43	53.06	79.20	58.05±15.32
Average time (ms)	2430	2887	3897	2818	3649	2916	2413	2903	2882	2447	2924±1547

TABLE V

TYPING PERFORMANCE ACROSS STROKE PATIENTS FOR THE EYE-TRACKER ALONE WITH AOSSA DESIGN IN EXPERIMENT 3.

Performance indexes	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10	Mean±Std.
Speed (letters/min)	11.50	6.39	5.34	5.76	4.37	6.00	9.16	9.99	7.68	11.44	7.76±2.60
ITR_{com} (bits/min)	78.55	67.73	45.97	72.65	49.22	53.22	68.85	69.88	55.07	76.01	63.72±11.75
ITR_{letter} (bits/min)	73.48	40.97	34.30	37.05	28.09	38.40	59.16	64.54	49.62	73.90	49.95±16.76
Average time (ms)	2512	2924	4315	2709	4036	3716	2895	2852	3620	2622	3220±1849

B. Experiment 2

The typing performances of healthy participants for all five GUI layouts using the eye-tracker with soft-switch (ETSS) and eye-tracker only (ET) conditions are presented in Table III. The average typing speed with the FTOSSA layout (18.99±5.06 letters/min) was found superior among all the five designs during the ETSS condition. Subsequently, we found that the FTOSSA design leads to fastest performance in terms of speed and ITR among all the four designs ($p < 0.05$). In particular, a high speed of 28.06 letters/min was achieved by the participant H1. During ET condition, the average typing speed with the FTOSSA (11.39±0.84 letters/min) was also found greatest of all four designs. Consequently, we found that the FTOSSA leads to better performance in terms of speed and ITR than the other four designs ($p < 0.05$). Furthermore, statistical significance test was also used to compare the typing speed across the conditions. The average typing speed is measured with ETSS and ET for all GUI layouts and found that ETSS leads to a faster speed than ET ($p < 0.05$). The overall results showed that the top two performing layouts were the FTOSSA and the AOSSA.

C. Experiment 3

The typing performances of stroke patients for both the FTOSSA and the AOSSA layouts with the eye-tracker (ET) condition are presented in Table IV and Table V, respectively. The average typing speed with the FTOSSA design

(9.07±2.33 letters/min) was found superior when compared to the AOSSA design (7.76±2.60 letters/min), ($p < 0.05$). A similar pattern of performance is measured in terms of ITR_{com} and ITR_{letter} for the FTOSSA layout. The ITR_{com} and ITR_{letter} with the FTOSSA design (69.11±11.20 bits/min and 58.05±15.32 bits/min) were greater than the AOSSA design (63.72±11.75 bits/min and 49.95±16.76 bits/min), ($p < 0.05$). Hence, the ITRs of the stroke patients are comparable to healthy participants (compared to Table III). In particular, the stroke patient P1 with the loss of both motor and speech skills achieved 12.04 letters/min typing speed with the FTOSSA design.

D. Subjective evaluation

The system usability scale (SUS) [42] and NASA Task Load Index (NASA-TLX) [43] were used to further evaluate the participant's opinions on the systems usability level and the workload. These tests were conducted with the two best performing layouts on all the participants. The results of the system usability test for both the FTOSSA and the AOSSA layouts and for both healthy and patient groups are shown in Fig 5(a). Both layouts received an average SUS score above 85% for both groups of participants, which indicates an excellent grade on the adjective rating scale. Moreover, the FTOSSA design obtained a significantly higher SUS score than that of the AOSSA design (more than 4%) with both groups ($p < 0.05$). With the healthy participants, the SUS score was close to 90% which represents the best imaginable score

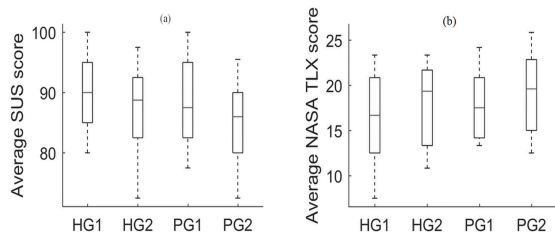


Fig. 5. The average SUS (a) and NASA TLX (b) scores for two best GUI design (FTOSSA (i.e. 1) and AOSSA (i.e. 2)) for both healthy (HG) and patient (PG) group of participants.*On each box, the central mark indicates the median, and the bottom and top edges of the box indicate the 25th and 75th percentiles, respectively

on the adjective rating scale with the FTOSSA layout in a single session [44]. The workload indices for both layouts and for both groups are presented in Fig 5(b). It was found that both groups experienced a lower workload while using the FTOSSA design as compared to the AOSSA design ($p < 0.05$). Moreover, the average NASA TLX weighted rating with the FTOSSA design was about 17 on a 0 to 100 scale for both healthy and patient groups, indicating that our proposed system can be used for the patients with the low workload.

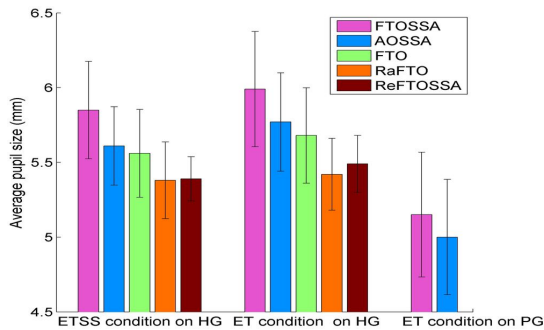


Fig. 6. The average pupil size across the proposed GUI designs for both eye-tracker with soft-switch (ETSS) condition and eye-tracker (ET) only condition with healthy (HG) and patient (PG) group of participants.

E. Pupillary data

The results of pupillary data of healthy group (HG) of participants across the five different layouts for both ETSS and ET conditions are shown in Fig 6. The average pupil size (in mm) with the FTOSSA design was found to be the highest ($p < 0.05$) among all the five designs during the ETSS condition (5.85 ± 1.03) and ET condition (5.99 ± 1.22). Furthermore, the results of pupillary data of patient group (PG) of participants across the FTOSSA and the AOSSA layouts for ET only condition are also shown in Fig 6. The average pupil size with the FTOSSA design was found to be larger than the AOSSA design (5.00 ± 1.22). However, we haven't found statistical significance between these two conditions ($p = 0.07$).

VIII. DISCUSSION

A recent review of virtual keyboard related studies provided several developments and optimization approaches for these systems [4]. The majority of the studies focused on improving

systems performance (increasing text entry rate and reducing error rate) without considering multimodality (i.e. the inclusion of several input devices). Moreover, there are limited studies related to gaze based virtual keyboards. In addition, the validations of most of the proposed systems are limited to healthy users. However, the performance can be affected significantly with patient groups, in particular, stroke patients. Keeping these issues in mind, an optimized gaze-controlled HVK has been designed wherein inputs can be provided using a portable non-invasive eye-tracker with or without a single input switch.

The main focus of this study was to design and optimize a gaze-controlled HCI system for stroke patients (mainly speech and motor impaired). There are four major findings of this study: First, we estimated the relative frequencies of all Hindi alphabets from a large corpus of characters. Second, we designed and implemented a tree-based menu selection method for the gaze-controlled GUI virtual keyboard to improve its speed and accuracy. Third, we have designed five different layouts of the proposed GUI for a HVK application. These layouts were designed on the basis of letter frequencies and the letters selection time. Fourth, the performances of these proposed GUI layouts were evaluated and compared empirically with healthy and patient groups using several performance indices. Furthermore, we have demonstrated the usability of the system and associated workload by implementing SUS questionnaires and NASA TLX test, respectively for the two best GUIs.

The performance evaluation of a virtual keyboard depends on several factors such as the nature and length of the typing task, the category of users (i.e. healthy or patient), and their experience and motivation in the typing task. For effectively accounting for all these factors, the performance evaluation of a virtual keyboard becomes challenging. Moreover, the typing speed can also be affected by the word completion and word prediction methods [27]. Thus, to avoid performance variations, we simply evaluated our system on the basis of a fixed number of commands per letter (i.e. 2 commands/letter) without any word completion or prediction procedure.

The study showed that average typing speed on a single session with the FTOSSA layout was significantly superior among all the five layouts with both eye-tracker only and eye-tracker with soft-switch conditions. In the current study we provided the results based on a single experimental session, however, the performance of the system with the FTOSSA layout can be further increased by providing some training sessions before evaluating the system. The average time for each command can be calculated using FTOSSA layout and may be used in later versions. Also, in future study, we will seek to utilize several phrases with uniform distribution to compute the average command frequency over the ten command boxes (items). Although a variation in performance was expected across conditions, the average performance with the FTOSSA layout in the eye-tracker only condition remains high enough to be used efficiently, including for stroke patients. As this study considered the patient group with varied time durations since the stroke, the performance evaluation showed efficient usage of the layout for a large range of stroke patients.

In this study the eye-tracker with soft-switch condition was evaluated with only healthy participants and the performance of the system was found superior when compared to the eye-tracker only condition. Furthermore, the performance can be improved by utilizing the eye-tracker with soft-switch for stroke patients as well. The soft-switch can be used using feet, elbow, or part of any limb, and hence it doesn't require the full operational ability to use the hand or a finger. It can therefore be used by patients who are not able to use a regular keyboard with their hands. It is exactly what can be faced by post stroke patients through their recovery and rehabilitation process where the available modalities can change and increase over time. This work focuses on the position of the items to be selected in the available commands, which can be used through gaze detection. In other words, this work improves the arrangement of the letters and symbols in the layout for increasing the typing speed. However, other parameters such as the number of commands, the distances between the commands, the feedback, and other elements could be further optimized.

The major confounding factor to achieve high accuracy and speed in an eye-tracker based system is the number of commands, which is further constrained by the quality of the calibration method. In terms of performance comparison, virtual keyboards based on brain response detection, such as event-related potentials (e.g. P300) and steady-state visual evoked potentials (SSVEP), offer significantly lower performance than the proposed system. Studies reported an average ITR of 25 bits/min with a P300 speller [45] and 37.62 bits/min (average speed of 5.51 letters/min) with a SSVEP speller [46]. The proposed system outperformed all these systems with an average ITR and average typing speed of 80.06 ± 4.97 bits/min and 11.38 ± 0.84 letters/min, respectively with healthy participants. Furthermore, it provides an average ITR and average typing speed of 69.8 ± 11.20 bits/min and 9.07 ± 2.33 letters/min, respectively with stroke patients. An individual stroke patient with the loss of both motor and speech skills achieved a high typing speed of 12.04 letters/min that demonstrated the high system usability. This system achieved an excellent grade on the adjective rating scale to the SUS and minimum work load (i.e. 17%) to the NASA TLX. In addition, the FTOSSA layout provides larger pupil size among all the five designs. An optimal performance was achieved by utilizing the baseline layout (AOSSA) for estimation of these parameters.

A recent study provided a viable usage of eye-gaze indices (i.e., eye fixation, smooth pursuit, and blinking) for the assessment of pathological states in stroke patients [47]. Therefore, a possible extension of the proposed system may be used for assessment purposes to help patients to determine their current pathological state. This system may also help patients to improve the loss of motor skills by the addition of other possible input modalities (e.g. motor imagery) wherein search and target selection of items can be achieved by gaze and motor imagery, respectively [48].

IX. CONCLUSION

This paper presented an optimized gaze-controlled virtual keyboard specifically created for the Hindi language that can be used by 490 million speakers. This virtual keyboard is based on a tree-based selection method and it requires only two commands to select any particular character. This work set out to identify a suitable Hindi virtual keyboard for stroke patients. The primary challenge was to accommodate a large set of alphabets, punctuation marks, and numbers in the display area, which was accomplished by proposing a tree-based menu optimization method. It also includes audio and visual feedback and incorporates 88 letters and extra commands such as delete, clear-all, and go-back to correct the errors. Five possible designs were proposed and empirically evaluated. It was found that the frequency and time-based organization with script specific arrangement design gives the best typing performance, high usability (87% SUS score), and low workload (NASA TLX with 17 scores) with stroke patients. Moreover, the optimization processes can be applied to any other language. The alphabetic organization with script specific arrangement can be directly applied to other languages such as Bengali (215 million speakers), Telugu (75 million), Tamil (77 million speakers), and Panjabi (88 million speakers) without including the optimization process. Therefore, the present research findings have potential applications for a large user population. Future work will involve further enhancement in the flexibility and usability of this system by adapting the dwell time for item selection.

ACKNOWLEDGMENT

Y.K.M. is supported by Govt. of India (Education-11016152013). G.P., K.W.-L., and H.C. are supported by the Northern Ireland Functional Brain Mapping Facility (1303/101154803). This work was supported by the Santander Mobility Grant by Ulster University and the UK-India Education and Research Initiative (UKIERI) Thematic Partnership project (DST/ME/20130354).

REFERENCES

- [1] "World health organization: Global report. 2011," http://www.who.int/disabilities/world_report/2011/en/, accessed:2016-9-9.
- [2] K. M. Wilkinson and S. Hennig, "The state of research and practice in augmentative and alternative communication for children with developmental/intellectual disabilities," *Ment Retard Dev Disabil Res Rev*, vol. 13, pp. 58–67, 2007.
- [3] D. C. DeCoste and S. Glennen, *The Handbook of Augmentative and Alternative Communication*. San Diego, London: Singular, 1997.
- [4] R. d. S. Gomide *et al.*, "A new concept of assistive virtual keyboards based on a systematic review of text entry optimization techniques," *Research on Biomed Eng*, vol. 32, no. 2, pp. 176–198, 2016.
- [5] M. L. Mele and S. Federici, "A psychotechnological review on eye-tracking systems: towards user experience," *Disabil Rehabil Assist Technol*, vol. 7, no. 4, pp. 261–281, 2012.
- [6] S.-W. Park *et al.*, "Augmentative and alternative communication training using eye blink switch for locked-in syndrome patient," *Ann Rehabil Med*, vol. 36, pp. 268–272, 2012.
- [7] P. Cipresso *et al.*, "The combined use of brain computer interface and eye-tracking technology for cognitive assessment in amyotrophic lateral sclerosis," in *Proc. IEEE Int. Conf. PervasiveHealth*, 2011, pp. 320–324.
- [8] G. Schalk *et al.*, "Brain-computer interfaces (bcis): Detection instead of classification," *J. Neurosci. Methods*, vol. 167, no. 1, pp. 51–62, 2008.

- [9] A. B. Usakli and S. Gurkan, "Design of a novel efficient human-computer interface: An electrooculogram based virtual keyboard," *IEEE Trans. Instr. and Meas.*, vol. 59, no. 8, pp. 2099–2108, 2010.
- [10] Y.-F. Fu *et al.*, "A fast text-based communication system for handicapped aphasiacs," in *Proc. IEEE Conf. IHH-MSP*, 2009, pp. 583–594.
- [11] U. Orhan *et al.*, "Rsvp keyboard: An eeg based typing interface," in *Proc. IEEE Int. Conf. Acoust Speech Signal Process.*, 2013, pp. 1–11.
- [12] P. Biswas and D. Samanta, "Friend: A communication aid for persons with disabilities," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 16, no. 2, pp. 205–209, 2008.
- [13] A. J. Molina, O. Rivera, and I. Gomez, "Measuring performance of virtual keyboards based on cyclic scanning," in *Proc. IEEE Int. Conf. Autonomic and Autonomous Systems*, 2009, pp. 174–178.
- [14] S. Ghosh and W. Stuerzlinger, "Designing an efficient virtual keyboard for text composition in bengali," in *Proc. ACM Int. Conf. IndiaHCI*, 2011, pp. 84–87.
- [15] S. Bhattacharya and S. Laha, "Bengali text input interface design for mobile devices," *Univ Access Inform Soc*, vol. 12, no. 4, pp. 441–451, 2013.
- [16] D. Samanta, S. Sarcar, and S. Ghosh, "An approach to design virtual keyboards for text composition in indian languages," *Int. J. of Human-Computer Interaction*, vol. 29, no. 8, pp. 516–540, 2013.
- [17] M. Shenoy, "Persons with disability and the india labour market: challenges and opportunities," *Int. Labour Organ.*, p. 46, 2011.
- [18] T. Mohanan, *Argument Structure in Hindi*. Stanford Uni: CSLI, 1994.
- [19] "Devanagari alphabet and its romanization," <http://hindinideshalaya.nic.in/>, accessed: 2017-01-21.
- [20] A. Joshi, A. Ganu, A. Chand, V. Parmar, and G. A. Mathur, "Keylekh: a keyboard for text entry in indic scripts," in *Proc. ACM Int. Conf. Human Factors in Computing Systems*, 2004, pp. 928–942.
- [21] S. Sarcar *et al.*, "Virtual keyboard design: State of the arts and research issues," in *Proc. IEEE Int. STS*, 2010, pp. 289–299.
- [22] A. Achom and A. Basu, "Design and evaluation of unicode compliance meitei/meetei mayek keyboard layout," in *Proc. IEEE Int. Symposium on Advanced Computing and Communication*, 2015, pp. 1–8.
- [23] D. J. Ward and J. C. MacKay, D, "Artificial intelligence: Fast hands-free writing by gaze direction," *Nature*, vol. 418, p. 838, 2002.
- [24] D. Rough *et al.*, "An evaluation of dasher with a high-performance language model as a gaze communication method," in *Proc. ACM Conf. on Advanced Visual Interfaces*, 2014, pp. 169–176.
- [25] H. Cecotti, "A multimodal gaze-controlled virtual keyboard," *IEEE Trans. Human-Machine Systems*, vol. 46, no. 4, pp. 601–606, 2016.
- [26] S. Sarcar and P. Panwar, "Eyeboard++: an enhanced eye gaze-based text entry system in hindi," in *Proc. ACM Int. Conf. Computer Human Interaction*, 2013, pp. 354–363.
- [27] D. Anson *et al.*, "The effects of word completion and word prediction on typing rates using on-screen keyboards," *Assist Technol*, vol. 18, no. 2, pp. 146–54, 2006.
- [28] S. Pouplin *et al.*, "Effect of a dynamic keyboard and word prediction systems on text input speed in patients with functional tetraplegia," *J. of Rehabil. Research & Dev.*, vol. 51, no. 3, pp. 467–480, 2014.
- [29] R. J. K. Jacob, "What you look at is what you get: eye movement-based interaction techniques," in *Proc. ACM Int. Conf. Human Factors in Computing Systems*, 1990, pp. 11–18.
- [30] V. Prabhu and G. Prasad, "Designing a virtual keyboard with multi-modal access for people with disabilities," in *Proc. IEEE Int. Conf. Information and Communication Technologies*, 2011, pp. 1133–1138.
- [31] J. V. Singh and G. Prasad, "Enhancing an eye-tracker based human-computer interface with multi-modal accessibility applied for text entry," *Int. J. of Computer Applications*, vol. 130, no. 16, pp. 16–22, 2015.
- [32] Y. K. Meena, H. Cecotti, K. Wong-Lin, and G. Prasad, "A novel multimodal gaze-controlled hindi virtual keyboard for disabled users," in *Proc. IEEE Int. Conf. Syst Man Cybern*, 2016, pp. 3688–3693.
- [33] S. Shanbhag *et al.*, "An intelligent multi-layered input scheme for phonetic scripts," in *Proc. ACM Symp. Smart Graphics*, 2002, pp. 35–38.
- [34] "Rajasthan patrika," <http://rajasthanpatrika.patrika.com/>, accessed:3-6-16.
- [35] "Bharatdarshan," <http://www.bharatdarshan.co.nz/>, accessed: 2016-4-1.
- [36] H. Pande and H. S. Dhama, "Mathematical modelling of occurrence of letters and words initials in texts of hindi language," *J. of Quantitative Linguistics*, vol. 20, pp. 19–38, 2013.
- [37] "Frequency table," <http://www.math.cornell.edu/~mec/2003-2004/cryptography/subs/frequencies.html>, accessed: 2016-10-9.
- [38] S. Bhattacharya and S. Laha, "Bengali text input interface design for mobile devices," *Univ Access Inf Soc*, vol. 12, pp. 441–451, 2013.
- [39] E. S. Dalmaijer, "Is the low-cost eyetribe eye tracker any good for research?" *PeerJ Preprints*, pp. 1–35, 2014.
- [40] H. M. Hodkinson, "Evaluation of a mental test score for assessment of mental impairment in the elderly," *Age Ageing*, vol. 1, no. 4, pp. 233–238, 1972.
- [41] R. C. Lyle, "A performance test for assessment of upper limb function in physical rehabilitation treatment and research," *International Journal of Rehabilitation Research*, vol. 4, no. 4, pp. 483–492, 1981.
- [42] J. Brooke, "Sus - a quick and dirty usability scale," *Usability evaluation in industry*, vol. 189, no. 194, pp. 4–7, 1996.
- [43] S. G. Hart *et al.*, "Development of nasa-tlx: Results of empirical and theoretical research," *Adv Psychol*, vol. 52, pp. 139–183, 1988.
- [44] A. Bangor *et al.*, "Determining what individual sus scores mean: Adding an adjective rating scale," *J. of usability studies*, vol. 4, no. 3, pp. 114–123, 2009.
- [45] G. Townsend *et al.*, "A novel p300-based brain-computer interface stimulus presentation paradigm: moving beyond rows and columns," *Clinical Neurophysiology*, vol. 121, no. 7, pp. 1109–1120, 2011.
- [46] H. Cecotti, "A self-paced and calibration-less ssvp-based brain-computer interface speller," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 18, no. 2, pp. 127–133, 2010.
- [47] D. Kumar, A. Dutta, A. Das, and U. Lahiri, "Smarteye: Developing a novel eye tracking system for quantitative assessment of oculomotor abnormalities," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 24, no. 10, pp. 1051–59, 2016.
- [48] Y. K. Meena *et al.*, "Towards increasing the number of commands in a hybrid brain-computer interface with combination of gaze and motor imagery," in *Proc. IEEE Conf. Med and Bio Soci*, 2015, pp. 506–09.